北京大学

本科生毕业论文

题目：数据分布视角下的可信机器学习

Data Distribution Insights into Trustworthy Machine Learning

姓　　名：魏泽明

学　　号：2000011015

院　　系：数学科学学院

专　　业：信息与计算科学

导师姓名：孙猛 教授

二〇二五 年 六 月

北京大学本科毕业论文导师评阅表

学生姓名	魏泽明	学号	2000011015
院系	数学科学学院	专业	信息与计算科学（数据科学方向）
指导教师	孙猛	职称	教授
毕业论文题目	数据分布视角下的可信机器学习		
导师评语	（包括但不限于对论文选题意义、行文逻辑、专业素养、学术规范以及是否符合培养方案目标等方面评价） 机器学习系统的可信性是当前人工智能领域的研究热点，论文从数据分布视角出发研究机器学习的多个可信性问题，选题具有重要的理论意义和实用价值。 论文取得的创新性研究成果如下： 1. 提出了可扩展的序列模型抽象提取与解释技术，实现了在更大规模循环神经网络上的抽象模型提取和机制解释； 2. 提出了分类校准的公平对抗训练算法，能够同时提升模型的整体对抗鲁棒性和类别间公平性； 3. 提出了上下文学习启发的大语言模型对抗攻防算法，效果显著优于已有方法，为理解和提升大语言模型安全性提供了新思路。 论文写作规范，结构清晰，逻辑严谨，实验数据充分，参考文献引用恰当，对应三篇论文已发表或投稿于相关领域的顶级会议或期刊，是一篇优秀的本科生毕业论文。论文工作表明魏泽明掌握了本学科的理论基础和专业知识，独立从事科研工作能力强，达到理学学士学位的培养目标。 导师签名：孙猛 2025 年 5 月 14 日		

版权声明

摘要

近年来，机器学习在多个领域取得了重要突破，但在实际应用中也面临着严重的可信性问题，如模型的透明性、鲁棒性和安全性等方面，这些问题可能导致模型在部署时产生不可靠、误导性甚至有害的结果。例如，深度神经网络的复杂性使其决策机制难以被直观理解，而对抗样本和越狱攻击则可能使机器学习模型产生不准确或有害的输出。

针对以上问题，本论文从数据分布的视角出发，为解决这些问题提供了新的思路。与现有关注优化算法或模型设计的可信机器学习研究不同，本论文探索了数据分布在理解与防御机器学习模型方面的潜力，这在模型规模不断增大的背景下有望突破传统方法面临的可扩展性等瓶颈。通过对特定数据分布特征的分析与利用，本论文为多种机器学习模型在机制解释、鲁棒泛化和安全对齐等方面提出了相应的理论与算法。

论文的第一部分提出了可扩展的序列模型抽象提取与解释技术。该框架受到模型处理自然语言分布动态特征的启发，尽管已有工作致力于从循环神经网络等序列机器学习模型中提取有限自动机以用于解释和分析，但现有方法在可扩展性或准确性方面存在局限性。本文中提出的提取与解释框架指出并解决了自然语言任务中自动机提取的迁移稀疏性和上下文丢失问题。该框架采用启发式方法补全状态转移图中缺失的规则，并调整转移矩阵以增强自动机的上下文感知能力，同时利用两种数据增强策略进一步提高提取精度。基于提取结果，本框架还提出了循环神经网络的机制解释方法，包括一种基于转移矩阵嵌入的词嵌入方法用于面向特定任务的循环神经网络机制解释。实验表明，所提出的方法不仅在提取精度上优于现有方法，且基于转移矩阵嵌入的解释方法能够有效用于模型预训练和对抗样本生成等可信应用场景。

论文的第二部分提出了分类校准的公平对抗训练算法。该算法基于对类别对抗数据分布的分析，能够同时提升模型的整体鲁棒性和最差类别鲁棒性。虽然对抗训练已被广泛认为是提高深度神经网络提高对抗鲁棒性的有效方法，但大多数现有工作主要关注提升模型的整体鲁棒性，忽略了最差类别鲁棒性对模型安全性带来的潜在隐患。本部分从理论和实证两方面研究了不同类别对训练中对抗配置的偏好，包括扰动幅度、正则化和权重平均等。基于这些研究，我们进一步提出了分类校准的公平对抗训练算法，能够自动为每个类别定制特定的训练配置，从而有效提升整体鲁棒性和类间公平性。

论文的第三部分提出了基于上下文数据分布的大语言模型对抗攻防算法。尽管大语言模型在各种应用中取得了显著成功，但仍然面临越狱攻击的威胁，存在输出有害

内容等隐患。针对这一问题，受大语言模型独特的上下文学习能力启发，本文探索了其在控制大语言模型安全对齐方面的潜力。在此视角下，本文提出了上下文攻击，通过有害示例破坏大语言模型的安全机制，以及上下文防御，通过展示拒绝产生有害回应的例子增强其安全鲁棒性。本文进一步提供理论见解，解释为何少量的对抗性上下文示例能够高效地控制模型安全对齐能力。实验表明，本文提出的上下文攻防算法在多个模型和数据集上具有有效性与可扩展性，为红队测试与实际部署中的安全防护提供了有效解决方案。这部分工作揭示了上下文学习在大语言模型安全中重要但尚未得到充分关注的作用，为理解和提升大语言模型安全开辟了新的范式。

总体而言，本论文从数据分布的视角探索了可信机器学习的研究，为机器学习模型的可解释性、鲁棒性和安全性提供了新的理论与算法。

关键词：可信机器学习，数据分布，机制解释，对抗鲁棒性，安全对齐

Data Distribution Insights into Trustworthy Machine Learning

Zeming Wei (Data Science and Big Data Technology)
Directed by Prof. Meng Sun

ABSTRACT

In recent years, machine learning (ML) has made milestone advancements across a variety of applications. However, it also faces significant trustworthiness issues that raise concerns about its practical implementation in real-world scenarios. These issues include perspectives like transparency, robustness, and safety, which can result in unreliable, misleading, or harmful outcomes in their deployment. For example, the complex nature of deep neural networks (DNNs) makes their decisions difficult to interpret. Additionally, adversarial examples and jailbreaking attacks can lead ML models to produce inaccurate or toxic outcomes.

This thesis aims to provide new insights into these research problems from the data distribution perspective. Complementary to existing trustworthy ML research that emphasizes optimization or model design, this thesis explores the potential of data distribution for understanding and defending ML models, which has notable potential as model sizes grow and traditional methods may encounter scalability bottlenecks. By analyzing and utilizing characteristics of specific data distributions, this thesis proposes theories and algorithms motivated by them for mechanism interpretation, robust generalization, and alignment inspection across different types of ML models.

The first part of this thesis proposes a scalable abstract model extraction and explanation framework inspired by model dynamic characteristics in processing natural language distributions. While many efforts have been made to extract finite automata from stateful ML models like Recurrent Neural Networks (RNNs) for explanation and analysis, existing approaches have limitations in either scalability or precision. In this part, the proposed framework of Weighted Finite Automata (WFA) extraction and explanation tackles the limitations for natural language tasks. First, to address the transition sparsity and context loss problems this thesis identified in WFA extraction for natural language tasks, this part proposes an empirical method to complement missing rules in the transition diagram, and adjust transition matrices to enhance the context-awareness of the WFA. This part also proposes two data augmentation tactics to track more dynamic RNN behaviors, which further allows us to improve the extraction precision.

Based on the extracted model, this part proposes an explanation method for RNNs, including a word embedding method named Transition Matrix Embeddings (TME) and TME-based task-oriented explanation for the target RNN. Our evaluation demonstrates the advantage of our method in extraction precision over existing approaches, and the effectiveness of the TME-based explanation method in applications to pretraining and adversarial example generation.

The second part of this thesis proposes an adversarial training algorithm for enhancing both overall and worst-class robustness, motivated by observations in analyzing class-wise adversarial data distributions. Though adversarial training has been widely acknowledged as the most effective method to improve the adversarial robustness against adversarial examples for DNNs, most existing works focus on enhancing the overall model robustness, treating each class equally during both the training and testing phases. However, there are few efforts aimed at ensuring fairness in adversarial training at the class level without compromising overall robustness. This part theoretically and empirically investigates the preference of different classes for adversarial configurations, including perturbation margin, regularization, and weight averaging. Motivated by these studies, this part further proposes a Class-wise calibrated Fair Adversarial training framework, named CFA, which customizes specific training configurations for each class automatically. Experiments on benchmark datasets demonstrate that our proposed CFA can improve both overall robustness and fairness notably over other state-of-the-art methods.

The third part of this thesis explores the safety of Large Language Models (LLMs) from an in-context data distribution perspective. LLMs have demonstrated remarkable success across diverse applications, yet their susceptibility to malicious exploitation remains a critical challenge. In this thesis, motivated by the unique effectiveness and scalability of In-Context Learning (ICL) in LLMs, this part explores its potential to modulate the safety alignment of LLMs. Specifically, this part proposes the In-Context Attack (ICA), which employs harmful demonstrations to subvert LLMs' safety, and the In-Context Defense (ICD), which bolsters their resilience through examples that demonstrate refusal to produce harmful responses. By adjusting the distribution of safety in LLM outputs through adversarial demonstrations, our proposed in-context attack and defense facilitate effective manipulation of their alignment. This part first provides theoretical insights to illustrate how minimal in-context demonstrations can efficiently alter safety alignment. Empirically, this part validates ICA and ICD across multiple models, datasets, and attack baselines, showing their efficacy and scalability for red-teaming evaluations and robust safeguards for real-world deployment. Overall, our work unveils the

pivotal yet understudied role of ICL in LLM safety, opening new avenues for understanding and improving it.

Overall, this thesis explores trustworthy ML research from data distribution perspectives, contributing novel insights into the interpretability, robustness, and safety of ML models.

KEY WORDS: Trustworthy Machine Learning, Data Distribution, Mechanism Interpretation, Adversarial Robustness, Safe Alignment

Contents

Chapter 1 Introduction

Machine Learning (ML) has made tremendous progress across various applications, such as image classification [42], semantic segmentation [41], and chat completion [87]. Despite these achievements, ML models face significant trustworthiness challenges, raising serious concerns about their practical use in the real world [69, 5, 130]. Generally, ML models can be classified into two categories: discriminative models and generative models. Discriminative models analyze input data to predict specific labels, such as those used in classification or image segmentation tasks. In contrast, generative models create content, such as text or images, based on user prompts. Both types of models rely on Deep Neural Networks (DNNs), which inherit a black-box nature that makes their decision-making processes difficult to understand. This lack of transparency raises concerns about their reliability, especially in safety-critical applications. Additionally, discriminative models are particularly vulnerable to adversarial examples [110, 39], where slight modifications to input data can lead to incorrect predictions. On the other hand, generative models pose risks of producing harmful or inappropriate content [106, 129], given their powerful generation capabilities. So far, most threads of research aimed at improving the trustworthiness of ML have focused on model-centric approaches, such as the design of robust modules or optimization algorithms. In this thesis, we take a different approach by examining the issue primarily from the perspective of data distribution, aiming to contribute new insights to the existing literature on trustworthy ML.

1.1 Motivation and Challenges

Along with the fast development of ML models, a broad series of trustworthy ML models and algorithms has been designed. For example, abstract models extraction techniques can provide interpretations for DNNs, while adversarial training has been acknowledged as one of the most effective approaches for improving their robustness. Unlike many existing works that explore model or optimization design, the foundational role of data distributions in trustworthy ML problems is relatively underexplored. This thesis aims to probe how to leverage data distributions to improve various trustworthy perspectives of ML models. The main challenges we focused on can be outlined as:

- For mechanism interpretability of stateful DNNs, model-based explanation methods are hard to scale up to large-scale data distributions like natural languages.

- For adversarial robustness of discriminative models, existing adversarial training methods mostly take different classes equally, yet overlook the unique function of class-wise distribution-specific characteristics.
- For safe alignment of generative models, existing evaluation or protection paradigms based on optimization or heuristic designs fail to leverage the safety properties of language distributions, making them inefficient or unscalable.

1.1.1 Scalable Abstract Model Extraction

Stateful ML models like recurrent neural networks (RNNs) have achieved great success in sequential data processing, *e.g.*, time series forecasting [19], text classification [123], and language translation [30]. However, the complex internal design and gate control of RNNs make the interpretation and analysis of their behaviors rather challenging.

Recently, much progress has been made to abstract RNN as a finite automaton with explicit states and transition matrices to characterize the behaviors of RNN in processing sequential data. Up to the present, a series of extraction approaches leverage explicit learning algorithms (*e.g.*, L^* algorithm [3]) to extract a surrogate model of RNN. Such an exact learning procedure has achieved great success in capturing the state dynamics of RNNs when processing formal languages [131, 132, 83]. However, the computational complexity of the exact learning algorithm limits its scalability to construct abstract models from RNNs for natural language tasks. Another technical line for automata extraction from RNNs is the compositional approach, which uses unsupervised learning algorithms to obtain discrete partitions of RNNs' state vectors and construct the transition diagram based on the concrete state dynamics of RNNs [122, 121, 37, 34, 36, 140], yet fall short in extraction precision. A precise and scalable extraction approach for RNNs in the context of natural language tasks is needed.

Regarding model-based explanation, current extraction methods are limited to utilizing finite automata as a global interpretable model with explicit states and transition rules for RNNs. The information extracted from the transition diagram of automata is not fully exploited in understanding RNN behaviors for natural language tasks. In particular, given that the alphabet size of natural language datasets is quite large, the extracted rules in the transition matrix are difficult to grasp and interpret. A more comprehensible explanation method that can effectively exploit the extracted information to assist in understanding RNN behaviors in scalable natural language distributions remains underexplored.

1.1.2 Fair Class-wise Adversarial Robustness

The vulnerability of DNNs against adversarial examples [110, 39] has caused serious concerns about their application in safety-critical scenarios [21, 73]. DNNs can be easily fooled by adding small, even imperceptible perturbations to the natural examples. To address this issue, numerous defense approaches have been proposed [91, 29, 138, 8, 82], among which Adversarial Training (AT) [75, 127] has been demonstrated as the most effective method to improve the model robustness against such attacks [6, 135].

Although AT and its variants have achieved certain robustness, there still exists a stark difference among class-wise robustness in adversarially trained models, *i.e.*, the model may exhibit strong robustness on some classes while it can be highly vulnerable on others, as firstly revealed in [143, 111, 12]. This disparity raises the issue of robustness fairness, which can lead to further safety concerns of DNNs, as the models that exhibit good overall robustness may be easily fooled on some specific classes, *e.g.*, the stop sign in automatic driving. To address this issue, Fair Robust Learning (FRL) [143] has been proposed, which adjusts the margin and weight among classes when fairness constraints are violated. However, this approach only brings limited improvement on robust fairness while causing a drop in overall robustness. A critical challenge in this regard is how to effectively leverage the characteristics of class-wise adversarial data distributions for improving robust generalization and fairness.

1.1.3 Large Language Model Safety

Large Language Models (LLMs) have achieved remarkable success across various tasks. However, their widespread use has raised serious safety concerns [5, 146, 20, 38], particularly regarding their potential for generating harmful content (*e.g.*, toxic, unethical, or illegal content). To mitigate these concerns, extensive efforts have been made to align these language models and prevent harmful outputs during the training [88, 9, 58] and fine-tuning phases [109, 28, 157]. These aligned language models are expected to properly refuse to answer harmful requests (*e.g.*, how to make a bomb). Despite these efforts, recent works show that even aligned LLMs are still vulnerable to adversarial attacks, typically called the *jailbreak* issue of LLMs [129, 97, 47, 2]. By crafting adversarial prompts, attackers may successfully bypass the safeguard of LLMs and induce them to generate unethical outputs.

Upon discovery and formulation of jailbreaking attacks [106, 71, 164], recent studies have established a preliminary research convention of jailbreaking attacks and defenses. Existing jailbreaking attacks can be generally categorized into two types: optimization-based and

template-based. Optimization-based attacks iteratively refine a harmful prompt with query or gradient heuristics to elicit the LLMs to generate harmful content [164, 70, 18, 77], yet often face the efficiency bottleneck. Template-based attacks manually design persuasive instructions and attach them to harmful prompts [129, 65, 149], but their derived jailbreak prompts lack flexibility. From the defense side, preliminary techniques are also proposed against jailbreaking, like pre-processing-based filtering [1, 15] and detection [95, 51] methods. Similar to the attack scenario, safety prompt templates for defenses like Self-reminder [141] are designed. However, current attack and defense techniques often encounter a scalability issue, as their prompts cannot be extended to achieve greater capabilities. For instance, template-based attacks can only create fixed jailbreaking prompts, making it difficult to enhance them for increased effectiveness. Such bottlenecks of existing methods call for a more efficient and scalable paradigm of evaluation and defense for LLM safety.

1.2 Contributions

Scalable automata extraction for natural language distribution. In this part, we propose a general framework of Weighted Finite Automata (WFA) extraction and explanation for RNNs to tackle the above challenges. To address the challenge of scalable extraction, we propose a complete pipeline to extract more precise automata for RNNs in the context of natural language distributions. We identify two problems that cause precision deficiency in natural language distributions: (1) *transition sparsity*: the transition dynamics are usually sparse in natural language tasks, due to the large alphabet size and the dependency on a finite set of (sequential) data in the extraction procedure. (2) *context loss:* the tracking of long-term context of RNNs (e.g., LSTM networks [44]) is inevitably compromised due to the abstraction. To deal with the transition sparsity problem, we propose a method to fill in the missing transition rules based on the semantics of abstract states. We also propose two tactics to augment the data samples, enabling the learning of more transition behaviors of RNNs, which further alleviates the transition sparsity problem. To enhance the context awareness of WFAs, we adjust the transition matrices to preserve partial context information from the previous states.

To address the challenge of effective explanation, we utilize the extracted WFAs to interpret the behaviors of RNNs. Motivated by the observation that the transition matrices of the extracted WFAs capture the behavior of the source RNNs, we propose a word embedding method – Transition Matrix Embeddings (TME) to construct task-oriented explanations for the target RNNs. Further, by leveraging the information captured in TME, we propose a global explana-

tion method for word attribution to RNNs' decisions and a contrastive method to investigate the difference between task-oriented TME and pretrained word embeddings (*e.g.*, Glove [93]). We validate the effectiveness of the contrastive explanation with applications to pretraining boost and adversarial example generation. The main contributions in this part can be summarized as follows:

- We propose a complete WFA extraction algorithm from RNNs specialized for natural language distributions.
- Experiments on benchmark datasets demonstrate that the proposed heuristic methods effectively improve the extraction precision by alleviating the transition sparsity and context loss problems.
- We propose a novel word embedding – Transition Matrix Embeddings (TME), based on which a global explanation method for word attribution and a contrastive approach for task-oriented explanation of RNNs are proposed.

Class-wise calibrated fair adversarial training. In this part, we first present some theoretical insights on how different adversarial configurations impact class-wise robustness, and reveal that strong attacks can be detrimental to the *hard* classes (classes that have lower clean accuracy). This finding is further empirically confirmed through evaluations of models trained under various adversarial configurations. Additionally, we observe that the worst robustness among classes fluctuates significantly between different epochs during the training process. It indicates that simply selecting the checkpoint with the best overall robustness like the previous method [103] may result in poor robust fairness, *i.e.*, the worst class robustness may be extremely low.

Inspired by these observations, we propose to dynamically customize different training configurations for each class, based on their adversarial data distributions. Note that unlike existing instance-wise customized methods that aim to enhance overall robustness [33, 10, 128, 24, 152], we also focus on the fairness of class-wise robustness. Furthermore, we modify the weight averaging technique to address the fluctuation issue during the training process. Overall, we name the proposed framework as **C**lass-wise calibrated **F**air **A**dversarial training (CFA). The main contributions in this part can be summarized as follows:

- We show both theoretically and empirically that different classes require appropriate training configurations. In addition, we reveal the fluctuating effect of the worst class robustness during adversarial training, which indicates that selecting the model with the best overall robustness may result in poor robust fairness.

- We propose a novel approach called Class-wise calibrated Fair Adversarial training (CFA), which dynamically customizes adversarial configurations for different classes during the training phase, and modifies the weight averaging technique to improve and stabilize the worst class's robustness.
- Experiments on benchmark datasets demonstrate that our CFA outperforms state-of-the-art methods in terms of both overall robustness and fairness, and can also be easily incorporated into other adversarial training approaches to further improve their performance.

Harnessing in-context learning for LLM safety. In this part, motivated by the unique effectiveness and scalability of In-Context Learning (ICL) [13, 35] in eliciting LLM capabilities, we explore how in-context data distributions can be utilized within the realm of jailbreaking. ICL is an intriguing property of LLMs that by prompting a few input-output pairs demonstrating a new task, LLMs can quickly adapt to the new task and give correct answers to new test examples *without* modifying any model parameters. Utilizing this property, we explore a new paradigm of adversarial attack and defense on LLMs, called **In-Context Attack (ICA)** and **In-Context Defense (ICD)**. Specifically, ICA incorporates demonstrations sampled from harmful data distributions that positively respond to malicious requests to the prompt. In turn, ICD utilizes a similar notion to defend LLMs with demonstrations sampled from safe data distributions, which teach the LLM to resist jailbreaking by adding a few examples of refusing harmful queries.

Notably, unlike conventional demonstrations used in ICL for a **particular** task, our harmful and safe demonstrations (**collectively called adversarial demonstrations**) are crafted to manipulate the **general** safety of LLMs, which means the task of the demonstrations may be **irrelevant** to the query task. For instance, harmful demonstrations of ICA on hacking into a secure network could successfully deceive the LLM into creating tutorials on bomb-making, which is unrelated to the demonstration. This intriguing property reveals that the safety distribution in LLM outputs can be easily manipulated by a few adversarial demonstrations of ICA and ICD. To intuitively understand this underlying mechanism, we build a theoretical framework to interpret the effectiveness of these adversarial demonstrations, where we illustrate how they can manipulate the safety of the LLM by inducing the generation distribution bias toward the target language distribution (harmful or safe).

We also present extensive experiments across various models, datasets, and attack baselines to demonstrate the effectiveness and potential of ICA and ICD as practical red-teaming

and safeguarding techniques. For instance, ICA achieved 81% Attack Success Rate (ASR) on jailbreaking GPT-4 [87] evaluated by the AdvBench dataset [164], and ICD can reduce the ASR of Llama-2 [112] against transferable GCG attack from 21% to 0% while maintaining the natural performance of LLMs. These remarkable results demonstrate the power of adversarial demonstrations, which suggests that aligned LLMs still have great flexibility to be revoked for certain beneficial or harmful behaviors using a few in-context demonstrations. Finally, we wrap up our evaluation by assessing the robustness of ICA and ICD in relation to their demonstration selection, highlighting the universality of adversarial demonstrations, and further exploring the interaction between ICA and ICD. In summary, our research reveals the essential role of in-context data distributions in modulating the safety of LLMs, paving the way for new evaluation and defense paradigms of LLM safety. The main contributions in this part can be summarized as follows:

- We explore the power of in-context learning in manipulating the safety of LLMs and propose In-Context Attack (ICA) and Defense (ICD) with adversarial demonstrations for jailbreaking and safeguarding purposes.
- Theoretically, we build a simplified framework to analyze how a few adversarial demonstrations can manipulate the safety of LLMs, offering insights into the effectiveness of ICA and ICD.
- Empirically, we show the effectiveness and practicality of ICA and ICD in terms of attacking and defending LLMs through comprehensive experiments, shedding light on the potential of adversarial demonstrations for advancing the safety and security of LLMs.

1.3 Thesis Outline

This thesis is outlined in the following structure. Chapter **2** presents the preliminaries about ML models and algorithms related to this thesis, followed by related work on model-based analysis, adversarial robustness, and LLM jailbreaking issues. Chapter **3** presents the weighted finite automata extraction and explanation framework designed for natural language distributions. In Chapter **4**, we present a class-wise adversarial data calibrated AT algorithm for robust generalization. In Chapter **5**, we present harnessing in-context data distributions for evaluating and safeguarding LLMs. Chapter **6** concludes the thesis and highlights potential future research directions.

Chapter 2 Preliminaries and Backgrounds

In this chapter, we present the preliminaries and related works of this thesis.

2.1 Preliminaries

We start with introducing backgrounds on ML models, from DNNs to RNNs and LLMs. Then we present the stateful abstract model, WFA, which was applied to our extraction framework. Lastly, we introduce backgrounds on adversarial robustness of DNNs and the in-context learning paradigm of LLMs.

2.1.1 Machine Learning Models

2.1.1.1 Deep Neural Networks (DNNs)

DNNs have revolutionized ML by demonstrating hierarchical feature learning capabilities surpassing traditional shallow architectures. Generally, a DNN comprises multiple hierarchical transformation layers that learn feature representations through nonlinear function composition, which are called Multilayer Perceptron (MLP) modules.

Multilayer Perceptron (MLP) Modules. An MLP model contains $L + 1$ perception layers, where the network depth L corresponds to the number of hidden layers. Taking fully connected layers as example, for an input vector in the l-th layer, $\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$, the activation $\mathbf{h}^{(l)}$ at layer l is computed as:

$$\mathbf{h}^{(l)} = \phi\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right) \tag{2.1}$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ denotes the weight matrix, $\mathbf{b}^{(l)} \in \mathbb{R}^{d_l}$ the bias vector, and $\phi(\cdot)$ an element-wise activation function (*e.g.*, ReLU: $\phi(z) = \max(0, z)$). During inference, the input data $\mathbf{x}$ is processed by the first layer, whose output $\mathbf{h}^{(1)}$ becomes the input vector of the next layer. Finally, the output of the last layer $\mathbf{h}^{(L)}$ gives the output of the model.

Parameter Optimizations. To learn the parameters of the DNNs, backpropagation efficiently computes gradients through automatic differentiation and the chain rule. Denote all learnable parameters in the DNN as θ, *e.g.* $\{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L}$ in the example above. These parameters can be optimized via empirical risk minimization (ERM):

$$\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathcal{L}(f(\mathbf{x};\theta), y)\right] \tag{2.2}$$

where $\mathcal{L}(\cdot)$ represents the task-specific loss function (e.g., cross-entropy for classification). Then, with the gradient descent algorithms, the parameters can be updated using a learning rate $\eta > 0$:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t), \tag{2.3}$$

where θ_t denotes the parameters in step t during optimization.

2.1.1.2 Recurrent Neural Networks (RNNs)

This section presents the notations and definitions for RNN abstraction and explanation. Given a finite alphabet Σ, we use Σ^* to denote the set of sequences over Σ and ε to denote the empty sequence. For $w \in \Sigma^*$, we use $|w|$ to denote its length, its i-th word as w_i, and its prefix with length i as $w[:i]$. For $x \in \Sigma$, $w \cdot x$ represents the concatenation of w and x.

Definition 1 (RNN) *A Recurrent Neural Network (RNN) for natural languages is a tuple $\mathcal{R} = (\mathcal{X}, \mathcal{S}, \mathcal{O}, f, p)$, where $\mathcal{X}$ is the input space; $\mathcal{S}$ is the internal state space; $\mathcal{O}$ is the probabilistic output space; $f : \mathcal{S} \times \mathcal{X} \to \mathcal{S}$ is the transition function; $p : \mathcal{S} \to \mathcal{O}$ is the prediction function.*

RNN Configuration. Our abstraction framework considers RNN a black-box model and focuses on its stepwise probabilistic output for each input sequence. The following configuration definition characterizes the probabilistic outputs in response to a sequential input fed to the RNN. Given an alphabet Σ, let $\xi : \Sigma \to \mathcal{X}$ be the function that maps each word in Σ to its embedding vector in $\mathcal{X}$. We define $f^* : \mathcal{S} \times \Sigma^* \to \mathcal{S}$ recursively as $f^*(s_0, \xi(w \cdot x)) = f(f^*(s_0, \xi(w)), \xi(x))$ and $f^*(s_0, \varepsilon) = s_0$, where s_0 is the initial state of $\mathcal{R}$. The RNN configuration $\delta : \Sigma^* \to \mathcal{O}$ is defined as $\delta(w) = p(f^*(s_0, w))$.

Output Trace. To record the stepwise behavior of RNN when processing an input sequence w, we define the *Output Trace* of w, *i.e.*, the probabilistic output sequence, as $T(w) = \{\delta(w[:i])\}_{i=1}^{|w|}$. The i-th item of $T(w)$ indicates the probabilistic output given by $\mathcal{R}$ after taking the prefix of w with length i as input.

2.1.1.3 Large Language Models (LLMs)

Building upon attention mechanisms [116], LLMs have achieved milestone success in modern ML research. We briefly formulate their principles below.

Transformer Architecture. The transformer architecture [116] introduced a paradigm shift in sequence modeling through self-attention mechanisms. Given an input sequence represented as token embeddings $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n denotes the sequence length (number of

tokens) and d specifies the embedding dimension, the scaled dot-product attention computes contextualized representations through learned linear projections.

The attention mechanism first transforms the input embeddings into three distinct components: queries $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$, keys $\mathbf{K} = \mathbf{X}\mathbf{W}_K$, and values $\mathbf{V} = \mathbf{X}\mathbf{W}_V$. Here, $\mathbf{W}_Q \in \mathbb{R}^{d\times d_k}$ and $\mathbf{W}_K \in \mathbb{R}^{d\times d_k}$ are trainable projection matrices that map the input embeddings into a d_k-dimensional key-query space, while $\mathbf{W}_V \in \mathbb{R}^{d\times d_v}$ projects the embeddings into a d_v-dimensional value space. The attention scores are computed through the matrix product $\mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{n\times n}$, whose elements $(\mathbf{Q}\mathbf{K}^\top)_{ij}$ represent the compatibility between the i-th query and j-th key. These weights are then multiplied with the value matrix $\mathbf{V}$ to produce the final attention output $\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{n\times d_v}$, where each row corresponds to a contextualized token representation aggregating information from all positions through the learned attention patterns. Overall, the transformer block in DNNs can be formulated as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}. \tag{2.4}$$

Pre-training Paradigms. Modern LLMs employ self-supervised pre-training objectives over web-scale corpora C. For example, the masked language modeling (MLM) [32] objective reconstructs randomly masked tokens:

$$\mathcal{L}_{\text{MLM}} = -\mathbb{E}_{\mathbf{x}\sim C} \sum_{m\in\mathcal{M}} \log P(x_m|\mathbf{x}_{\setminus\mathcal{M}}), \tag{2.5}$$

where $\mathcal{M}$ denotes the masked token positions. By contrast, the causal language modeling (CLM) [99] objective maximizes:

$$\mathcal{L}_{\text{CLM}} = -\mathbb{E}_{\mathbf{x}\sim C} \sum_{t=1}^{T} \log P(x_t|\mathbf{x}_{<t}). \tag{2.6}$$

Post-training Pipelines. After pre-training, LLMs often take post-training strategies before deployment. These post-training pipelines may include Supervised Fine-Tuning (SFT), which is domain adaptation on curated datasets $\mathcal{D}_{\text{SFT}} = \{(\mathbf{x}_i, \mathbf{y}_i^*)\}_{i=1}^N$:

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{|\mathbf{y}^*|} \log P(y_t^*|\mathbf{x}, \mathbf{y}_{<t}^*) \tag{2.7}$$

where $\mathbf{y}^*$ represents expert demonstrations. This phase typically uses a very small amount of pretraining compute while improving instruction-following capabilities. Besides, Reinforcement Learning from Human Feedback (RLHF) improves human alignment with human preference reward modeling. In this phase, human preferences are encoded through reward models

trained on pairwise comparisons, and the final policy of RLHF is optimized using algorithms like proximal policy optimization (PPO) [105].

2.1.2 Stateful Abstract Models

In this part, we elaborate on the preliminaries for the WFA used in our extraction scheme.

Definition 2 (WFA) *Given a finite alphabet Σ, a Weighted Finite Automaton (WFA) over Σ is a tuple $\mathcal{A} = (\hat{S}, \Sigma, E, \hat{s}_0, I, F)$, where $\hat{S}$ is the finite set of abstract states; $E = \{E_\sigma | \sigma \in \Sigma\}$ is the set of transition matrix E_σ with size $|\hat{S}| \times |\hat{S}|$ for each token $\sigma \in \Sigma$; $\hat{s}_0 \in \hat{S}$ is the initial state; I is the initial vector, a row vector with size $|\hat{S}|$; F is the final vector, a column vector with size $|\hat{S}|$.*

Abstract States. Given a RNN $\mathcal{R}$ and a dataset $\mathcal{D}$, let $\hat{O}$ denote all stepwise probabilistic outputs given by executing $\mathcal{R}$ on $\mathcal{D}$, i.e. $\hat{O} = \bigcup_{w \in \mathcal{D}} T(w)$. The abstraction function $\lambda : \hat{O} \to \hat{S}$ maps each probabilistic output to an abstract state $\hat{s} \in \hat{S}$. As a result, the output set is divided into a number of abstract states by λ. For each $\hat{s} \in \hat{S}$, the state $\hat{s}$ has explicit semantics that the probabilistic outputs corresponding to $\hat{s}$ has a similar distribution. In this paper, we leverage the *k-means* algorithm to construct the abstraction function. We cluster all probabilistic outputs in $\hat{O}$ into some abstract states. In this way, we construct the set of abstract states $\hat{S}$ with these discrete clusters and an initial state $\hat{s}_0$.

For a state $\hat{s} \in \hat{S}$, we define the *center* of $\hat{s}$ as the average value of the probabilistic outputs $\hat{o} \in \hat{O}$ which are mapped to $\hat{s}$. More formally, the center of $\hat{s}$ is defined as follows:

$$\rho(\hat{s}) = \underset{\lambda(\hat{o})=\hat{s}}{\text{Avg}} \{\hat{o}\}. \tag{2.8}$$

The center $\rho(\hat{s})$ represents an approximation of the distribution tendency of probabilistic outputs $\hat{o}$ in $\hat{s}$. We then use the center $\rho(\hat{s})$ as its weight for each state $\hat{s} \in \hat{S}$. The final vector F is thus formulated as $(\rho(\hat{s}_0), \rho(\hat{s}_1), \cdots, \rho(\hat{s}_{|\hat{S}|-1}))^t$.

Abstract Transitions. In order to capture the dynamic behavior of RNN $\mathcal{R}$, we define the abstract transition as a triple $(\hat{s}, \sigma, \hat{s}')$ where the original state $\hat{s}$ is the abstract state corresponding to a specific output y, i.e. $\hat{s} = \lambda(y)$; σ is the next word of the input sequence to consume; $\hat{s}'$ is the destination state $\lambda(y')$ after $\mathcal{R}$ reads σ and outputs y'. We use $\mathcal{T}$ to denote the set of all abstract transitions tracked from the execution of $\mathcal{R}$ on training samples.

Abstract Transition Count Matrices. For each word $\sigma \in \Sigma$, the abstract transition count matrix of σ is a matrix $\hat{T}_\sigma$ with size $|\hat{S}| \times |\hat{S}|$. The count matrices record the number

of times that each abstract transition is triggered. Given the set of abstract transitions $\mathcal{T}$, the count matrix of σ can be calculated as

$$\hat{T}_\sigma[i,j] = \mathcal{T}.count((\hat{s}_i, \sigma, \hat{s}_j)), \quad 1 \leq i,j \leq |\hat{S}|. \tag{2.9}$$

As for the remaining components, the alphabet Σ is consistent with the alphabet of the training set $\mathcal{D}$. The initial vector I is formulated according to the initial state $\hat{s}_0$. For an input sequence $w = w_1 w_2 \cdots w_n \in \Sigma^*$, the WFA will calculate its weight following

$$I \cdot E_{w_1} \cdot E_{w_2} \cdots E_{w_n} \cdot F. \tag{2.10}$$

2.1.3 Adversarial Robustness

We offer brief formulations for adversarial examples and AT is in this section.

Adversarial Examples. DNNs are known to be vulnerable to adversarial examples [39, 110], which can be generally formulated as:

$$\max_{\|x'-x\| \leq \epsilon} \mathcal{L}(\boldsymbol{\theta}; x', y), \tag{2.11}$$

where ϵ is the margin of perturbation, x is the original input example, x' is the adversarial example that fools the DNN (represented by parameter $\boldsymbol{\theta}$) into misprediction, and $\mathcal{L}$ is the loss function, *e.g.* the cross-entropy loss. Generally, Projected Gradient Descent (PGD) attack [75] has shown satisfactory effectiveness to find adversarial examples in the perturbation bound $\mathcal{B}(x,\epsilon) = \{x' : \|x'-x\| \leq \epsilon\}$, which is commonly used in solving the maximization problem:

$$x^{t+1} = \Pi_{\mathcal{B}(x,\epsilon)}(x^t + \alpha \cdot \text{sign}(\nabla_{x^t} \mathcal{L}(\boldsymbol{\theta}; x^t, y))), \tag{2.12}$$

where Π is the projection function and α controls the step size of gradient ascent.

Adversarial Training (AT). To defend against such attacks and robustify DNNs, AT has been acknowledged as one of the most reliable paradigms [6, 16]. AT can be formulated as the following min-max optimization problem:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{\|x'-x\| \leq \epsilon} \mathcal{L}(\boldsymbol{\theta}; x', y), \tag{2.13}$$

where $\mathcal{D}$ is the data distribution. TRADES [151] is another variant of AT, which adds a regularization term to adjust the trade-off between robustness and accuracy [113, 119]:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \{\mathcal{L}(\boldsymbol{\theta}; x, y) + \beta \max_{\|x'-x\| \leq \epsilon} \mathcal{K}(f_{\boldsymbol{\theta}}(x), f_{\boldsymbol{\theta}}(x'))\}, \tag{2.14}$$

where $\mathcal{K}(\cdot)$ is the KL divergence and β is the *robustness regularization* to adjust the robustness-

accuracy trade-off.

2.1.4 In-context Learning

In-Context Learning (ICL) [13, 35] is an intriguing property that emerges in LLMs in which they learn a specific task demonstrated by a few *input-label pair* examples. Formally, considering a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ in this task and given a demonstration set

$$C = \{(x_1, y_1), \cdots, (x_k, y_k) | f(x_i) = y_i\} \tag{2.15}$$

where x_i are query inputs and y_i are their corresponding labels in this task, a model can learn this mapping and successfully predict the label $y_{\text{new}} = f(\boldsymbol{x}_{\text{new}})$ of a new input query $\boldsymbol{x}_{\text{new}}$ by prompting $[x_1, y_1, \cdots, x_k, y_k, \boldsymbol{x}_{\text{new}}]$ as a concatenation of the demonstrations C and input query $\boldsymbol{x}_{\text{new}}$.

This mysterious property of LLMs has attracted much research attention on understanding [139, 81, 27, 23] and improving [158, 124, 80, 142] ICL. However, unlike existing works that mainly focus on leveraging ICL to improve the performance of a specific task (*e.g.*, classification), our work focuses on manipulating the safety level of LLMs, which can be regarded as a more general generation property. Therefore, our work collects demonstrations from broad types of tasks that are not necessarily aligned with the query $\boldsymbol{x}_{\text{new}}$, but at a specific safety level. Moreover, our proposed in-context attack and defense only require simple input-output pairs from the target safety distribution that do not require additional format editions, which is also different from general ICL application that often requires meticulous design of demonstration formats.

2.2 Related Work

2.2.1 Model-based Interpretation and Analysis

Many research efforts have been made to understand the behaviors of RNNs with an extracted model. We discuss the extraction methods and their applications respectively in the following.

2.2.1.1 Model Extraction of RNNs

As reviewed in [50], the extraction approach of RNNs can be divided into two categories: pedagogical approaches and compositional approaches.

Pedagogical Approaches. Many research works consider using pedagogical approaches to abstract RNNs by leveraging explicit learning algorithms, such as the L^* algorithm [3]. Earlier works date back to two decades ago, when Omlin et al. attempted to extract a finite model for Boolean-output RNNs [86, 84, 85]. Recently, Weiss et al. [131] proposed an approach to extracting DFA from RNN-acceptors based on L^* algorithm. Later, they presented a weighted extension of L^* algorithm that extracted probabilistic determininstic finite automata (PDFA) from RNNs [132]. Okudono et al. [83] proposed a weighted extension of L^* algorithm to extract WFA for real-value-output RNNs. Overall, pedagogical approaches have achieved great success in abstracting RNNs for small-scale languages, particularly formal languages. Such exact learning approaches have intrinsic limitation in scalability w.r.t. the language complexity, thus are not suitable for automata extraction for natural language models.

Compositional Approach. Another technical line for automata extraction from RNNs is the compositional approach, which leverages unsupervised algorithms (e.g. k-means, GMM) to cluster state vectors as abstract states [150, 17]. Wang et al. [122] studied the key factors in the compositional approach that influence the reliability of the extraction process, and proposed an empirical rule to extract DFA from RNNs. Later, Zhang et al. [155] followed the state encoding of compositional approach and proposed a WFA extraction approach from RNNs, which can be applied to both grammatical languages and natural languages. In this paper, the proposed WFA extraction approach from RNNs also falls into the line of compositional approach, but aims at proposing transition rule extraction method to address the transition sparsity problem and enhance the context-aware ability, which is customized for natural language tasks.

2.2.1.2 Model-based RNN Analysis and Explanation

There are a series of works focusing on deriving the extracted models for further applications, where the abstract models are more amenable to analysis and explanation.

Model-based Analysis. Model extraction techniques have been widely used to aid the analysis of RNNs, since the extracted models can be regarded as an approximation of the target RNNs, on which are easier to operate and perform analysis. [37] is a representative work for model-based RNN analysis, which leverages the extracted model to detect adversarial examples and increase test coverage of the target RNNs. Later, [36] proposed a model-based approach for robustness analysis of RNNs. Xie et al. [140] proposed to leverage the extracted model to identify buggy behaviors and further for automatic repairment of RNNs. In this paper, based on the extracted WFA, we proposed a new embedding method TME, which provides a new insight on RNN analysis for natural language tasks. With the proposed contrastive pairs derived by

TME, we can analyze task-oriented semantics of the target RNNs, which further can be applied to boost pretraining and adversarial example generation for RNNs.

Model-based Explanation. There are also several works devoted to explaining the mechanism of RNNs with the aid of surrogate models. Krakovna et al. [59] presented an interpretation method for RNNs by using hidden markov models (HMMs) to simulate the source RNNs. Hou et al. [45] proposed an approach to interpreting the effect of gates on the mechanism of RNNs by using the extracted finite state automata. Jiang et al. [55] proposed a hybrid model *FA-RNNs*, which is trainable, generalizable as well as interpretable. There are also works operating directly on the structure of RNNs. Guo et al. [40] proposed an interpretable LSTM neural network equipped with tensorized hidden states, which could learn variable-specific representations. In this work, by leveraging the extracted WFA, we proposed a global explanation method, which computes the word-wise influence score on RNN decisions, and a contrastive explanation method, where the identified collaborative and adversarial repairs effectively characterize the task-oriented semantics learned by the target RNN.

2.2.2 Adversarial Training

This section discusses the perspectives of AT related to our work, including two key concepts: class-wise adversarial robustness and adaptive AT.

2.2.2.1 Class-wise Adversarial Robustness

This metric focuses on the adversarial robustness of the DNN model exhibited on different classes, particularly on the worst class. Since adversaries may create examples specifically from one class, a difference between the worst-class robustness and overall robustness could provide a false sense of confidence in model safety when evaluating only the overall robustness. However, DNNs under AT consistently exhibit significant disparity of class-wise robustness, which was first revealed and discussed concurrently in [143] and [12]. For example, a model on the CIFAR-10 dataset [60] can achieve 50% average robust accuracy after AT, but its worst-class robustness may be lower than 20%. To mitigate this issue, FRL [143] is the first algorithm towards addressing this issue by enlarging the class-wise margin and weight, but it decreases the overall robustness significantly. Later, Tian et al. [111] analyzed the class-wise robustness systematically and showed that the robust fairness issue also exists in various datasets. Ma et al. [74] and Hu et al. [46] theoretically study the trade-offs between overall robustness and worst-class robustness through linear functions. Besides these, the unfairness issue in AT has not been well explored and solved yet.

2.2.2.2 Adaptive AT

To enhance the overall robustness of adversarial training, a series of adaptive AT methods were proposed [14, 33, 10, 128, 24, 127, 153]. These methods dynamically adjust the train configurations, *e.g.*, maximizing perturbation margin [33], adjusting loss function [128], and early-stopping on attack iteration [152], typically based on customization based on instance-wise analysis. Though achieving improvement over vanilla AT, the instance-wise methods only focus on improving overall robustness, yet overlook robust fairness. Moreover, how to leverage class-wise data distributions for adaptive AT, which may calibrate the training configurations more holistically than instance-wise data adjustment, has also not been explored in this literature.

2.2.3 Jailbreaking Attack and Defense

As for the LLM safe alignment, we present jailbreaking attacks and defenses, as well as in-context perspectives on alignment that are related to this thesis, in the following.

2.2.3.1 Jailbreaking Attack on LLMs

Despite techniques like Reinforcement Learning from Human Feedback (RLHF) [88, 9, 58, 109] are dedicated to aligning the value of LLMs with humans and teaching them not to generate any harmful content [94, 52, 28], recent studies show that LLMs are still vulnerable against jailbreaking attacks [71, 106, 129], where carefully crafted jailbreak prompts can bypass the safeguard of LLMs and trick them generate the requested harmful content. One popular thread of attacks attempts to manually design jailbreaking templates that attach a persuasive instruction to the harmful prompt, like DAN (*do anything now*) [106], prefix injection (*start with "sure, here's"*) [129] and DeepInception [65] that construct a fictitious scene to modify the personification ability of LLMs.

Another line of research extends these manually designed templates by optimizing an adversarial substring in the jailbreak prompt with heuristics derived from gradients or queries. **Gradient**-heuristic attacks like GCG attack [108, 164] which attach a suffix to the harmful request and then optimize it with gradient heuristics, but often require the white-box access to the victim model and also face the bottleneck of optimization efficiency [160, 156]. Besides, **Query**-heuristic attacks derive jailbreak prompts by collecting responses from the model with existing prompts and then refining the jailbreak prompt with them. For example, AutoDAN [70] and GA [63] utilize genetic algorithms to refine the prompt, while PAIR [18] and

TAP [77] use another red-teaming LLM to achieve this.

In addition to these template- and optimization-based attacks, other attack paradigms like cipher encoding [148, 102, 56], low-resource languages [31, 147], and training generative models for jailbreaking prompts [67, 92, 11, 62] are also explored. Nevertheless, current jailbreaking techniques encounter a shared limitation: their capabilities are often static and hard to scale up.

2.2.3.2 Defending LLMs against Jailbreaking

In response to these jailbreaking attacks, several preliminary defense techniques are designed. Notably, unlike conventional neural networks where Adversarial Training (AT) [75] is one of the most effective defenses against adversarial attacks [16], the huge amount of parameters and data of LLMs makes it impractical and ineffective to direct conduct AT on them [51, 136, 107]. Therefore, current defense methods are typically designed during the inference of LLMs, including pre-processing, inference, and post-processing stages.

Pre-processing methods detect or purify the potential harmful prompts before generation, like perplexity filter [1], harmful string detection [61, 15], retokenization [51], and prompt smoothing [22, 104]. These methods are easy to plug into the model but may cause unaffordable false positives [115]. Inference-based defenses incorporate safe instruction into the prompt [141] or modify decoding logic [66, 145]. Post-processing monitors the output [48] or hidden spaces of LLMs [154, 161]. Notably, the evaluations of most defenses except safe prompts are generally not adaptive [22], which undermines the reliability of their evaluation. Besides, many existing defenses face the bottleneck of over-refusal [89] and computational overhead [145] issues, limiting their practicality for real-world applications.

2.2.3.3 In-Context Perspectives for LLM Alignment

The mysterious ICL ability of LLMs has inspired researchers to investigate diverse aspects of LLM alignment from in-context perspectives. For example, URIAL [68] improves the alignment of LLMs by in-context token distribution shift, and CaC [125] studies the self-correction ability of LLMs through in-context alignment. Another research position considers attacking LLMs with ICL, like injecting backdoor triggers [159, 137] or undermining classification robustness [120, 101, 98] with malicious in-context demonstrations.

In addition to these different perspectives of LLM alignment, this work places a primary focus on the safety of LLMs with a particular view on jailbreaking. Concurrent with our work, a few ICL-based jailbreak attacks were also proposed, like long-context window scaling [4] or

incorporating system tokens and random search [90]. Unlike those focusing on tricks of ICL prompt design, our work investigates the fundamental problem of manipulating the safety of LLMs with naive adversarial demonstrations, from both attack and defense perspectives.

Chapter 3 Scalable Automata Extraction and Explanation

This chapter proposes a scalable automata extraction and explanation framework designed for natural language distributions, which is organized as follows. In Section 3.1, we present our automata extraction approach, including an overview of the automata extraction procedure, the transition rule complement method for transition sparsity, the transition rule adjustment method for context-awareness enhancement, and the data augmentation tactics. In Section 3.2, we present the experimental evaluation of the extraction consistency of our approach on two natural language tasks. We introduce the transition matrix embedding-based explanation framework for RNNs in Section 3.3, and also discuss our extraction algorithm, including computational complexity analysis and applicability to other RNNs, at the end of this Section. Finally, we summarize this chapter in Section 3.4.

3.1 Weighted Automata Extraction Scheme

3.1.1 Overview

We present the workflow of our extraction procedure in Figure 3.1. As the first step, we generate an augmented sample set $\mathcal{D}$ from the original training set $\mathcal{D}_0$ to enrich the transition dynamics of RNN behaviors and alleviate the transition sparsity. Then, we execute RNN $\mathcal{R}$ on the augmented sample set $\mathcal{D}$, and record the probabilistic output trace $T(w)$ of each input sentence $w \in \mathcal{D}$. With the output set $\hat{O} = \bigcup_{w \in \mathcal{D}} T(w)$, we cluster the probabilistic outputs into abstract states $\hat{S}$, and generate abstract transitions $\mathcal{T}$ from the output traces $\{T(w)|w \in \mathcal{D}\}$. All transitions constitute the abstract transition count matrices $\hat{T}_\sigma$ for all $\sigma \in \Sigma$.

Next, we construct the transition matrices $E = \{E_\sigma|\sigma \in \Sigma\}$. Based on the abstract states $\hat{S}$ and count matrices $\hat{T}$, we construct the transition matrix E_σ for each word $\sigma \in \Sigma$. Specifically, we use frequencies to calculate the transition probabilities. Suppose that there are n abstract states in $\hat{S}$. The i-th row of E_σ, which indicates the probabilistic transition distribution over states when $\mathcal{R}$ is in state $\hat{s}_i$ and consumes σ, is calculated as

$$E_\sigma[i, j] = \frac{\hat{T}_\sigma[i, j]}{\sum_{k=1}^{n} \hat{T}_\sigma[i, k]}. \tag{3.1}$$

This empirical rule faces the problem that the denominator of (3.1) could be zero, which means that the word σ never appears when the RNN $\mathcal{R}$ is in abstract state $\hat{s}_i$. In this case, one should

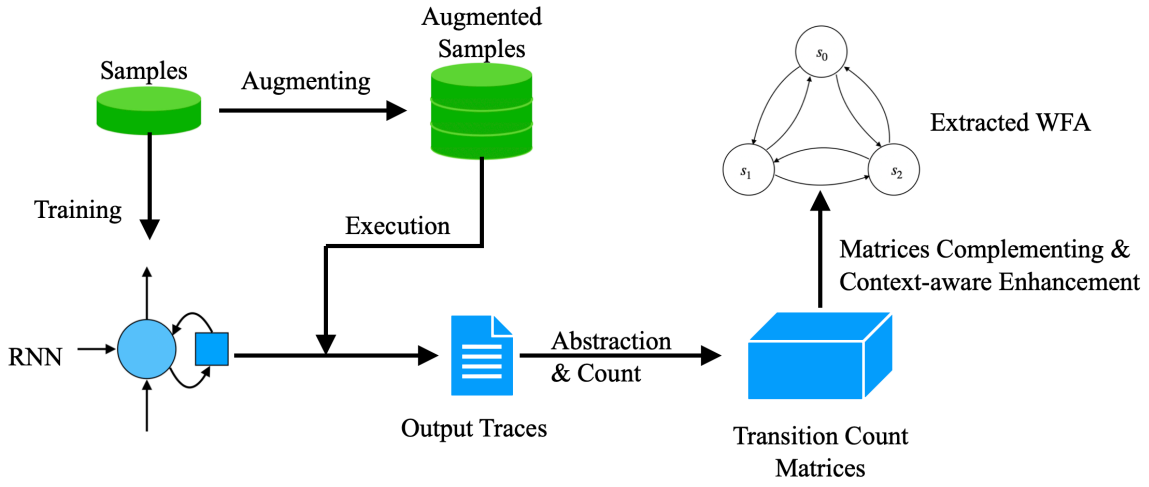


Figure 3.1 An illustration of our approach to extracting WFA from RNN.

decide how to fill in the transition rule of the *missing rows* in E_σ. In Section 3.1.2, we present a novel approach for transition rule complement. Further, to preserve more contextual information, we propose an approach to enhancing the context-awareness of WFA by adjusting the transition matrices, which is presented in Section 3.1.3.

3.1.2 Missing Rows Complement

Existing approaches for transition rule extraction usually face the problem of transition sparsity, *i.e.*, *missing rows* in the transition diagram. In the context of formal languages, the probability of the occurrence of missing rows is quite low, since the size of the alphabet is small and each token in the alphabet can appear sufficient times. However, in the context of natural language processing, the occurrence of missing rows is quite frequent. The following proposition gives an approximation of the occurrence frequency of missing rows.

Proposition 1 *Assume an alphabet Σ with $m = |\Sigma|$ words, a natural language dataset $\mathcal{D}$ over Σ which has N words in total, a RNN $\mathcal{R}$ trained on $\mathcal{D}$, the extracted abstract states $\hat{S}$ and transitions $\mathcal{T}$. Let σ_i denote the i-th most frequent word occurred in $\mathcal{D}$ and $t_i = \mathcal{T}.count((*, \sigma_i, *))$ indicates the occurrence times of σ_i in $\mathcal{D}$. The median of $\{t_i | 1 \leq i \leq m\}$ can be estimated as*

$$t_{[\frac{m}{2}]} = \frac{2N}{m \cdot \ln m}. \tag{3.2}$$

Proof. The Zipf's law [96] shows that

$$\frac{t_i}{N} \approx \frac{i^{-1}}{\sum_{k=1}^{m} k^{-1}}. \tag{3.3}$$

Note that $\sum_{k=1}^{m} k^{-1} \approx \ln m$ and take i to be $\frac{m}{2}$, we complete our proof.

Example 1 *In the QC news dataset [64], which has $m = 20317$ words in its alphabet and $N = 205927$ words in total, the median of $\{t_i\}$ is approximated to $\frac{2N}{m \cdot \ln m} \approx 2$. This indicates that about half of E_σ are constructed with no more than 2 transitions. In practice, the number of abstract states is usually far more than the transition numbers of these words, making most rows of their transition matrices missing rows.*

Filling the missing row with $\vec{0}$ is a simple solution, since no information was provided from the transitions. However, as estimated above, this solution will lead to the problem of transition sparsity, *i.e.*, the transition matrices for uncommon words are nearly null. Consequently, if the input sequence includes some uncommon words, the weights over states tend to vanish. We refer to this solution as *null filling*.

Another simple idea is to use the uniform distribution over states for fairness. In [132], the uniform distribution is used as the transition distribution for unseen tokens in the context of formal language tasks. However, for natural language processing, this solution still loses information about the current word, despite the fact that it avoids the weight vanishing over states. We refer to this solution as *uniform filling*. Besides, [155] uses the *synonym* transition distribution for an unseen token at a certain state. However, it increases the computation overhead when performing inference on test data, since it requires calculating and sorting the distance between the available tokens at a certain state and the unseen token.

To this end, we propose a novel approach to constructing the transition matrices based on two empirical observations. First, each abstract state $\hat{s} \in \hat{S}$ has explicit semantics, *i.e.*, the probabilistic distribution over labels, and similar abstract states tend to share more similar transition behaviors. The semantic distance between abstract states is defined as follows.

Definition 3 (State Distance) *For two abstract states $\hat{s}_1$ and $\hat{s}_2$, the distance between $\hat{s}_1$ and $\hat{s}_2$ is defined by the Euclidean distance between their center:*

$$dist(\hat{s}_1, \hat{s}_2) = \|\rho(\hat{s}_1) - \rho(\hat{s}_2)\|_2.$$

We calculate the distance between each pair of abstract states, which forms a *distance matrix M* where each element

$$M[i, j] = dist(\hat{s}_i, \hat{s}_j), \quad 1 \le i, j \le |\hat{S}|. \tag{3.4}$$

For a missing row in E_σ, following the heuristics that similar abstract states are more likely to have similar behaviors, we observe the transition behaviors from other abstract states, and

simulate the missing transition behaviors weighted by the distance between states. Particularly, in order to avoid numerical underflow, we leverage *softmin* on distance to bias the weight to states that share more similarity. Formally, for a missing row $E_\sigma[i]$, the weight of information set for another row $E_\sigma[j]$ is defined by $e^{-M[i,j]}$.

Second, it is also observed that sometimes the RNN just remains in the current state after reading a certain word. Intuitively, this is because some words in the sentence do not deliver significant information in the task. Therefore, we consider simulating behaviors from other states whilst remaining in the current state with a certain probability.

In order to balance the trade-off between referring to behaviors from other states and remaining still, we introduce a hyper-parameter β named *reference rate*, such that when WFA is faced with a missing row, it has a probability of β to refer to the transition behaviors from other states, and in the meanwhile has a probability of $1-\beta$ to keep still. We select the parameter β according to the proportion of self-transitions, *i.e.*, transitions $(\hat{s}, \sigma, \hat{s}')$ in $\mathcal{T}$ where $\hat{s} = \hat{s}'$.

To sum up, the complete transition rule for the missing row is

$$E_\sigma[i,j] = \beta \cdot \frac{\sum_{k=1}^{n} e^{-M[i,k]} \cdot \hat{T}_\sigma[k,j]}{\sum_{l=1}^{n}\sum_{k=1}^{n} e^{-M[i,k]} \cdot \hat{T}_\sigma[k,l]} + (1-\beta)\cdot\delta_{i,j}. \tag{3.5}$$

Here $\delta_{i,j}$ is the Kronecker symbol:

$$\delta_{i,j} = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases}. \tag{3.6}$$

In practice, we can calculate $\sum_{k=1}^{n} e^{-M[i,k]} \cdot \hat{T}_\sigma[k,j]$ for each j and then make division on their summation once and for all, which can reduce the computation overhead on transition rule extraction.

3.1.3 Context-Awareness Enhancement

For NLP tasks, the memorization of long-term context information is crucial. One of the advantages of RNN and its advanced designs, like LSTM networks, is the ability to capture long-term dependency. We expect the extracted WFA to simulate the step-wise behaviors of RNNs whilst keeping track of context information along with the state transition. To this end, we propose an approach to adjusting the transition matrix such that the WFA can remain in the current state with a certain probability.

Specifically, we select a hyper-parameter $\alpha \in [0,1]$ as the *static probability*. For each

word $\sigma \in \Sigma$ and its transition matrix E_σ, we replace the matrix with the *context-awareness enhanced matrix* $\hat{E}_\sigma$ as follows:

$$\hat{E}_\sigma = \alpha \cdot I_n + (1 - \alpha) \cdot E_\sigma \tag{3.7}$$

where I_n is the identity matrix.

The context-awareness enhanced matrix has explicit semantics. When the WFA is in state $\hat{s}_i$ and ready to process a new word σ, it has a probability of α (the *static probability*) to remain in $\hat{s}_i$, or follows the original transition distribution $E_\sigma[i, j]$ with a probability $1 - \alpha$.

Here we present an illustration of how context-awareness enhanced matrices deliver long-term context information through the proposition below. Suppose that a context-awareness enhanced WFA $\mathcal{A}$ is processing a sentence $w \in \Sigma^*$ with length $|w|$. We denote d_i as the distribution over all abstract states after $\mathcal{A}$ reads the prefix $w[: i]$, and particularly $d_0 = I$ is the initial vector of $\mathcal{A}$. We use Z_i to denote the decision made by $\mathcal{A}$ based on d_{i-1} and the original transition matrix E_{w_i}. Formally, $d_i = d_{i-1} \cdot \hat{E}_{w_i}$ and $Z_i = d_{i-1} \cdot E_{w_i}$. The d_i can be regarded as the information obtained from the prefix $w[: i]$ by $\mathcal{A}$ before it consumes w_{i+1}, and Z_i can be considered as the decision made by $\mathcal{A}$ after it reads w_i.

Proposition 2 *The i-th step-wise information d_i delivered by processing $w[: i]$ contains the decision information Z_j of prefix $w[: j]$ with a proportion of $(1 - \alpha) \cdot \alpha^{i-j}$, $1 \leq j \leq i$.*

Proof. Since $\hat{E}_{w_i} = \alpha \cdot I_n + (1 - \alpha) \cdot E_{w_i}$, we can calculate that

$$d_i = d_{i-1} \cdot \hat{E}_{w_i} = d_{i-1} \cdot [\alpha \cdot I_n + (1 - \alpha) \cdot E_{w_i}] = \alpha \cdot d_{i-1} + (1 - \alpha) \cdot Z_i. \tag{3.8}$$

Using (3.8) recursively, we have

$$d_i = (1 - \alpha) \sum_{k=1}^{i} \alpha^{i-k} \cdot Z_k + \alpha^i \cdot I.$$

This analysis shows the information delivered by $w[: i]$ refers to the decision made by $\mathcal{A}$ on each prefix included in $w[: i]$, and the portion vanishes exponentially. The effectiveness of the context-awareness enhancement method for transition matrix adjustment will be discussed in Section 3.2.

A concrete example of the pipeline. The following example presents the complete approach for transition rule extraction, *i.e.*, to generate the transition matrix $\hat{E}_\sigma$ with the missing row filled in and context enhanced, from the count matrix $\hat{T}_\sigma$ for a word $\sigma \in \Sigma$.

Example 2 *Assume that there are three abstract states in* $\hat{S} = \{\hat{s}_1, \hat{s}_2, \hat{s}_3\}$. *Suppose the count matrix for* σ *is* $\hat{T}_\sigma$.

$$\hat{T}_\sigma = \begin{bmatrix} 1 & 3 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, E_\sigma = \begin{bmatrix} 0.25 & 0.75 & 0 \\ 0.5 & 0.5 & 0 \\ 0.15 & 0.35 & 0.5 \end{bmatrix}, \hat{E}_\sigma = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.4 & 0.6 & 0 \\ 0.12 & 0.28 & 0.6 \end{bmatrix}. \tag{3.9}$$

For the first two rows (states), there exist transitions for σ. Thus, we can calculate the transition distribution of these two rows in E_σ in the usual way. However, the third row is a *missing row*. We set the *reference rate* as $\beta = 0.5$, and suppose that the distance between states satisfies $e^{-M[1,3]} = 2e^{-M[2,3]}$, generally indicating the distance between $\hat{s}_1$ and $\hat{s}_3$ is nearer than $\hat{s}_2$ and $\hat{s}_3$. With the transitions from $\hat{s}_1$ and $\hat{s}_2$, we can complement the transition rule of the third row in E_σ through (3.5). The result shows that the behavior from $\hat{s}_3$ is more similar to $\hat{s}_1$ than $\hat{s}_2$, due to the smaller distance. Finally, we construct $\hat{E}_\sigma$ with E_σ. Here we set the *static probability* $\alpha = 0.2$, thus $\hat{E}_\sigma = 0.2 \cdot I_3 + 0.8 \cdot E_\sigma$. The result shows that the WFA with $\hat{E}_\sigma$ has a higher probability of remaining in the current state after consuming σ, which can preserve more information from the prefix before σ.

3.1.4 Data Augmentation

Our proposed approach for transition rule extraction provides a solution to the transition sparsity problem. Still, we hope to learn more dynamic transition behaviors from the target RNN, especially for the words with relatively low frequency, to characterize their transition dynamics sufficiently based on the finite data samples. Different from formal languages, we can generate more natural language samples automatically, as long as the augmented sequential data are reasonable with clear semantics and compatible with the original learning task. Based on the augmented samples, we are able to track more behaviors of the RNN and build the abstract model with higher precision. In this section, we introduce two data augmentation tactics for natural language processing tasks: *Synonym Replacement* and *Dropout*.

Synonym Replacement. Based on the distance quantization among the word embedding vectors, we can obtain a list of synonyms for each word in Σ. For a word $\sigma \in \Sigma$, the *synonyms* of w are defined as the top-k most similar words of σ in Σ, where k is a hyper-parameter and we set k to 5 by default based on an empirical observation that top-5 similar words are sufficiently reasonable to keep the semantics. The similarity among the words is calculated based on the Euclidean distance between the word embedding vectors over Σ.

Given a dataset $\mathcal{D}_0$ over Σ, for each sentence $w \in \mathcal{D}_0$, we generate a new sentence w'

by replacing some words in w with their synonyms. Specifically, each word is replaced by a randomly selected synonym in its top-k synonyms with probability p_r (0.4 by default).

Dropout. Inspired by the regularization technique *dropout*, we also propose a similar tactic to generate new sentences from $\mathcal{D}_0$. Initially, we introduce a new word named *unknown word* and denote it as $\langle$**unk**$\rangle$. For the sentence $w \in \mathcal{D}_0$ that has been processed by synonym replacing, we further replace the words that haven't been replaced with $\langle$**unk**$\rangle$ with a certain probability p_d (0.2 by default). Finally, new sentences generated by both synonym replacement and dropout form the augmented dataset $\mathcal{D}$.

With the dropout tactic, we can observe the behaviors of RNNs when it processes an unknown word $\hat{\sigma} \notin \Sigma$ that hasn't appeared in $\mathcal{D}_0$. Therefore, the extracted WFA can also have better generalization ability. The complete pipeline of the data augmentation algorithm is elaborated in Algorithm 11. Note that the $rand()$ function samples from $[0, 1]$ in a uniform manner.

Algorithm 1: Data Augmentation for Transition Rule Extraction

Input: Original dataset $\mathcal{D}_0$, hyper-parameter $k = 5$, $p_r = 0.4$, $p_d = 0.2$
Output: Augmented dataset $\mathcal{D}$

```
Obtain the synonyms σ_1, σ_2, ··· , σ_k of each word σ ∈ w in the vocabulary of D;
D ← {};
for each sentence w ∈ D_0 do
    for each word σ ∈ w do
        if rand() < p_r then
            Replace σ with selected synonym from {σ_1, σ_2, ··· , σ_k};
        else
            if rand() < p_d then
                Replace σ with ⟨unk⟩;
    Obtain a new sentence w', and add w to D;
return D;
```

We illustrate the above data augmentation algorithm using the following example to generate a new sentence w' from $\mathcal{D}_0$.

Example 3 *Consider a sentence w from the original training set $\mathcal{D}_0$, w =['I', 'really', 'like', 'this', 'movie'].*

First, the word 'like' is chosen to be replaced by one of its synonyms, 'appreciate'. Next, the word 'really' is dropped from the sentence, *i.e.*, replaced by the unknown word $\langle$**unk**$\rangle$.

Finally, we get a new sentence w' =['I', '⟨**unk**⟩', 'appreciate', 'this', 'movie'] and put it into the augmented dataset $\mathcal{D}$. Since the word 'appreciate' may be an uncommon word in Σ, we can capture new transition information provided by RNNs. We can also capture the behavior of RNN when it reads an unknown word after the prefix ['I'].

Note that the role of *data augmentation* in our extraction approach is different from that used in the training phase of RNNs. While data augmentation used in the training phase aims to improve the performance of RNNs, the goal of data augmentation in this work is to improve the WFA extraction precision. To this end, we use data augmentation in the testing phase to extract more transition dynamics to construct the abstract model.

3.2 Evaluation

In this section, we evaluate our extraction approach on two natural language datasets and demonstrate its performance on precision and scalability.

3.2.1 Datasets and RNNs

We select two popular datasets for NLP tasks and train the target RNNs on them.

- The CogComp QC Dataset (abbrev. QC) [64] contains news titles which are labeled with different topics. The dataset is divided into a training set containing 20k samples and a test set containing 8k samples. Each sample is labeled with one of seven categories. We train an LSTM model $\mathcal{R}$ on the training set, which achieves an accuracy of 81% on the test set.
- The Jigsaw Toxic Comment Dataset (abbrev. Toxic) [57] contains comments from Wikipedia's talk page edits, with each comment labeled toxic or not. We select 25k non-toxic samples and toxic samples respectively, and divide them into the training set and test set in a ratio of four to one. We train an LSTM model which achieves 90% accuracy on the test set.

Metrics. We use *Consistency Rate (CR) and Jensen–Shannon Divergence (JSD)* as our evaluation metrics. For a sentence in the test set $w \in \mathcal{D}_{test}$, we use $\mathcal{R}(w)[i]$ and $\mathcal{A}(w)[i]$ to denote the prediction score on class i of the RNNs and WFA, respectively. The *Consistency Rate* measures the consistency of the output decision between the two models, which is formally defined as

$$CR = \frac{|\{w \in D_{test} : \arg\max_i \mathcal{A}(w)[i] = \arg\max_i \mathcal{R}(w)[i]\}|}{|\mathcal{D}_{test}|}. \tag{3.10}$$

Table 3.1 Evaluation results of different filling approaches on missing rows.

Dataset	QC			Toxic		
Metric	CR(↑)	JSD(↓)	Time(s)	CR(↑)	JSD(↓)	Time(s)
$\mathcal{A}_0$	0.26	0.25	47	0.57	0.09	167
$\mathcal{A}_U$	0.60	0.21	56	0.86	0.06	180
$\mathcal{A}_E$	**0.80**	**0.10**	70	**0.91**	**0.02**	200

The *Jensen–Shannon Divergence* [78] measures the distance of two probability distributions, i.e., the outputs of WFA and RNN, which is formally defined as

$$JSD = \frac{1}{2}\sum_i (\mathcal{A}(w)[i]\log(\frac{2\mathcal{A}(w)[i]}{\mathcal{A}(w)[i]+\mathcal{R}(w)[i]}) + \mathcal{R}(w)[i]\log(\frac{2\mathcal{R}(w)[i]}{\mathcal{A}(w)[i]+\mathcal{R}(w)[i]})). \tag{3.11}$$

Note that the Consistency Rate measures the consistency of the classification decision between the WFA and the RNN, while Jensen–Shannon Divergence evaluates the similarity of the output probability distributions between the two models. These two metrics evaluate the consistency between the abstract model and RNN to a different degree. In this chapter we mainly focus on the consistency of predicted labels, hence we apply Consistency Rate as our major measurement.

3.2.2 Missing Rows Complementing

As discussed in Section 3.1.2, we take two approaches as baselines, the *null filling* and the *uniform filling*. The extracted WFA with these two approaches are denoted as $\mathcal{A}_0$ and $\mathcal{A}_U$, respectively. The WFA extracted by our *empirical filling* approach is denoted as $\mathcal{A}_E$.

Table 3.1 shows the evaluation results of three rule filling approaches. We conduct the comparison experiments on QC and Toxic datasets and select the cluster number for state abstraction as 40 and 20 for the QC and Toxic datasets, respectively.

The three rows labeled with the type of WFA show the evaluation results of different approaches. For the $\mathcal{A}_0$ based on null filling, the WFA returns the weight of most sentences in $\mathcal{D}$ with $\vec{0}$, which fails to provide sufficient information for prediction. For the QC dataset, only a quarter of the sentences in the test set are classified correctly. The second row shows that the performance of $\mathcal{A}_U$ is better than $\mathcal{A}_0$. The last row presents the evaluation result of $\mathcal{A}_E$, which fills in the missing rows by our approach. In this experiment, the hyperparameter *reference rate* is set as $\beta = 0.3$. We can see that our empirical approach achieves significantly better accuracy, which is 20% and 5% higher than uniform filling on the two datasets, respectively. As for JSD,

Table 3.2 Evaluation results of with and without context-awareness enhancement.

Rule	Config.	QC			Toxic		
		CR(↑)	JSD(↓)	Time(s)	CR(↑)	JSD(↓)	Time(s)
$\mathcal{A}_U$	None	0.60	**0.21**	56	0.86	**0.06**	180
	Context	**0.71**	0.22	64	**0.89**	**0.06**	191
$\mathcal{A}_E$	None	0.80	**0.10**	70	0.91	**0.02**	200
	Context	**0.82**	0.13	78	**0.92**	0.03	211

we can see that our empirical approach also outperforms the baselines notably over both QC and Toxic datasets.

The columns labeled *Time* show the execution time of the whole extraction workflow, from tracking transitions to evaluation on test set, but not include the training time of RNNs. We can see that the extraction overhead of our approach ($\mathcal{A}_E$) is about the same as $\mathcal{A}_U$ and $\mathcal{A}_0$.

3.2.3 Context-Awareness Enhancement

In this experiment, we leverage the context-awareness enhanced matrices when constructing the WFA. We adopt the same configuration on cluster numbers n as the comparison experiments above, *i.e.*, $n = 40$ and $n = 20$. The experiment results are summarized in Table 3.2. The columns titled *Config.* indicate if the extracted WFA leverages context-awareness matrices. We also take the WFA with different filling approaches, the uniform filling and empirical filling, into comparison. Experiments on null filling are omitted due to limited precision.

For the QC dataset, we set the *static probability* as $\alpha = 0.4$. The consistency rate of WFA $\mathcal{A}_U$ improves 11% with the context-awareness enhancement, and $\mathcal{A}_E$ improves 2%. As for the Toxic dataset, we take $\alpha = 0.2$ and the consistency rate of the two WFA improves 3% and 1% respectively. This shows that the WFA with context-awareness enhancement retains more context information from the prefixes of sentences, making it simulate RNNs' classification decision better. However, the WFA equipped with context-awareness enhancement exhibits larger JSD, which is caused by the fact that context-awareness enhancement reduces the transition magnitude, since larger α leads to higher probability of remaining in the current state. This reveals a trade-off between the abstraction precision evaluated by decision label consistency and prediction score consistency.

Still, the context-awareness enhancement processing costs little time, since we only calculate the adjusting formula (3.7) for each E_σ in E. The additional extra time consumption is

Rule	Data	QC			Toxic		
		CR(↑)	JSD(↓)	Time(s)	CR(↑)	JSD(↓)	Time(s)
$\mathcal{A}_U$	$\mathcal{D}_0$	0.71	0.22	64	0.89	0.06	191
	$\mathcal{D}$	**0.76**	**0.18**	81	**0.91**	**0.05**	295
$\mathcal{A}_E$	$\mathcal{D}_0$	0.82	0.13	78	0.92	0.03	211
	$\mathcal{D}$	**0.84**	**0.12**	85	**0.94**	**0.02**	315

Table 3.3 Evaluation results of with and without data augmentation.

8s for the QC dataset and 11s for the Toxic dataset.

3.2.4 Data Augmentation

Finally, we evaluate the WFA extracted with transition behaviors from augmented data. Note that the two experiments above are based on the primitive dataset $\mathcal{D}_0$. In this experiment, we leverage the data augmentation tactics to generate the augmented training set $\mathcal{D}$, and extract WFA with data samples from $\mathcal{D}$. In order to get the best performance, we build WFA with context-awareness-enhanced matrices.

Table 3.3 shows the results of the consistency rate of WFA extracted with and without augmented data. The rows labeled $\mathcal{D}_0$ show the results of WFA that are extracted with the primitive training set, and the result from the augmented data is shown in rows labeled $\mathcal{D}$. With more transition behaviors tracked, the WFA extracted with $\mathcal{D}$ demonstrates better precision. Specifically, the WFA extracted with both empirical filling and context-awareness enhancement achieves a further 2% increase in consistency rate on the two datasets. In addition, the extractions with augmented data also exhibit better JSD.

To summarize, by using our transition rule extraction approach, the consistency rate of extracted WFA on the QC dataset and the Toxic dataset achieves 84% and 94%, respectively. Taking the primitive extraction algorithm with uniform filling as the baseline, whose experimental results in terms of CR are 60% and 86%, our approach achieves an improvement of 22% and 8% in consistency rate. Regarding the Jensen–Shannon Divergence, though the context-awareness enhancement makes a little drop, our approach still outperforms the baseline methods significantly. Taking uniform filling for comparison, our overall approach improves the JSD from 0.21 to 0.12 on QC dataset and 0.06 to 0.02 on Toxic dataset. For the time complexity, the time consumption of our approach increases from 56s to 81s on QC dataset, and from 180s to 315s on Toxic dataset. There is no significant time cost increase when adopting our approach for complicated natural language tasks. We can conclude that our transition rule

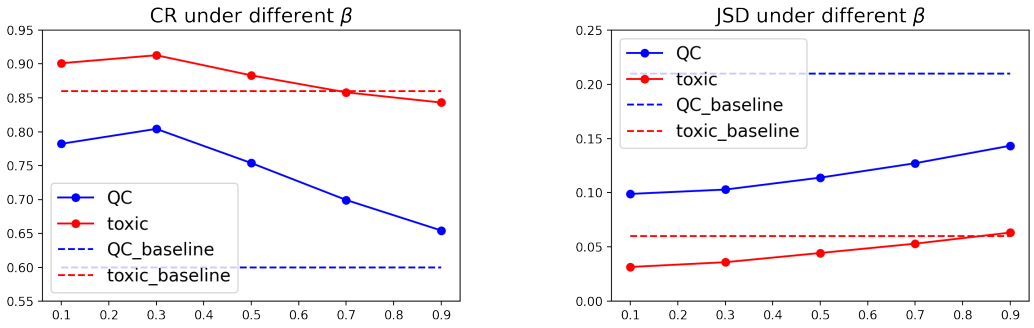


Figure 3.2 CR and JSD on the two datasets under different β.

extraction approach makes a better approximation of RNNs, and is also efficient enough to be applied to practical applications for large-scale natural language tasks.

3.2.5 Parameter Effect Evaluation

In this section, we conduct experiments to evaluate the impact of the hyperparameter on the validity of our extraction approach, including the reference rate β, static probability α, and the number of clusters K.

Reference rate β. We first evaluate the impact of the *reference rate*. To this end, we set β to different values from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. Meanwhile, we set α to a fixed value 0. The results are shown in Figure 3.2, where we take the uniform filling as baseline (the dotted lines). We observe that our filling method outperforms uniform filling for a large range of parameter values (less than 0.7), under both CR and JSD metrics. A relatively small β (e.g., less than 0.5) leads to better extraction precision.

Static probability α. Similarly, we conduct an experiment to evaluate the impact of different *static probability* values $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ on the performance of our approach. We set β to 0.3 based on the above evaluation results. The results are illustrated in Figure 3.3. Compared with the case of $\alpha = 0$ where we do not apply the context-awareness enhancement, it leads to improvements on the CRs when setting α to values from $\{0.2, 0.4\}$. Meanwhile, as discussed before, context-awareness enhancement reduces the transition scale of WFA, which leads to performance degradation in terms of JSD. This reveals a trade-off between CR and JSD among different selections on α. Based on the results, we suggest setting α to a small positive value (less than 0.4).

Cluster number K. Finally, we evaluate the impact of cluster number $K \in \{10, 20, 30, 40, 50\}$ on the performance of our approach. The results are shown in Figure 3.4, where our approach

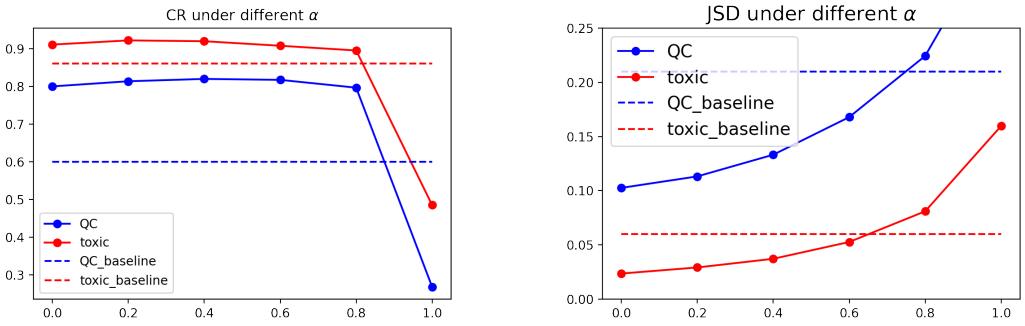
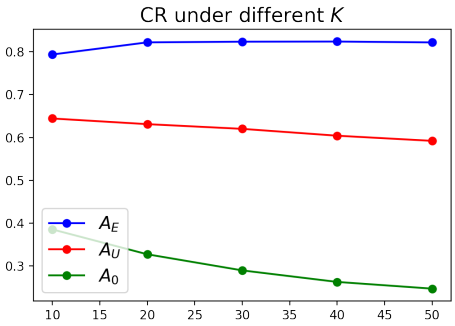
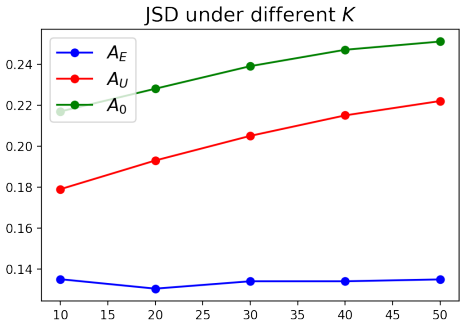


Figure 3.3 CR and JSD on the two datasets under different α.

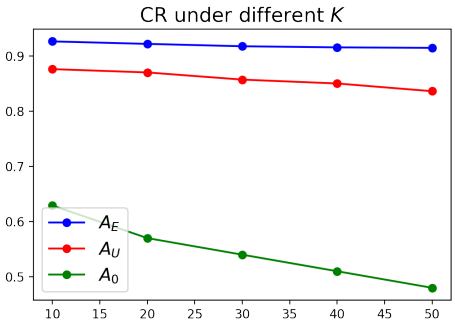
is denoted by A_E, and the uniform filling and null filling are denoted as A_U, A_0 respectively. We can see that our approach outperforms the baselines in all cases. Note that as K increases from 10 to 50, the performance of A_U and A_0 consistently decreases, which is caused by the identified transition sparsity problem. The evaluation results demonstrate the robustness of our method against the cluster number K.



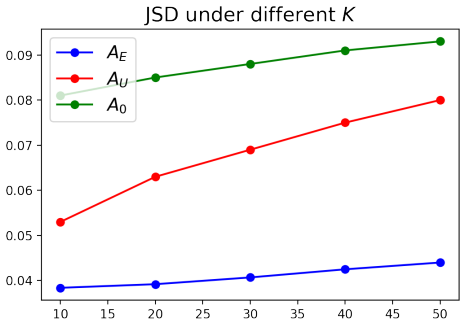
(a) CR on QC dataset



(b) JSD on QC dataset



(c) CR on Toxic dataset



(d) JSD on Toxic dataset

Figure 3.4 Overall comparison under different K.

3.3 Weighted Automata-based Explanation of RNNs

In this section, we propose a novel explanation framework of RNNs for natural language tasks based on the extracted WFA, as illustrated in Figure 3.5. In the explanation framework, we consider using the transition matrix① E_σ for each word σ as its word embedding, named as Transition Matrix Embeddings (TME). In the following, we first introduce TME and explain its difference from traditional pretrained word embeddings. Next, we present a global explanation and contrastive explanation method based on TME to interpret the behaviors of RNNs, including two perspectives:

1. *Global explanation*, where we use TME to calculate the word-wise attribution of RNN's decisions,
2. *Contrastive explanation*, where we compare TME with the conventional word embedding method to analyze the task-oriented word semantics learned by RNNs.

We also reveal two intriguing properties of RNNs identified by the contrastive method, and validate the effectiveness of the proposed embedding and explanation framework for RNNs by applying it to pretraining and adversarial example generation.

3.3.1 Transition Matrix as Word Embeddings

Suppose the extracted WFA $\mathcal{A}$ from RNN $\mathcal{R}$ has $n = |\hat{S}|$ words. For each word σ and its corresponding transition matrix $E_\sigma \in \mathbb{R}^{n\times n}$ in $\mathcal{A}$, recall that the (i, j)-th element of E_σ represents the transition probability of $\mathcal{A}$ from s_i to s_j after reading word σ, which is an approximation of the transition probability of $\mathcal{R}$ between these two states. Therefore, if two words share similar transition matrices, they trigger similar behaviors of RNN $\mathcal{R}$, and hence represent similar semantics from the RNN $\mathcal{R}$'s perspective for the current task.

This observation motivates us to use the transition matrices to craft word embeddings. In order to obtain the task-oriented embedding vector, we flatten the transition matrix $E_\sigma \in \mathbb{R}^{n\times n}$ into a vector $\boldsymbol{e}_\sigma$ of $N = n^2$ dimensions:

$$\boldsymbol{e}_\sigma[j + n \cdot (i - 1)] = E_\sigma[i, j], \quad \text{for } 1 \le i, j \le n, \tag{3.12}$$

which we refer to as Transition Matrix Embeddings (TME).

Note that TMEs are fundamentally different from the traditional pretrained word embeddings,*e.g.*, Word2vec [79], Gloves [93]. The TME characterize transition behaviors of RNNs when processing each word, which are task oriented (*e.g.*, text classification), while pretrained

① In this section, we focus on the final extracted transition matrix $\hat{E}_\sigma$, and abuse the notation E_σ for the sake of simplicity.

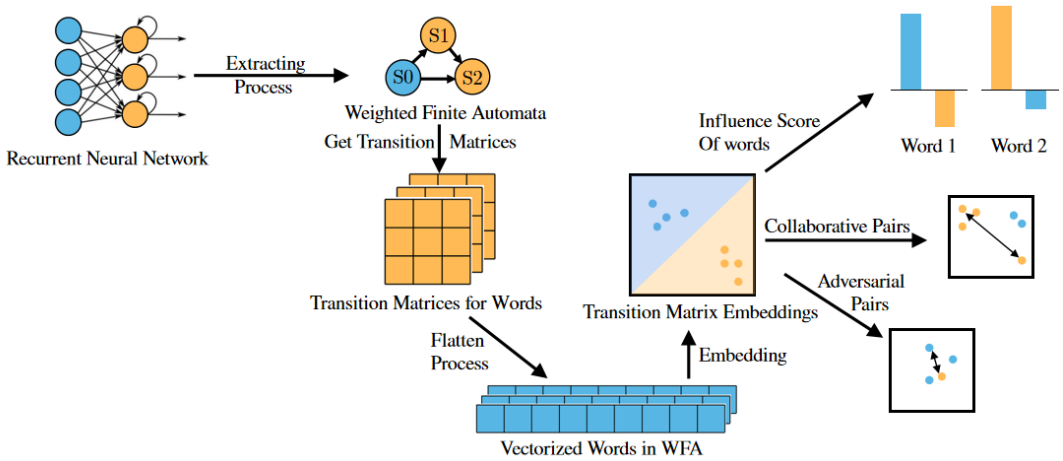


Figure 3.5 WFA-based explanation framework.

embeddings are word-semantic oriented. Therefore, for two words σ_1 and σ_2, they may share similar TME $\boldsymbol{e}_{\sigma_1}, \boldsymbol{e}_{\sigma_2}$, yet represent different semantics and hence their embedding differences in pretrained embeddings may be large. We detail such cases in Section 3.3.3. Further, the applications of TME are different from the general pretrained embeddings. The extracted TME can be seen as a global explanation of the source RNN $\mathcal{R}$ (see Section 3.3.2 for details), which aids us in understanding the decision logic of the source RNN.

3.3.2 Word-Wise Attribution with TME

We introduce a global explanation method based on TME for analyzing the decision attribution of the source RNN $\mathcal{R}$. We investigate the impact of each word σ on the decision of $\mathcal{R}$ based on its TME $\boldsymbol{e}_\sigma$. Recall that each abstract state $\hat{s}$ has an explicit semantic represented by its center $\rho(\hat{s})$, the probability distribution of labels. Therefore, each transition between two states, such as $\hat{s}_1 \longrightarrow \hat{s}_2$, can be interpreted as a shift in the probability distribution of labels, $\rho(\hat{s}_1) \longrightarrow \rho(\hat{s}_2)$. By multiplying the transition probability between states, which is saved in $\boldsymbol{e}_\sigma$, we can calculate the average variation of prediction scores among labels after $\mathcal{R}$ reading word σ. More formally, the variation contributed by the abstract transition $(\hat{s}_i, \sigma, \hat{s}_j)$ is given by $\boldsymbol{e}_\sigma[j + n \cdot (i-1)] \times (\rho(\hat{s}_j) - \rho(\hat{s}_i))$.

Additionally, we take into account the uneven significance among all abstract states, where states that appear more frequently should be assigned a larger weight. To reflect this, we calculate the frequency of each abstract state $\hat{s}$ as $u(\hat{s})$ and incorporate it into the computation of

Table 3.4 Top-10 Influential Words for 7 classes in the QC news Dataset [64]

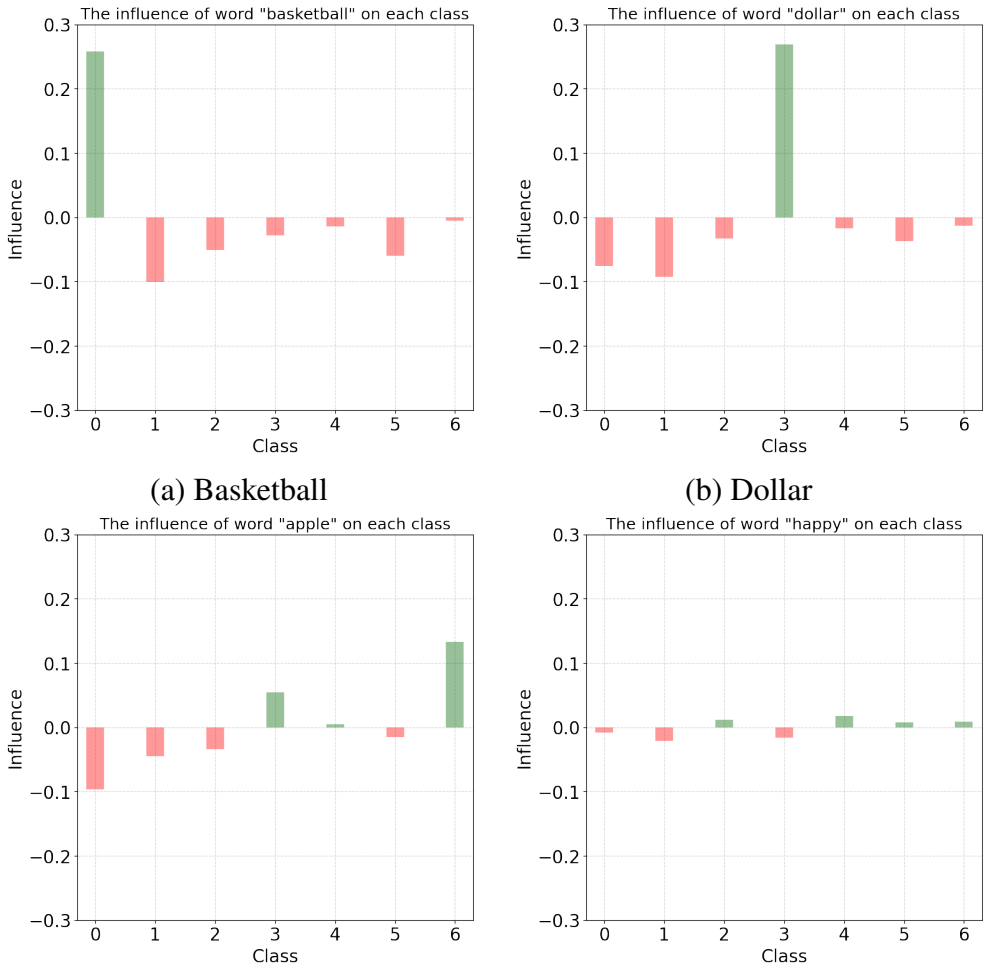
Label	Category	Top-10 Influencial Words
0	Sport	players, lockout, rangers, sox, knicks, basketball, coach, bruins, champions, djokovic
1	World	libyan, pakistan, yemen, sudan, gaddafi, syrian, egypt, mubarak, syria, afghans
2	US	florida, county, wildfire, layoffs, massachusetts, mid-atlantic, wildfires, cpsc, firefighters, blagojevich
3	Business	stocks, dollar, consumer, goldman, mortgage, wall, companies, rosneft, s&p, sec
4	Health	disease, crackers, cancer, asthma, patients, exercise, prevention, symptoms, therapy, obesity
5	Entertainment	idol, cannes, arthur, gotti, beaver, mccreery, mariah, lohan, baldwin, diana
6	Sci_tech	3ds, playstation, icloud, software, gmail, windows, tablets, linkedin, tablet, climate

the influence score as a weight. Formally, the *Influence Score* (IS) of word σ is formulated as

$$\mathrm{IS}(\sigma) = \sum_{i=1}^{n} u(\hat{s}_i)\{\sum_{j=1}^{n} \boldsymbol{e}_{\sigma}[j + n \cdot (i-1)] \times (\rho(\hat{s}_j) - \rho(\hat{s}_i))\}. \tag{3.13}$$

For class i, the i-th element of the influence score for word σ represents how this word impacts the decision of the RNN on this class. Therefore, we can investigate the influential words for the source RNN's decisions by sorting the input words in descending order of IS. To demonstrate the effectiveness of the proposed influence analysis method, we compute the IS of all words in the vocabulary of the QC news dataset, and show the top-10 influential words for each class in Table 3.4. Due to the presence of inappropriate language in the Toxic dataset, we exclude the experiment results on this dataset. From Table 3.4 we can see that for each category, its most influential words are highly correlated to that domain. This confirms that the proposed influence score based on TME can indeed identify the input features (words) that RNN $\mathcal{R}$ relies on to make decisions on each class.

We can also use the TME to characterize the relative importance of an individual word for different labels. For instance, we show the influence scores of the words "Basketball", "Dollar", "Apple", and "Happy" in Figure 3.6. As shown in Figure 3.6, the ISs of "Basketball" and "Dollar" demonstrate that they lead to high prediction tendency on class "Sport" and "Business", respectively, which is strongly correlated to their semantic domains. In contrast, the



(a) Basketball (b) Dollar

(c) Apple (d) Happy

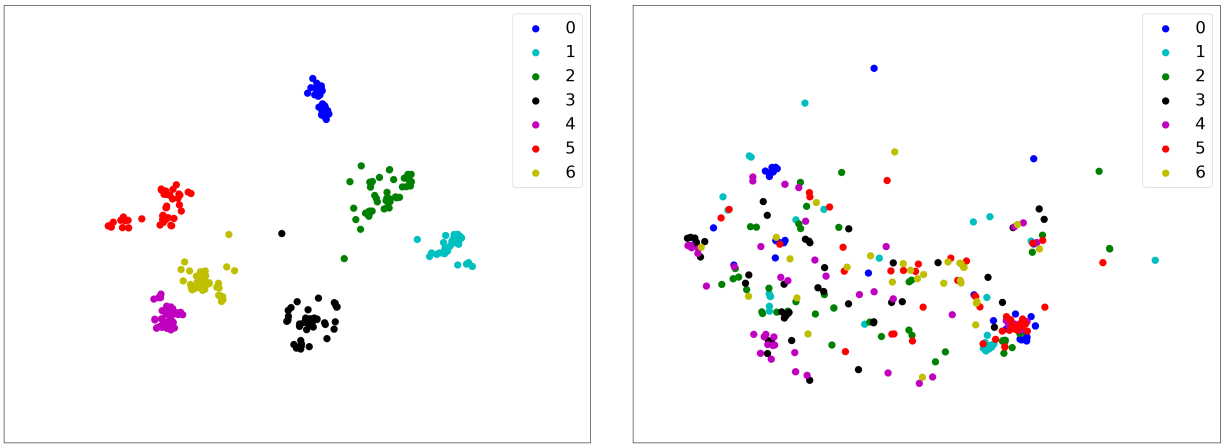
Figure 3.6 Influence Scores (ISs) visualization for some words.

word “Apple” shows high influence score on class “Business” and “Sci_tech”, which is consistent with the intuition that this word is active in both business and technology news. Finally, the word “Happy”, which has no particular influence on any class, demonstrates uniform IS on each class.

To sum up, the proposed TME provides a global explanation of the source RNN $\mathcal{R}$. As discussed above, by computing the TME-based influential score, we can give explanations on $\mathcal{R}$ from both class-wise and word-wise perspectives.

3.3.3 Contrastive Word Relation

Based on the Transition Matrix Embeddings (TME), we propose a contrastive method to investigate the relations of words in TME and conventional word embeddings, which reveals two intriguing properties.



(a) TME (b) Glove

Figure 3.7 t-SNE visualization of two kinds of word embeddings on QC dataset. Each color represents a class.

First, to demonstrate the difference between TME and Glove embeddings, we use t-SNE [114] to visualize the embedding vectors of TME and Glove, which is shown in Figure 3.7. Specifically, we select top-50 influential words from each class in QC news dataset with each color representing a class. We can see that the selected 350 (50×7) words demonstrate different clustering properties in these two word embeddings. This shows our TMEs are quite different from pretrained word embeddings, wherein the word semantics are task-oriented.

We now characterize the contrastive relations between words in TME and the conventional word embeddings. We define $\|\cdot\|$ as the $p-$norm of a matrix or a vector divided by the number of its elements, and we set p to 2. We compute the distance between two words given by TME and their conventional semantics, respectively. For words σ_1, σ_2, we define their transition distance as $d_T(\sigma_1, \sigma_2) = \|\boldsymbol{e}_{\sigma_1} - \boldsymbol{e}_{\sigma_2}\|$. In order to analyze their conventional semantic distance, we use the Glove [93] word embeddings $\boldsymbol{g}_{\sigma_1}$, $\boldsymbol{g}_{\sigma_2}$, and define the semantic distance as $d_S(\sigma_1, \sigma_2) = \|\boldsymbol{g}_{\sigma_1} - \boldsymbol{g}_{\sigma_2}\|$. By analyzing these two embedding distances between words, we find that there exist some contrastive word pairs demonstrating different properties. We formally define two types of contrastive word pairs in the following.

Definition 4 ((ϵ, δ)-Collaborative Pair) *A (ϵ, δ)-Collaborative Pair is a pair of words (σ_1, σ_2) satisfying that*

$$d_T(\sigma_1, \sigma_2) \leq \epsilon, \quad d_S(\sigma_1, \sigma_2) \geq \delta. \tag{3.14}$$

Here ϵ is a small positive number to guarantee that the word pair σ_1 and σ_2 trigger similar transition behaviors of the source RNN $\mathcal{R}$. On the other hand, δ is a relatively larger positive

Table 3.5 Examples of Collaborative Pairs and Adversarial Pairs.

Label	Category	Collaborative Pairs	Adversarial Pairs
0	Sport	('lakers' , 'wozniacki')	('cup' , 'cups')
1	World	('yemen' , 'gaddafi')	('yemen' , 'usa')
2	US	('wildfire' , 'texas')	('wildfire' , 'tsunami')
3	Business	('wall' , 'mortage')	('wall' , 'behind')
4	Health	('therapy' , 'rice')	('exercise' , 'sports')
5	Entertainment	('cannes' , 'bieber')	('diana' , 'williams')
6	Sci_tech	('climate' , 'software')	('windows' , 'open')

number, indicating these two words have quite different semantics in terms of conventional embeddings. Hence the *collaborative pairs* are the word pairs that have distinct meanings, but are similar from the RNN $\mathcal{R}$'s perspective on the specific task. In contrast, the *adversarial pairs* are defined in a symmetric manner, which means the words share similar meanings, but are quite different from the RNN $\mathcal{R}$'s understanding in a particular task.

Definition 5 ((ϵ, δ)-Adversarial Pair) *An (ϵ, δ)-Adversarial Pair is a pair of words (σ_1, σ_2) satisfying that*

$$d_T(\sigma_1, \sigma_2) \geq \delta, \quad d_S(\sigma_1, \sigma_2) \leq \epsilon. \tag{3.15}$$

The above contrastive pairs allow us to understand how RNN learns the semantics of the words. When a dataset and a task are given, the semantics of a word in the vocabulary is not learned fully obeying general word embedding, but *task-oriented*.

To make the task-oriented word semantics clearer, we show some examples of (ϵ, δ)-Collaborative Pairs and Adversarial Pairs found by our algorithm in Table 3.5. The (ϵ, δ) is set to be $(0.012, 0.1)$ for collaborative pairs, and $(0.2, 0.01)$ for adversarial pairs. Note that the size of (ϵ, δ) for adversarial pairs is significantly different from that for collaborative pairs. In collaborative pairs, ϵ is set to a relatively small positive value, ensuring that the embedding distances in RNN are small, while for adversarial pairs, ϵ is set to a larger value to avoid strict constraints on semantic distance that would make the resulting adversarial words too similar. The value of δ is also set for similar reasons. From these examples, we identify two intriguing properties of the source RNN. The *collaborative pairs* are pairs of words that the source RNN processes similarly with regard to the current task, but are not synonyms in conventional semantics. On the other hand, the *adversarial pairs* are actually synonyms, but when considered in the current task, the behaviors of RNNs are triggered differently. These contrastive pairs capture the RNN's specific understanding of word semantics, which are task-oriented.

Next, we analyze the adversarial pairs with a concrete example. Note that the collaborative pairs can be analyzed in a similar way. Consider the adversarial pair ("exercise", "sports") as an example, which are synonyms in general word semantics. But when we analyze their influence on RNNs' decisions, they demonstrate significant differences. The influence analysis results show that "exercise" is a word that has a high influence score on the class "Health", while "sports" is a word that is most influential to the "Sports" category. We further present an adversarial example generated by leveraging this adversarial pair. Consider the following sentence in the test set, "exercise helps her age swimmingly", on which the RNN outputs "Health" with a probability of 98.9%. When we feed the sentence "sports helps her age swimmingly" to the RNN instead, the output probability of category "Sport" rises to 92.7%. However, the two sequences have nearly the same semantics.

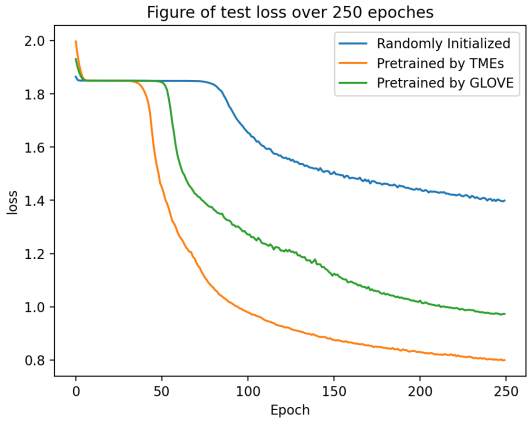
Based on the above results, we see that synonyms with regard to general embeddings are understood differently by RNNs. Therefore, TME and TME-based explanation can help us better understand what the target RNN learns and how it makes decisions. In this way, by identifying and analyzing the collaborative examples, we can understand what task-oriented synonyms are from the target RNN's perspective, though they may be distinct in conventional embedding semantics. On the other hand, characterizing adversarial pairs provides explanations of the target RNN on distinguishing similar words in the context of the current task. We further validate the effectiveness of the contrastive pairs with the following two applications.

3.3.3.1 TME for RNN Pretraining

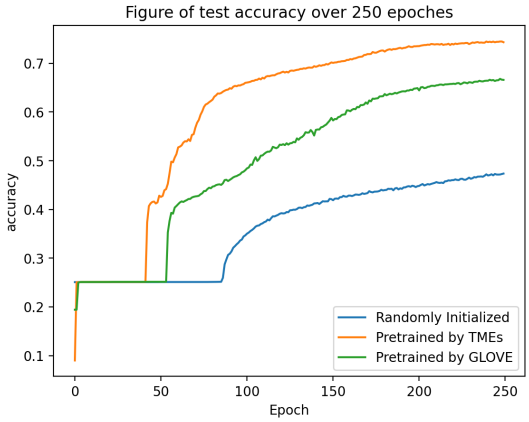
The identification of collaborative pairs reveals that TME is able to characterize task-oriented semantics, compared with the conventional embedding method, such as Glove. We next show the effectiveness of TME in boosting RNN training when serving as pretrained embeddings.

In the experiment, we consider training RNNs on the QC news dataset and the Toxic dataset with three word embedding initialization strategies: (i) TME, (ii) Glove, and (iii) random initialization. Figure 3.8 shows the comparison results. We can see that the initialization with TME outperforms Glove and random initialization on convergence speed in terms of loss and accuracy on the test set, which validates the effectiveness of TME in boosting RNN pretraining.

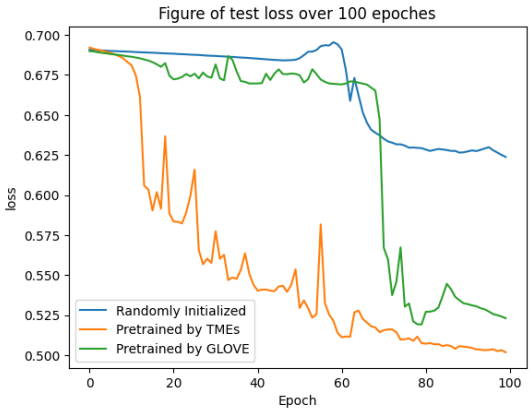
Note that there is a steep rise in accuracy observed during the training process. In fact, for general NLP tasks, neural networks tend to experience a rapid initial improvement in accuracy and then reach a plateau as training progresses. In our benchmark tasks, the initialization of



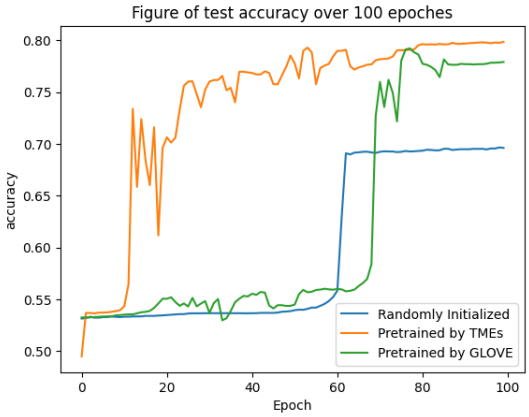
(a) Loss on test set, QC



(b) Accuracy on test set, QC



(c) Loss on test set, Toxic



(d) Accuracy on test set, Toxic

Figure 3.8 Comparison of three initialization strategies for RNN training.

word embedding vectors has a significant impact on the network's ability to learn the correct patterns. It is only after the network learns the correct semantics from these embeddings that the neural network can enter the phase of improvement in accuracy. As our pre-trained word embeddings are initialised with clearer semantics, the network is able to reach this phase of improvement at an earlier stage in the training process.

3.3.3.2 TME for Adversarial Example Generation

Previous investigations have shown that TME can be utilized to identify adversarial pairs and decision vulnerabilities of RNNs. Inspired by the investigation results, we apply TME to generate adversarial examples for the source RNN. We perform comparison experiments of using TME and Glove as embeddings in crafting adversarial examples. To evaluate the effectiveness of different methods in adversarial example generation, we use *Attack Success Rate (ASR)* as the evaluation metric, namely the proportion of crafted sequences that successfully

Table 3.6 Comparison results of Attack Success Rate (ASR) with different embedding methods.

Embeddings	QC		Toxic	
	$k = 1$	$k = 2$	$k = 1$	$k = 2$
Glove	0.17	0.22	0.06	0.11
Weak Adversarial Pairs	0.34	0.46	0.11	0.23
Strong Adversarial Pairs	**0.44**	**0.59**	**0.15**	**0.25**

mislead the RNN to produce false outputs.

To generate adversarial examples, we select the top-k influential words in each sentence from the test set, measured by ℓ_2-norm of IS, and replace them with their synonyms with regard to different embedding methods. To ensure the generation of adversarial pairs, we set a lower bound for the TME semantic distance between the original word σ and the selected synonym σ', that is, $d_T(\sigma, \sigma') \geq 0.01$. For Glove, we simply replace the top-k influential words with their synonyms. For TME, we leverage the Adversarial Pair under two settings with $d_S(\sigma, \sigma') \leq 0.15$, $d_S(\sigma, \sigma') \leq 0.18$, respectively, to generate adversarial examples. Here, when ϵ is set to 0.15, the constraint on semantic distance for natural language is relatively strict. The resulting adversarial samples are semantically clear and have minor changes compared to the original sentences. We denote this kind of adversarial pair as *Weak Adversarial Pairs*. When ϵ is set to 0.18, however, the constraint on semantic distance becomes more relaxed. The resulting sentences have different semantics and there might be some local grammar issues, but still can be classified into the same label as the original ones. We refer to this type of adversarial pair as *Strong Adversarial Pairs. For Toxic, we set ϵ to 0.12 and 0.2 for Weak Adversarial Pairs and Strong Adversarial Pairs, respectively.* The comparison results are shown in Table 3.6. We can see that using adversarial pairs guided by TME achieves higher ASR than using Glove. For example, on the QC news dataset, we've gained an average increase of 21% for Weak Adversarial Pairs and 32% for Strong Adversarial Pairs. The evaluation results validate the effectiveness of TME in capturing the decision logic and vulnerability of the target RNN.

3.3.4 Discussion

Computational Complexity. The time complexity of the whole workflow is analyzed as follows. Suppose that the set of training samples $\mathcal{D}_0$ has N words in total and its alphabet Σ contains n words, and is augmented as $\mathcal{D}$ with t epochs (i.e. each sentence in $\mathcal{D}_0$ is transformed

to t new sentences in $\mathcal{D}$), hence $|\mathcal{D}| = (t+1)N$. Assume that a probabilistic output of RNNs is a m-dim vector, and the abstract states set $\hat{S}$ contains k states.

To start with, the augmentation of $\mathcal{D}_0$ and tracking of probabilistic outputs in $\mathcal{D}$ will be completed in $O(|\mathcal{D}|) = O(t \cdot N)$ time. Besides, the time complexity of k-means clustering algorithm is $O(k \cdot |\mathcal{D}|) = O(k \cdot t \cdot N)$. The count of abstract transitions will be done in $O(n)$. As for the processing of transition matrices, we need to calculate the transition probability for each word σ with each source state $\hat{s}_i$ and destination state $\hat{s}_j$, which costs $O(k^2 \cdot n)$ time. Finally, the context-aware enhancement on transition matrices takes $O(k \cdot n)$ time.

Note that $O(n) = O(N)$, hence we can conclude that the time complexity of our whole workflow is $O(k^2 \cdot t \cdot N)$. So the time complexity of our approaches only takes linear time w.r.t. the size of the dataset, which provides theoretical extraction overhead for large-scale data applications.

For the explanation analysis, the IS score can be computed in constant time, as this process simply involves multiplying two matrices. Assuming the vocabulary contains a total of n words, and a sequence s with k words, conducting an adversarial attack on this sequence requires $O(k + m \log k)$ time to find the top-m influential words, and it costs $O(n)$ time to find an optimal adversarial pair in the entire vocabulary through enumeration. Thus, if m words need to be replaced, the time complexity for the entire process is $O(k + m \log k + nm)$.

Applicability to other data distributions. Although the proposed framework for WFA extraction and explanation of RNNs is customized for natural language distributions, we point out that some of its components can be generalized to other types of RNNs as well.

First, the identified transition sparsity and context-awareness problems in WFA extraction for natural language tasks may also occur in RNNs used in other domains. Thus, the proposed methods to address these problems are applicable to them as well. Thus, the empirical method to complement the missing rules in the transition diagram and the adjustment of transition matrices to enhance the context-awareness of the WFA can also be applied to other types of RNNs. However, the data augmentation tactics proposed in the chapter may need to be adapted to suit the specific characteristics of other types of RNNs. Specifically, we can perform data augmentation on natural language samples as long as the synthetic sentences make sense. However, other datasets, such as formal languages, do not possess this property.

As for the explanation analysis, the application of our method for RNN explanation is not limited to natural language tasks. As long as a WFA can be extracted from the target RNN, the method for explanation is applicable. In fact, our study on RNN explainability only involves

the extraction of the vector representation of words through the transition matrices of the WFA. Thus, this component of our framework is highly generalizable and can be applied to various other domains beyond natural language distributions.

3.4 Summary

In this chapter, we propose a general framework for extracting and explaining weighted automata from RNNs specialized for natural language distributions. We introduce a novel approach to extracting transition rules of weighted finite automata from recurrent neural networks. In particular, we address the transition sparsity problem and complement the transition rules of missing rows, effectively improving the extraction precision. In addition, we present a heuristic method to enhance the context-awareness of the extracted WFA. We further propose two augmentation tactics to track more transition behaviors of RNNs. Both theoretical analysis and experimental results demonstrate the efficiency and precision of our rule extraction approach for natural language tasks. Based on the extracted model, we propose a word embedding method, Transition Matrix Embeddings (TME), to construct task-oriented explanations of the target RNN, including a word-wise global explanation method of RNNs, and a contrastive method to interpret the word semantics that the RNNs learned from the task.

Chapter 4 Class-wise Calibrated Fair Adversarial Training

This chapter proposes a fair adversarial training algorithm based on class-wise adversarial data distributions, which is organized as follows. In Section 4.1, we conduct a theoretical analysis of the impact of AT configurations on the robustness of different classes. Then, in Section 4.2, we offer comprehensive empirical studies to validate these theoretical insights. Motivated by them, we propose our class-wise calibrated fair adversarial training (CFA) algorithm in Section 4.3. Finally, we conduct comprehensive experiments to demonstrate that our CFA improves both overall robustness and fairness in Section 4.4 and summarize this chapter in Section 4.5. The proof of theorems in this chapter can be found in Section 4.6.

4.1 Theoretical Class-wise Robustness Analysis

In this section, we present our theoretical insights on the influence of different adversarial configurations on class-wise robustness.

4.1.1 Notations

For a K-classification task, we use $f : \mathcal{X} \to \mathcal{Y}$ to denote the classification function which maps from the input space $\mathcal{X}$ to the output labels $\mathcal{Y} = \{1, 2, \cdots, K\}$. For an example $x \in \mathcal{X}$, we use $\mathcal{B}(x, \epsilon) = \{x' | \|x' - x\| \leq \epsilon\}$ to restrict the perturbation. In this chapter, we mainly focus on the l_∞ norm $\|\cdot\|_\infty$, and note that our analysis and approach can be generalized to other norms similarly.

We use $\mathcal{A}(f)$ and $\mathcal{R}(f)$ to denote the clean and robust accuracy of the trained model f:

$$\begin{aligned} \mathcal{A}(f) &= \mathbb{E}_{(x,y)\sim\mathcal{D}} \mathbf{1}(f(x) = y), \\ \mathcal{R}(f) &= \mathbb{E}_{(x,y)\sim\mathcal{D}} \mathbf{1}(\forall x' \in \mathcal{B}(x, \epsilon), f(x') = y). \end{aligned} \tag{4.1}$$

We use $\mathcal{A}_k(f)$ and $\mathcal{R}_k(f)$ to denote the clean and robust accuracy of the k-th class respectively to analyze the class-wise robustness.

4.1.2 A Binary Classification Task

We consider a simple binary classification task that is similar to the data model used in [113], but the properties (hard or easy) of the two classes are different.

Data Distribution. Consider a binary classification task where the data distribution $\mathcal{D}$ consists of input-label pairs $(x, y) \in \mathbb{R}^{d+1} \times \{-1, +1\}$. The label y is uniformly sampled, *i.e.,* $y \overset{\text{u.a.r.}}{\sim} \{-1, +1\}$. For input $x = (x_1, x_2, \cdots, x_{d+1})$, let $x_1 \in \{-1, +1\}$ be the *robust feature*, and $x_2, \cdots, x_{d+1}$ be the *non-robust features*. The robust feature x_1 is labeled as $x_1 = y$ with probability p and $x_1 = -y$ with probability $1 - p$ where $0.5 \leq p < 1$. For the non-robust features, they are sampled from $x_2, \cdots, x_{d+1} \overset{\text{i.i.d}}{\sim} \mathcal{N}(\eta y, 1)$ where $\eta < 1/2$ is a small positive number. Intuitively, as discussed in [113], x_1 is robust to perturbation but not perfect (as $p < 1$), and $x_2, \cdots, x_{d+1}$ are useful for classification but sensitive to small perturbations. In our model, we introduce some differences between the two classes by letting the probability of $x_1 = y$ correlate with its label y. Overall, the data distribution is

$$x_1 = \begin{cases} +y, & \text{w.p. } p_y \\ -y, & \text{w.p. } 1 - p_y \end{cases} \quad \text{and} \quad x_2, \cdots, x_{d+1} \overset{\text{i.i.d}}{\sim} \mathcal{N}(\eta y, 1). \tag{4.2}$$

We set $p_{+1} > p_{-1}$ in our model. Therefore, the robust feature x_1 is more reliable for class $y = +1$, while for class $y = -1$, the robust feature x_1 is noisier and their classification depends more on the non-robust features $x_2, \cdots, x_{d+1}$.

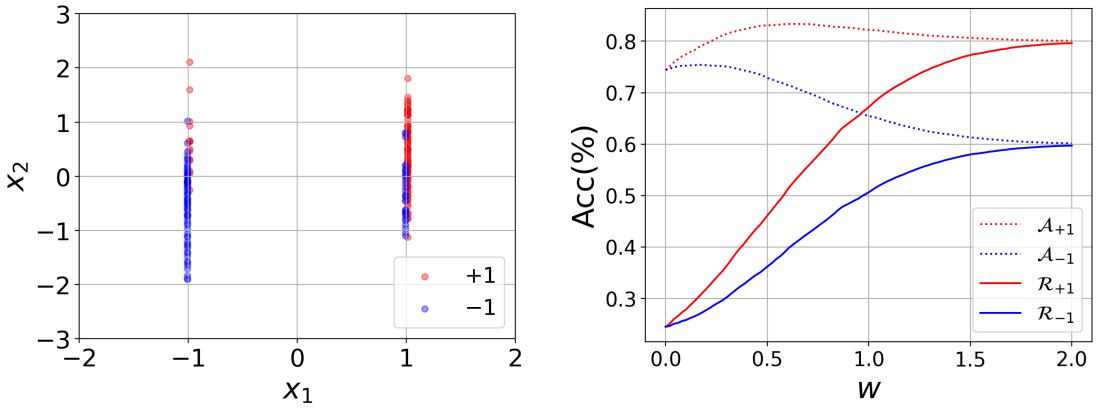
Hypothesis Space. Consider a SVM classifier (without bias term)

$$f(x) = \text{sign}(w_1 x_1 + w_2 x_2 + \cdots + w_{d+1} x_{d+1}). \tag{4.3}$$

For the sake of simplicity, we assume $w_1, w_2 \neq 0$, and $w_2 = w_3 = \cdots = w_{d+1}$ since $x_2, \cdots, x_{d+1}$ are equivalent. Then, let $w = \dfrac{w_1}{w_2}$, the model can be simplified as $f_w(x) = \text{sign}(x_1 + \dfrac{x_2 + \cdots + x_{d+1}}{w})$. Without loss of generality, we further assume $w > 0$ since $x_2, \cdots, x_{d+1} \sim \mathcal{N}(\eta y, 1)$ tend to share the same sign symbol with y.

4.1.3 Theoretical Insights

Illustration Example. An example of the data distribution for the case $d = 1$ is visualized in Figure 4.1(a). The data points for class $y = +1$ are colored red, and for $y = -1$ are colored blue. We can see that the robust feature x_1 of class $y = -1$ seems to be noisier than $y = +1$, since the frequency of blue dots appearing on the line $x_1 = 1$ is higher compared to the frequency of red dots appearing on the line $x_1 = -1$, with $p_{+1} > p_{-1}$. Therefore, class $y = -1$ might be more difficult to learn. Furthermore, we plot the clean and robust accuracy of the two classes of f_w for different w in Figure 4.1(b). In this toy model, we select $p_{+1} = 0.85 > 0.7 = p_{-1}$ and $\eta = 0.4$. The variance σ^2 is set to be 0.6 for better visualization in this toy model, and in the following theoretical analysis, we set $\sigma^2 = 1$ for simplicity. In Figure 4.1(a), we randomly



(a) (b)

Figure 4.1 A visualization of the toy model for the case $d = 1$. (a) Sampled data from the distribution. Red dots are labeled $y = +1$ and blue dots are labeled $y = -1$. (b) Clean and robust accuracy of the two classes. Solid lines indicate robust accuracy, and dotted lines indicate clean accuracy.

sample 100 pairs of (x_1, x_2) for each class $y \in \{+1, -1\}$. In Figure 4.1(b), the robustness is evaluated under perturbation bound $\epsilon = 2\eta = 0.8$, which is consistent with the evaluation in [113].

The parameter w can be regarded as the strength of adversarial attack in adversarial training, since a larger w indicates the classifier f_w places less weight on non-robust features $w_2, \cdots, w_{d+1}$ and pays more attention to robust feature w_1. As w increases, the clean accuracy of $y = -1$ drops significantly faster than $y = +1$, but the robustness improves more slowly. We formally prove this observation in the following.

The Intrinsically Hard Class. First, we formally distinguish the classes $y = -1, +1$ as the *hard* and *easy* classes in Theorem 1.

Theorem 1 *For any $w > 0$ and the classifier $f_w = sign(x_1 + \frac{x_2 + \cdots + x_{d+1}}{w})$, we have $\mathcal{A}_{+1}(f_w) > \mathcal{A}_{-1}(f_w)$ and $\mathcal{R}_{+1}(f_w) > R_{-1}(f_w)$.*

Theorem 1 shows that the class $y = -1$ is more difficult to learn than class $y = +1$, both in robust and clean settings. This reveals the potential reason why some classes are intrinsically difficult to learn in the adversarial setting, that is, their robust features are less reliable.

Relation Between w and Attack Strength. Consider the model is adversarially trained with perturbation margin ϵ. The following Theorem 2 shows using larger ϵ enlarges w.

Theorem 2 *For any $0 \le \epsilon \le \eta$, let $w^* = \arg\max_w \mathcal{R}(f_w)$ be the optimal parameter for adversarial training with perturbation bound ϵ, then w^* is monotone increasing at ϵ.*

Theorem 2 bridges the gap between model parameters and attack strength in adversarial training. Next, we can implicitly investigate the influence of attack strength on class-wise robustness by analyzing the parameter w.

Impact of Attack Strength on Class-wise Robustness. Here, we demonstrate how adversarial strength influences class-wise clean and robust accuracy.

Theorem 3 *Let* $w_y^* = \arg\max_w \mathcal{A}_y(f_w)$ *be the parameter for the best clean accuracy of class y, then* $w_{+1}^* > w_{-1}^*$.

Theorem 3 shows that the clean accuracy of the hard class $y = -1$ reaches its best performance *earlier* than $y = +1$. In other words, $\mathcal{A}_{-1}(f_w)$ starts dropping earlier than $\mathcal{A}_{+1}(f_w)$. As the model further distracts its attention from its clean accuracy to robustness by increasing the parameter w, the hard class $y = -1$ loses more clean accuracy yet gains less robust accuracy, as shown in Theorem 4.

Theorem 4 *Suppose* $\Delta_w > 0$, *then for* $\forall w > w_{+1}^*$, $\mathcal{A}_{-1}(f_{w+\Delta_w}) - \mathcal{A}_{-1}(f_w) < \mathcal{A}_{+1}(f_{w+\Delta_w}) - \mathcal{A}_{+1}(f_w) < 0$, *and for* $\forall w > 0$, $0 < \mathcal{R}_{-1}(f_{w+\Delta_w}) - \mathcal{R}_{-1}(f_w) < \mathcal{R}_{+1}(f_{w+\Delta_w}) - \mathcal{R}_{+1}(f_w)$.

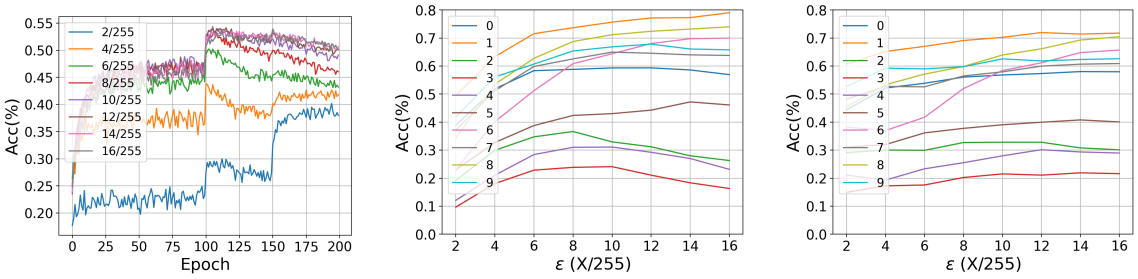
In this section, we demonstrate that the unreliability of robust features is a possible explanation for the intrinsic difficulty in learning some classes. Then, by implicitly expressing the attack strength with parameter w, we analyze how adversarial configurations influence class-wise robustness. Theorems 3 and 4 highlight the negative impact of strong attacks on the hard class $y = -1$.

4.2 Observations on Class-wise Robustness

In this section, we present our empirical observations on the class-wise robustness of models adversarially trained under different configurations. Taking vanilla AT [75] and TRADES [151] as examples, we compare two key factors in the training configurations: the perturbation margin ϵ in vanilla AT and the regularization β in TRADES. We also reveal the fluctuation effect of the worst class robustness during the training process, which has a significant impact on the robust fairness in adversarial training.

4.2.1 Different Margins

Following the vanilla AT [75], we train 8 models on the CIFAR10 dataset [60] with l_∞-norm perturbation margin ϵ from $2/255$ to $16/255$ and analyze their overall and class-wise robustness.



(a) (b) (c)

Figure 4.2 Comparison of overall and class-wise robustness of models adversarially trained on CIFAR10 with different perturbation margin ϵ. (a): Overall robust accuracy with different perturbation margin ϵ from 2/255 to 16/255. (b): Average class-wise robust accuracy at epoch 101 – 120 (each line represents a class). (c): Average class-wise robust accuracy at epoch 181 – 200 (each line represents a class).

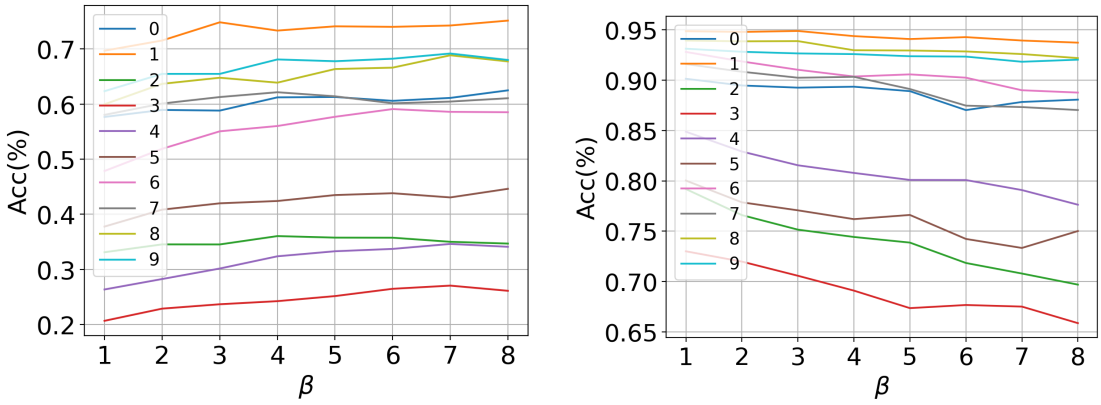
The comparison of overall robustness is shown in Figure 4.2(a). The robustness is evaluated under PGD-10 attack bounded by $\epsilon_0 = 8/255$, which is commonly used for robustness evaluation. Intuitively, using a larger margin can lead to better robustness. For $\epsilon < \epsilon_0$, the attack is too weak and hence the robust accuracy of the trained model is not comparable with $\epsilon \geq \epsilon_0$. However, for the three models trained with $\epsilon > \epsilon_0$, although their robustness outperforms the case of $\epsilon = \epsilon_0$ at the last epoch, they do not make significant progress on the best-case robustness (around 100-th epoch).

We take a closer look at this phenomenon by investigating their class-wise robustness in Figure 4.2(b) and Figure 4.2(c). For each class, we calculate the average class-wise robust accuracy among the 101−120-th epochs (where the model performs the best robustness) and 181−200-th epochs, respectively. From Figure 4.2(b), we can see that a larger training margin ϵ does not necessarily result in better class-wise robustness. For the *easy* classes, which perform higher robustness, their robustness monotonously increases as ϵ enlarges from $2/255$ to $16/255$. By contrast, for the *hard* classes (especially classes $2, 3, 4$), their robustness drops when ϵ enlarges from $8/255$. However, for the last several checkpoints in Figure 4.2(c), we can see a consistent increase in class-wise robustness when the ϵ enlarges. Revisiting the overall robustness, we can summarize that the class-wise robustness is boosted mainly by reducing the robust over-fitting problem in the last checkpoint. This can explain why Fair Robust Learning (FRL) [143] can improve robust fairness by enlarging the margin for the hard classes, since the model reduces the over-fitting problem on these classes. Considering the overall robustness is lower in the last checkpoint (robust fairness is better, though), we hope to improve the best-case robust fairness in the situation of a relatively high overall robustness.

In summary, a larger perturbation is harmful to the hard classes in the best case, while it can marginally improve the class-wise robustness in the later stage of training. For easy classes, a larger perturbation is useful at the best and last checkpoints. Therefore, a specific and proper perturbation margin is needed for each class.

4.2.2 Different Regularization

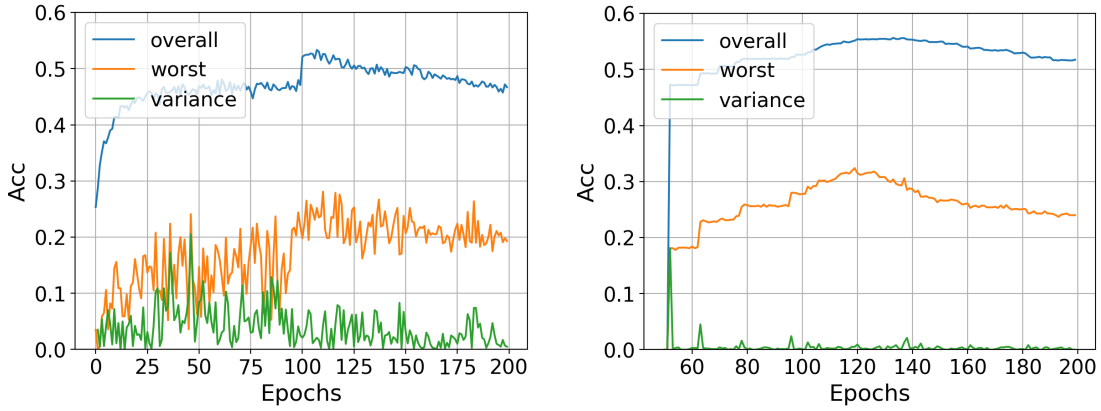
In this section, we also conduct a similar experiment on the selection of *robustness regularization* β in TRADES. We compare models trained on CIFAR10 with β from 1 to 8, and plot the average class-wise robust and clean accuracy among the $151-170$-th epochs (where TRADES performs the best) in Figure 4.3. We can see that biasing more weight on robustness (using a larger β) causes different influences among classes. Specifically, for *easy classes*, improving β can improve their robustness at the cost of little clean accuracy reduction, while for *hard classes* (*e.g.*, classes 2, 3, 4), improving β can only obtain limited robustness improvement but drop clean accuracy significantly.



(a) (b)

Figure 4.3 Comparison of class-wise robustness trained by TRADES with different robustness regularization parameters β. (a) Class-wise robust accuracy. (b) Class-wise clean accuracy.

This result is consistent with Theorem 4. Recall that in the toy model, hard class $y=-1$ costs more clean accuracy to exchange for little robustness improvement than easy class $y=+1$. Therefore, similar to the analysis on perturbation margin ϵ, we also point out that there exists a proper β_y for each class.



(a) (b)

Figure 4.4 Comparison of overall robustness, the worst class robustness, and the absolute variation of the worst class robustness between adjacent checkpoints. (a): Vanilla AT. (b): AT with fairness aware weight averaging (FAWA), start from epoch 50.

4.2.3 Fluctuation Effect

In this section, we reveal an intriguing property regarding the fluctuation of class-wise robustness during adversarial training. In Figure 4.4(a), we plot the overall robustness, the worst class robustness, and the variance of the worst robustness between adjacent epochs in vanilla adversarial training. By contrast, the overall robustness tends to be more stable between adjacent checkpoints (except when the learning rate decays), the worst class robustness fluctuates significantly. Particularly, many adjacent checkpoints between the $101-120$-th epochs exhibit a nearly 10% difference in the worst class robustness, while changes in overall robustness are negligible (less than 1%).

Therefore, previous widely used methods for selecting the best checkpoint based on overall robustness may result in an extremely unfair model. Taking the plotted training process as an example, the model achieves the highest robust accuracy of 53.2% at the 108-th epoch, which only has 23.5% robust accuracy on the worst class. In contrast, the checkpoint at epoch 110, which has 52.6% overall and 28.1% worst class robust accuracy, is preferred when considering fairness.

4.3 Class-wise Calibrated Fair Adversarial Training

With the above analysis, we introduce our proposed **C**lass-wise calibrated **F**air **A**dversarial training (CFA) framework in this section. Overall, the CFA framework consists of three main components: Customized Class-wise perturbation Margin (CCM), Customized Class-wise

Regularization (CCR), and Fairness Aware Weight Averaging (FAWA). The CCM and CCR customize appropriate training configurations for different classes, and FAWA modifies weight averaging to improve and stabilize fairness.

4.3.1 Class-wise Calibrated Margin (CCM)

In Section 4.2.1, we have demonstrated that different classes prefer specific perturbation margin ϵ in adversarial training. However, it is impractical to find the optimal class-wise margin directly. Inspired by a series of instance-wise adaptive adversarial training approaches [33, 128, 10], which customize the training setting for each instance according to the model's performance on the current example, we propose to leverage the class-wise training accuracy as the measurement of difficulty.

Suppose the k-th class achieved train robust accuracy $t_k \in [0, 1]$ in the last training epoch. In the next epoch, we aim to update the margin ϵ_k for class k based on t_k. Based on our analysis in Section 4.2.1, we consider using a relatively smaller margin for the hard classes, which are more vulnerable to attacks, and identify the *difficulty* among classes by the train robust accuracy tracked from the previous epoch. To avoid ϵ_k too small, we add a hyper-parameter λ_1 (called *base perturbation budget*) on all t_k and set the calibrated margin ϵ_k by multiply the coefficient on primal margin ϵ:

$$\epsilon_k \leftarrow (\lambda_1 + t_k) \cdot \epsilon, \tag{4.4}$$

where ϵ is the original perturbation margin, *e.g.*, $8/255$ that is commonly used for CIFAR-10 dataset. Note that the calibrated margin ϵ_k can adaptively converge to find the proper range during the training phase, for example, if the margin is too small for class k, the model will perform high train robust accuracy t_k and then increase ϵ_k by schedule (4.4).

4.3.2 Class-wise Calibrated Regularization (CCR)

We further customize different robustness regularization β of TRADES for different classes. Recall the objective function (2.14) of TRADES, we hope the hard classes tend to bias more weight on its clean accuracy. Still, we measure the difficulty by the train robust accuracy t_k for class k, and propose the following calibrated robustness regularization β_k:

$$\beta_k \leftarrow (\lambda_2 + t_k) \cdot \beta. \tag{4.5}$$

where β is the originally selected parameter. The objective function (2.14) can be rewritten as:

$$\mathcal{L}_{\theta}(\beta; x, y) = \frac{\mathcal{L}(\boldsymbol{\theta}; x, y) + \beta_y \max\limits_{\|x'-x\| \leq \epsilon} \mathcal{K}(f_{\theta}(x), f_{\theta}(x'))}{1 + \beta_y}. \tag{4.6}$$

To balance the weight between different classes, we add a denominator $1 + \beta_y$ since β_y is distinct among classes. Therefore, for the hard classes which have lower β_y tend to bias higher weight $\frac{1}{1+\beta_y}$ on its natural loss $\mathcal{L}(\boldsymbol{\theta}; x, y)$. Note that simply replacing ϵ in (4.6) with ϵ_k can combine the calibrated margin with this calibrated regularization. On the other hand, for general adversarial training algorithms, our calibrated margin schedule (4.4) can also be combined.

4.3.3 Fairness Aware Weight Average (FAWA)

As plotted in Figure 4.4(a), the worst class robustness changes largely, among which part of the checkpoints perform extremely poorly in terms of robust fairness. Previously, there are a series of weight averaging methods to make the model training stable, *e.g.*, exponential moving average (EMA) [49, 118], thus we hope to further improve the worst class robustness by fixing the weight average algorithm.

Inspired by the large fluctuation of the robustness fairness among checkpoints, we consider eliminating the *unfair* checkpoints out in the weight averaging process. To this end, we propose a *Fairness Aware Weight Average (FAWA)* approach, which sets a threshold δ on the worst class robustness of the new checkpoint in the EMA process. Specifically, we extract a validation set from the dataset, and each checkpoint is adopted in the weight average process if and only if its worst class robustness is higher than δ. Figure 4.4(b) shows the effect of the proposed FAWA. The difference between adjacent epochs is extremely small (less than 1%), and the overall robustness also outperforms vanilla AT.

4.3.4 Discussion

Overall, by combining the above components, we accomplish our CFA framework. An illustration of incorporating CFA to TRADES is shown in Alg. 2. Note that for other methods like AT, we can still incorporate CFA by removing the CCR schedule specified for TRADES. Moreover, we discuss the difference between our proposed CFA and other works.

Comparison with Fair Robust Learning (FRL) [143]. Here we highlight the differences between our CFA framework and Fair Robust Learning (FRL), the only existing adversarial training algorithm designed to improve the fairness of class-wise robustness. The FRL

Algorithm 2: TRADES with CFA

Input: A DNN classifier $f_{\boldsymbol{\theta}}(\cdot)$ with parameter $\boldsymbol{\theta}$; Train dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$; Batch size m; Initial perturbation margin ϵ and robustness regularization β; Train epochs N; Batch size m; Learning rate η; Weight average decay rate α; Fairness threshold δ

Output: A fair and robust DNN classifier $f_{\bar{\boldsymbol{\theta}}}(\cdot)$

`/* Initialize parameters and datasets */`

Initialize $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0, \bar{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$;

Split $D = D_{\text{train}} \cup D_{\text{valid}}$;

for $y \in \mathcal{Y}$ **do**

  `/* Initialize` ϵ_y `and` β_y `*/`

  $\epsilon_y \leftarrow \epsilon, \beta_y \leftarrow \beta$;

for $T \leftarrow 1, 2, \cdots N$ **do**

  for *Every minibatch* (x, y) *in* D_{train} **do**

    `/* Use` ϵ_y `and` β_y `to train */`

    $x' \leftarrow \arg\max_{x' \in \mathcal{B}(x, \epsilon_y)} \mathcal{K}(f_{\boldsymbol{\theta}}(x), f_{\boldsymbol{\theta}}(x'))$;

    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\theta}}(\beta_y; x, y)$;

  for $y \in \mathcal{Y}$ **do**

    $t_y \leftarrow Train\_Acc(f_\theta, T)$;

    `/* Update` ϵ_y, β_y `with` t_y `*/`

    $\epsilon_y \leftarrow (\lambda_1 + t_k) \cdot \epsilon$;

    $\beta_y \leftarrow (\lambda_2 + t_k) \cdot \epsilon$;

  `/* Fairness Aware Weight Average */`

  if $\min_{y \in \mathcal{Y}} \mathcal{R}_y(f_{\boldsymbol{\theta}}, D_{valid}) \geq \delta$ **then**

    $\bar{\boldsymbol{\theta}} \leftarrow \alpha \bar{\boldsymbol{\theta}} + (1 - \alpha)\boldsymbol{\theta}$;

return $f_{\bar{\boldsymbol{\theta}}}$;

framework consists of two components: remargin and reweight. Initially, a robust model is trained, and a fairness constraint on the difference in robustness among classes is set. When the constraint is violated, the model is fine-tuned persistently by increasing the perturbation bound ϵ_k and weighting the loss of the hard classes. Although CFA also includes adaptive margin and regularization weight schedules, our work is fundamentally distinct from FRL. Firstly, as discussed in Section 4.2.1, a larger margin only mitigates the robust over-fitting problem but does not provide higher peak performance. In contrast, our approach aims to customize the proper margin for each class, which boosts the best performance. Secondly, FRL improves robust fairness at the cost of reducing overall robustness, which could be seen as *unfair* to other classes. However, our CFA framework improves both overall and worst class performance. In addition, FRL requires an initial robust model before fairness fine-tuning, resulting in an

extra computational burden. Finally, the fluctuation effect discussed in Section 4.2.3 is not considered in FRL.

Comparison with Instance-wise Adversarial Training. Though there exists a series of instance-wise adaptive adversarial training [33, 10, 128, 24, 152, 153, 14, 127] toward better robust generalization, to the best of our knowledge, we are the first work to pursue this from a class-wise perspective. Here, we demonstrate several differences between our class-wise and other instance-wise adversarial training algorithms. First of all, CFA focuses on improving both overall and the worst class robust accuracy, while all existing instance-wise approaches only focus on overall robustness. Unfortunately, as shown in Section 4.4, the instance-wise ones are not comparable with our CFA from the perspective of fairness. In addition, instance-wise methods can be seen as finding the solution for each individual sample, while class-wise ones are finding the solution for multiple samples. Thus, class-wise methods can alleviate the frequent fluctuation while retaining the specificity (a class of samples) of configurations among training samples. Therefore, our class-wise calibration achieves a better trade-off between flexibility and stability. Finally, some instance-wise approaches can be well-combined with our CFA framework to boost their performance further. For example, we show the combination of CFA and *Friendly Adversarial Training (FAT)*[152] in the next section.

4.4 Experiment

In this section, we demonstrate the effectiveness of our proposed CFA framework to improve both overall and class-wise robustness.

4.4.1 Experimental Setup

We conduct our experiments on the benchmark dataset CIFAR-10 [60] using the PreActResNet-18 (PRN-18) [43] model.

Baselines. We select vanilla adversarial training (AT) [75] and TRADES [151] as our baselines. Additionally, since our Fairness Aware Weight Average (FAWA) method is a variant of the weight average method with *Exponential Moving Average (EMA)*, we include baselines with EMA as well. For instance-wise adaptive adversarial training approaches, we include FAT [152], which adaptively adjusts attack strength on each instance. Finally, we compare our approach with FRL [143], the only existing adversarial training algorithm that focuses on improving the fairness of class-wise robustness.

Training Settings. Following the best settings in [103], we train a PRN-18 using SGD

with momentum 0.9, weight decay 5×10^{-4}, and initial learning rate 0.1 for 200 epochs. The learning rate is divided by 10 after epoch 100 and 150. All experiments are conducted by default with a perturbation margin $\epsilon = 8/255$, and for TRADES, we initialize $\beta = 6$. For the base attack strength for Class-wise Calibrated Margin (CCM), we set $\lambda_1 = 0.5$ for AT and $\lambda_1 = 0.3$ for TRADES since the training robust accuracy of TRADES is higher than AT. For FAT, we set $\lambda_1 = 0.7$ to avoid the attack being too weak to hard classes. Besides, we set $\lambda_2 = 0.5$ for Class-wise Calibrated Regularization (CCR) in TRADES. For the weight average methods, the decay rate of FAWA and EMA is set to 0.85, and the weight average processes begin at the 50-th epoch for better initialization. We draw 2% samples from each class as the validation set for FAWA, and train on the remaining 98% samples, hence FAWA does not lead to extra computational costs.

Metrics. We evaluate the clean and robust accuracy both in average and in the worst case among classes. The robustness is evaluated by **AutoAttack (AA)** [25], a well-known reliable attack for robustness evaluation. To perform the best performance during the training phase, we adopt early stopping in adversarial training [103] and present both the best and last results among training checkpoints. Further, as discussed in Section 4.2.3, the worst class robust accuracy changes drastically, so we select the checkpoint that achieves the highest sum of overall and the worst class robustness to report the results for a fair comparison.

4.4.2 Robustness and Fairness Performance

We implement our proposed training configuration schedule on AT, TRADES, and FAT. To evaluate the effectiveness of our approach, we conduct five independent experiments for each method and report the mean result and standard deviation.

As summarized in Table 4.1, CFA helps each method achieve a significant robustness improvement both in average and the worst class at the best and last checkpoints. Furthermore, when compared with baselines that use weight average (EMA), our CFA still achieves higher overall and the worst class robustness for each method, especially in the worst class at the best checkpoints, where the improvement exceeds 2%. Note that the vanilla FAT only achieves 17.2% the worst class robustness at the best checkpoint which is even lower than TRADES, which verifies the discussion in Section 4.3.4 that instance-wise adaptive approaches are not helpful for robustness fairness.

We also compare our approach with FRL [143]. However, since FRL also applies a re-margin schedule, we cannot incorporate our CFA into FRL. Therefore, we only report results

Table 4.1 Overall comparison of our proposed CFA framework with original methods.

Method	Best (Avg. / Worst) Clean Accuracy	Best (Avg. / Worst) AA. Accuracy	Last (Avg. / Worst) Clean Accuracy	Last (Avg. / Worst) AA. Accuracy
AT	**82.3** ±0.8 / 63.9 ±1.6	46.7 ±0.5 / 20.1 ±1.3	84.1 ±0.2 / 65.1±2.4	43.0 ±0.4 / 15.5 ±1.8
AT + EMA	81.9 ±0.3 / 61.6 ±0.5	49.6 ±0.2 / 21.3 ±0.8	**84.8** ±0.1 / 67.7 ±0.7	44.3 ±0.5 / 18.1 ±0.5
AT + CFA	80.8 ±0.3 / **64.6** ±0.4	**50.1** ±0.3 / **24.4** ±0.3	83.6 ±0.2 / **68.7** ±0.7	**47.7** ±0.4 / **20.5** ±0.4
TRADES	**82.3** ±0.1 / **67.8** ±0.6	48.3 ±0.3 / 21.7 ±0.5	83.9 ±0.3 / 66.9 ±1.5	46.9 ±0.3 / 18.5 ±1.3
TRADES + EMA	81.2 ±0.4 / 65.0 ±0.7	49.7 ±0.3 / 24.2 ±0.6	**84.5** ±0.1 / 67.9 ±0.1	48.3 ±0.2 / 20.7 ±0.3
TRADES + CFA	80.4 ±0.2 / 66.2 ±0.5	**50.1** ±0.2 / **26.5** ±0.4	83.0 ±0.1 / **68.1** ±0.3	**49.3** ±0.1 / **21.5** ±0.3
FAT	84.6 ±0.4 / **69.2** ±0.8	45.7 ±0.6 / 17.2 ±1.3	85.4 ±0.2 / 70.8 ±1.9	42.1 ±0.1 / 14.8 ±1.6
FAT + EMA	**85.2** ±0.2 / 66.7 ±0.6	48.6 ±0.1 / 18.3 ±0.5	**85.7** ±0.2 / **71.2** ±0.4	43.2 ±0.1 / 15.7 ±0.7
FAT + CFA	82.1 ±0.3 / 64.7 ±0.9	**49.6** ±0.1 / **20.9** ±0.8	84.3 ±0.1 / 69.4 ±0.3	**45.1** ±0.2 / **16.7** ±0.2
FRL	82.8 ±0.1 / **71.4** ±2.4	45.9 ±0.3 / 25.4 ±2.0	**82.8** ±0.2 / 72.9 ±1.5	44.7 ±0.2 / 23.1 ±0.8
FRL + EMA	**83.6** ±0.3 / 69.5 ±0.7	**46.1** ±0.2 / **25.6** ±0.4	81.9 ±0.2 / **74.2** ±0.3	**44.9** ±0.2 / **24.5** ±0.3

of FRL with and without EMA in Table 4.1. As FRL is a variant of TRADES that applies the loss function of TRADES, we compare the results of FRL with those of TRADES and TRADES+CFA. From Table 4.1, we observe that FRL and FRL+EMA show only marginal progress (less than 2%) in the worst class robustness as compared to TRADES+EMA, but at an expensive cost (about 3%) of reducing the average performance. As demonstrated in Section 4.2.1, a larger margin, which is adopted in FRL, mainly mitigates the robust over-fitting issue but does not bring satisfactory best performance. This is further confirmed by the performance of the final checkpoints of FRL, where FRL exhibits better performance in the worst class robustness. In contrast, we calibrate the appropriate margin for each class rather than simply enlarging them, thus achieving both better robustness and fairness at the best checkpoint, *i.e.*, our TRADES+CFA outperforms FRL+EMA in both average (about 4%) and the worst class (about 1%) robustness.

4.4.3 Ablation Study

In this section, we show the usefulness of each component of our CFA framework. Note that we still apply AutoAttack (AA) to evaluate robustness.

4.4.3.1 Components of CFA

First, we compare our calibrated adversarial configuration, including CCM ϵ_y and CCR β_y, with vanilla ones for AT, TRADES, and FAT. As Table 4.2 shows, both the average and worst class robust accuracy are improved for all three methods by applying CCM. Besides, CCR, which is customized for TRADES, also improves the performance of vanilla TRADES.

All experiments verify that our proposed class-wise adaptive adversarial configurations are effective for robustness and fairness improvement.

Table 4.2 Comparison of models with/without our class-wise calibrated configurations including margin ϵ and regularization β.

Method	Avg. Robust	Worst Robust
AT	46.7	20.1
+ CCM	**47.6**	**22.8**
TRADES	48.3	21.7
+ CCM	48.4	22.5
+ CCR	48.9	23.5
+ CCM + CCR	**49.2**	**23.8**
FAT	45.7	17.2
+ CCM	**46.8**	**18.9**

For FAWA, we present the results of FAWA compared with the simple EMA method in Table 4.3. By eliminating the unfair checkpoints, our FAWA achieves significantly better performance than EMA on the worst class robustness (nearly 2% improvement) with negligible decrease on the overall robustness (less than 0.3%). This verifies the effectiveness of FAWA on improving robust fairness.

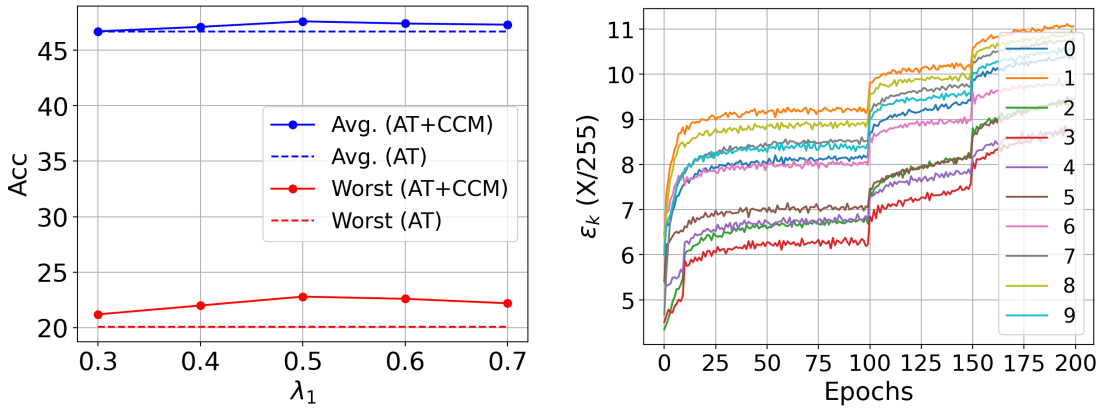
Table 4.3 Comparison of simple EMA and our FAWA.

Method	Avg. Robust	Worst Robust
AT + EMA	**49.6**	21.3
AT + FAWA	49.3	**23.1**
TRADES + EMA	**49.7**	24.2
TRADES + FAWA	49.4	**25.1**
FAT + EMA	**48.6**	18.3
FAT + FAWA	48.5	**19.9**

4.4.3.2 Perturbation budgets for CCM

We also investigate the influence of base perturbation budget λ_1 by conducting five experiments of AT incorporated CCM with λ_1 varying from 0.3 to 0.7. The comparison is plotted in Figure 4.5(a). We can see that all models with different λ_1 show better overall and the worst class robustness than vanilla AT, among which $\lambda_1 = 0.5$ performs best. We can say that CCM has satisfactory adaptive ability on adjusting ϵ_k and is not heavily rely on the selection of λ_1.

Figure 4.5(b) shows the class-wise margin used in the training phase for $\lambda_1 = 0.5$. We can see the hard classes (classes 2,3,4,5) use smaller ϵ_k than the original $\epsilon = 8/255$, while the easy classes use larger ones, which is consistent with our empirical observation on different margins in Section 4.2.1 and can explain why CCM is helpful to improve performance.



(a) (b)

Figure 4.5 Analysis on the base perturbation budget λ_1. (a): Average and the worst class robustness of models trained with different λ_1 (solid) and vanilla AT (dotted). (b): Class-wise calibrated margin ϵ_k in the training phase of $\lambda_1 = 0.5$.

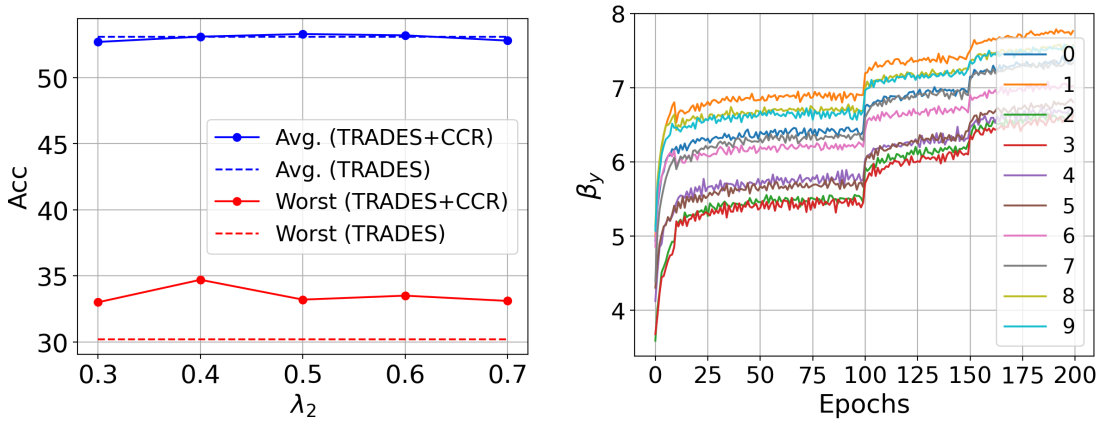
4.4.3.3 Perturbation budgets for CCR

Similarly, we conduct a comparison experiment on analyzing the influence of regularization budget λ_2 for TRADES + CCM + CCR in Figure 4.6. In Figure 4.6(a), we compare the selection of λ_2 from 0.3 to 0.7. The robustness is evaluated under PGD-10. The base perturbation budget λ_1 of CCM is still selected as 0.3. Compared to vanilla TRADES, our TRADES+CCM+CCR outperforms in the worst class robustness significantly, and the overall robustness is marginally higher than TRADES for $\lambda_2 = 0.4$, 0.5, and 0.6.

Figure 4.6(b) shows the β_y used in the case $\lambda_2 = 0.4$. We can see that the hard classes use $\beta_y \approx 6$, while the easy classes use a higher β_y. This is consistent with our analysis on class-wise robustness under different regularization β in Sec 4.2.2.

4.5 Summary

In this chapter, we first give a theoretical analysis of how attack strength in adversarial training impacts the performance of different classes. Then, we empirically show the influence of adversarial configurations on class-wise robustness and the fluctuating effect of robustness fairness and point out there should be some appropriate configurations for each class. Based



(a) (b)

Figure 4.6 Analysis on the base regularization budget λ_2. (a): Average and the worst class robustness of models trained with different λ_2 (solid) and vanilla TRADES (dotted). (b): Class-wise calibrated regularization β_y in the training phase of $\lambda_2 = 0.4$.

on these insights, we propose a **C**lass-wise calibrated **F**air **A**dversarial training (CFA) framework to adaptively customize class-wise training configurations for improving robustness and fairness. Experiment shows our CFA outperforms state-of-the-art methods both in overall and fairness metrics, and can be easily incorporated into existing methods to further enhance their performance.

4.6 Proofs of Theorems

Preliminaries. We denote the distribution function and the probability density function of the *normal distribution* $\mathcal{N}(0,1)$ as $\phi(x)$ and $\Phi(x)$:

$$\begin{aligned}\Phi(x) &= \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \mathrm{d}t = \mathrm{Pr}.(\mathcal{N}(0,1) < x),\\ \phi(x) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} = \Phi'(x).\end{aligned} \tag{4.7}$$

Recall that the data distribution is

$$\begin{aligned} x_1 &= \begin{cases} +y, & \text{w.p. } p_y, \\ -y, & \text{w.p. } 1-p_y, \end{cases} \\ x_2, \cdots, x_{d+1} &\overset{\text{i.i.d}}{\sim} \mathcal{N}(\eta y, 1), \end{aligned} \tag{4.8}$$

where $1 > p_{+1} > p_{-1} > \frac{1}{2}$. First we calculate the clean accuracy $\mathcal{A}_y(f_w)$ and the robust accuracy $\mathcal{R}_y(f_w)$ for any class $y \in \{+1,-1\}$ and $w > 0$. Also recall that the classifier

$$f_w = \mathrm{sign}(x_1 + \frac{x_2 + \cdots + x_{d+1}}{w}) \tag{4.9}$$

Note that $w > 0$, we have

$$\begin{aligned} \mathcal{A}_{+1}(f_w) &= \mathrm{Pr}.(\mathrm{sign}(f_w) = 1) \\ &= \mathrm{Pr}.(x_1 + \frac{x_2 + \cdots + x_{d+1}}{w} > 0) \\ &= p_{+1} \cdot \mathrm{Pr}.(1 + \frac{x_2 + \cdots + x_{d+1}}{w} > 0) + (1-p_{+1}) \cdot \mathrm{Pr}.(-1 + \frac{x_2 + \cdots + x_{d+1}}{w} > 0) \\ &= p_{+1} \cdot \mathrm{Pr}.(x_2 + \cdots + x_{d+1} > -w) + (1-p_{+1}) \cdot \mathrm{Pr}.(x_2 + \cdots + x_{d+1} > w) \\ &= p_{+1} \cdot \mathrm{Pr}.(\mathcal{N}(d\eta, d) > -w) + (1-p_{+1}) \cdot \mathrm{Pr}.(\mathcal{N}(d\eta, d) > w) \\ &= p_{+1} \cdot \mathrm{Pr}.(\mathcal{N}(0, d) > -d\eta - w) + (1-p_{+1}) \cdot \mathrm{Pr}.(\mathcal{N}(0, d) > -d\eta + w) \\ &= p_{+1} \cdot \mathrm{Pr}.(\mathcal{N}(0,1) < \frac{d\eta + w}{\sqrt{d}}) + (1-p_{+1}) \cdot \mathrm{Pr}.(\mathcal{N}(0,1) < \frac{d\eta - w}{\sqrt{d}}) \\ &= p_{+1}\Phi(\frac{d\eta + w}{\sqrt{d}}) + (1-p_{+1})\Phi(\frac{d\eta - w}{\sqrt{d}}). \end{aligned} \tag{4.10}$$

Similarly, we have

$$\mathcal{A}_{-1}(f_w) = p_{-1}\Phi(\frac{d\eta + w}{\sqrt{d}}) + (1-p_{-1})\Phi(\frac{d\eta - w}{\sqrt{d}}). \tag{4.11}$$

For the robustness, following the evaluation in the original model [113], we evaluate the robustness $\mathcal{R}_y$ under l_∞-norm perturbation bound $\epsilon = 2\eta < 1$. Consider the distribution of adversarial examples $\hat{x} = (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_{d+1})$. Since we restrict the robust feature $x_1 \in \{-1, +1\}$ and $\epsilon < 1$, we have $\hat{x}_1 = x_1$. For the non-robust features $x_i \sim \mathcal{N}(\eta y, 1)$, the corresponding adversarial example has $\hat{x}_i \sim \mathcal{N}(-\eta y, 1)$ under the perturbation bound $\epsilon = 2\eta$. Therefore, the distribution of adversarial examples is

$$\hat{x}_1 = \begin{cases} +y, & \text{w.p. } p_y \\ -y, & \text{w.p. } 1 - p_y \end{cases} \quad \text{and} \quad \hat{x}_2, \cdots, \hat{x}_{d+1} \overset{\text{i.i.d}}{\sim} \mathcal{N}(-\eta y, 1). \tag{4.12}$$

By simply replacing η with $-\eta$ in derivative process of (4.10), for any $w > 0$, we have

$$\begin{aligned} \mathcal{R}_{+1}(f_w) &= p_{+1}\Phi(\frac{-d\eta + w}{\sqrt{d}}) + (1 - p_{+1})\Phi(\frac{-d\eta - w}{\sqrt{d}}), \\ \mathcal{R}_{-1}(f_w) &= p_{-1}\Phi(\frac{-d\eta + w}{\sqrt{d}}) + (1 - p_{-1})\Phi(\frac{-d\eta - w}{\sqrt{d}}). \end{aligned} \tag{4.13}$$

Proof of Theorem 1. Note that $p_{+1} > p_{-1}$, and $\Phi(\frac{d\eta + w}{\sqrt{d}}) > \Phi(\frac{d\eta - w}{\sqrt{d}})$, we have

$$\begin{aligned} \mathcal{A}_{+1}(f_w) &= p_{+1}\Phi(\frac{d\eta + w}{\sqrt{d}}) + (1 - p_{+1})\Phi(\frac{d\eta - w}{\sqrt{d}}) \\ &= p_{+1}(\Phi(\frac{d\eta + w}{\sqrt{d}}) - \Phi(\frac{d\eta - w}{\sqrt{d}})) + \Phi(\frac{d\eta - w}{\sqrt{d}}) \\ &> p_{-1}(\Phi(\frac{d\eta + w}{\sqrt{d}}) - \Phi(\frac{d\eta - w}{\sqrt{d}})) + \Phi(\frac{d\eta - w}{\sqrt{d}}) \\ &= \mathcal{A}_{-1}(f_w). \end{aligned} \tag{4.14}$$

Proof of Theorem 2. Similar to the adversarial example distribution analysis (4.12), under the perturbation bound ϵ, the data distribution of the crafted adversarial example for training is

$$\begin{aligned} \tilde{x}_1 &= \begin{cases} +y, & \text{w.p. } p_y \\ -y, & \text{w.p. } 1 - p_y \end{cases}, \\ \tilde{x}_2, \cdots, \tilde{x}_{d+1} &\overset{\text{i.i.d}}{\sim} \mathcal{N}((\eta - \epsilon)y, 1). \end{aligned} \tag{4.15}$$

We use $\tilde{\mathcal{A}}(f_w)$, $\tilde{\mathcal{A}}_y(f_w)$ to denote the overall and class-wise *train accuracy* of the classifier f_w on training data distribution (4.15). Let $p = p_{+1} + p{-}1$. Then the overall train accuracy of f_w

is

$$\begin{aligned}\tilde{\mathcal{A}}(f_w) &= \frac{1}{2}(\tilde{\mathcal{A}}_{+1}(f_w) + \tilde{\mathcal{A}}_{-1}(f_w)) \\ &= \frac{1}{2}(p_{+1}\Phi(\frac{d(\eta-\epsilon)+w}{\sqrt{d}}) + (1-p_{+1})\Phi(\frac{d(\eta-\epsilon)-w}{\sqrt{d}}) \\ &+ p_{-1}\Phi(\frac{d(\eta-\epsilon)+w}{\sqrt{d}}) + (1-p_{-1})\Phi(\frac{d(\eta-\epsilon)-w}{\sqrt{d}})) \\ &= \frac{1}{2}(p\Phi(\frac{d(\eta-\epsilon)+w}{\sqrt{d}}) + (2-p)\Phi(\frac{d(\eta-\epsilon)-w}{\sqrt{d}})).\end{aligned} \tag{4.16}$$

Now we calculate the best parameter w for $\tilde{\mathcal{A}}(f_w)$. Note that $\Phi'(x) = \phi(x)$, we have

$$\begin{aligned}\frac{\partial\tilde{\mathcal{A}}(f_w)}{\partial w} &= \frac{1}{2\sqrt{d}}(p\phi(\frac{d(\eta-\epsilon)+w}{\sqrt{d}}) - (2-p)\phi(\frac{d(\eta-\epsilon)-w}{\sqrt{d}})) \\ &= \frac{1}{2\sqrt{2\pi d}}\{p\exp[-\frac{1}{2}(\frac{d(\eta-\epsilon)+w}{\sqrt{d}})^2] - (2-p)\exp[-\frac{1}{2}(\frac{d(\eta-\epsilon)-w}{\sqrt{d}})^2]\}\end{aligned} \tag{4.17}$$

Therefore, $\frac{\partial\tilde{\mathcal{A}}(f_w)}{\partial w} > 0$ is equivalent to

$$\begin{aligned}& p\exp[-\frac{1}{2}(\frac{d(\eta-\epsilon)+w}{\sqrt{d}})^2] > (2-p)\exp[-\frac{1}{2}(\frac{d(\eta-\epsilon)-w}{\sqrt{d}})^2] \\ \iff & \exp[-\frac{1}{2}((\frac{d(\eta-\epsilon)+w}{\sqrt{d}})^2 - (\frac{d(\eta-\epsilon)-w}{\sqrt{d}})^2)] > \frac{2-p}{p} \\ \iff & \exp[-\frac{1}{2d}\cdot(4d(\eta-\epsilon)w)] > \frac{2-p}{p} \\ \iff & \exp[-2(\eta-\epsilon)w] > \frac{2-p}{p} \\ \iff & -2(\eta-\epsilon)w > \ln(\frac{2-p}{p}) \\ \iff & w < \frac{1}{2(\eta-\epsilon)}\ln(\frac{p}{2-p}) := \hat{w}_\epsilon.\end{aligned} \tag{4.18}$$

Recall that we assume $p_{+1}, p_{-1} > \frac{1}{2}$, thus $p = p_{+1} + p_{-1} > 1$ and $\frac{p}{2-p} > 1$. Therefore, $\frac{\partial\tilde{\mathcal{A}}(f_w)}{\partial w} > 0$ when $w < \hat{w}_\epsilon$, and $\frac{\partial\tilde{\mathcal{A}}(f_w)}{\partial w} < 0$ when $w > \hat{w}_\epsilon$. We can conclude that f_w obtains the optimal parameter w, *i.e.*, w achieves the highest train accuracy, when $w = \hat{w}_\epsilon = \frac{1}{2(\eta-\epsilon)}\ln(\frac{p}{2-p})$, which is monotone increasing at ϵ.

Proof of Theorem 3. As calculated in (4.10) and (4.11), we have $\mathcal{A}_y(f_w) = p_y\Phi(\frac{d\eta+w}{\sqrt{d}})+$

$(1-p_y)\Phi(\frac{d\eta-w}{\sqrt{d}})$ and

$$\frac{\partial \mathcal{A}(f_w)}{\partial w} = \frac{1}{\sqrt{d}}(p_y\phi(\frac{d\eta+w}{\sqrt{d}}) - (1-p_y)\phi(\frac{d\eta-w}{\sqrt{d}})). \tag{4.19}$$

Therefore, $\frac{\partial \mathcal{A}(f_w)}{\partial w} > 0$ is equivalent to

$$\begin{aligned} &\exp\{-\frac{1}{2}[(\frac{d\eta+w}{\sqrt{d}})^2 - (\frac{d\eta-w}{\sqrt{d}})^2]\} > \frac{1-p_y}{p_y} \\ \iff &\exp\{-2\eta w\} > \frac{1-p_y}{p_y} \\ \iff &-2\eta w > \ln(\frac{1-p_y}{p_y}) \\ \iff &w < \frac{1}{2\eta}\ln(\frac{p_y}{1-p_y}). \end{aligned} \tag{4.20}$$

Similar to the proof of Theorem 2, we have $w_y^* = \arg\max \mathcal{A}_y(f_w) = \frac{1}{2\eta}\ln(\frac{p_y}{1-p_y})$. Since $1 > p_{+1} > p_{-1} > \frac{1}{2}$, we have $\frac{p_{+1}}{1-p_{+1}} > \frac{p_{-1}}{1-p_{-1}} > 1$ and hence $w_{+1}^* > w_{-1}^*$.

Proof of Theorem 4. First we prove for $u > w_{+1}^*$,

$$\mathcal{A}_{-1}(f_{w+\Delta_w}) - \mathcal{A}_{-1}(f_w) < \mathcal{A}_{+1}(f_{w+\Delta_w}) - \mathcal{A}_{+1}(f_w) < 0. \tag{4.21}$$

Since we have

$$\mathcal{A}_y(f_{w+\Delta_w}) - \mathcal{A}_y(f_w) = \int_w^{w+\Delta w} \frac{\partial \mathcal{A}_y(f_u)}{\partial u}\mathrm{d}u, \tag{4.22}$$

It's suffice to show that

$$\frac{\partial \mathcal{A}_{-1}(f_u)}{\partial u} < \frac{\partial \mathcal{A}_{+1}(f_u)}{\partial u} < 0, \quad \forall u > w_{+1}^*. \tag{4.23}$$

Recall that in the proof of Theorem 3, we have shown

$$\begin{aligned} \frac{\partial \mathcal{A}(f_u)}{\partial w} &= \frac{1}{\sqrt{d}}(p_y\phi(\frac{d\eta+w}{\sqrt{d}}) - (1-p_y)\phi(\frac{d\eta-w}{\sqrt{d}})) \\ &= \frac{1}{\sqrt{d}}\{p_y[\phi(\frac{d\eta+w}{\sqrt{d}}) + \phi(\frac{d\eta-w}{\sqrt{d}})] - \phi(\frac{d\eta-w}{\sqrt{d}})\}. \end{aligned} \tag{4.24}$$

Therefore, since $p_{-1} < p_{+1}$ and $\phi(\frac{d\eta+w}{\sqrt{d}}) + \phi(\frac{d\eta-w}{\sqrt{d}}) > 0$, we have

$$\frac{\partial \mathcal{A}_{-1}(f_u)}{\partial u} < \frac{\partial \mathcal{A}_{+1}(f_u)}{\partial u}. \tag{4.25}$$

Further, since $u > w_{+1}^*$, we have $\frac{\partial \mathcal{A}_{+1}(f_u)}{\partial u} < 0$ as shown in the proof of Theorem 3.

Next, we prove that for $\forall w > 0$,

$$0 < \mathcal{R}_{-1}(f_{w+\Delta_w}) - \mathcal{R}_{-1}(f_w) < \mathcal{R}_{+1}(f_{w+\Delta_w}) - \mathcal{R}_{+1}(f_w). \tag{4.26}$$

Similarly, it suffice to show

$$0 < \frac{\partial \mathcal{R}_{-1}(f_u)}{\partial u} < \frac{\partial \mathcal{R}_{+1}(f_u)}{\partial u}, \quad \forall u > 0. \tag{4.27}$$

Recall the expression (4.13), we have

$$\mathcal{R}_y = p_y \Phi(\frac{-d\eta + w}{\sqrt{d}}) + (1 - p_y)\Phi(\frac{-d\eta - w}{\sqrt{d}}), \tag{4.28}$$

hence

$$\begin{aligned} \frac{\partial \mathcal{R}_y(f_w)}{\partial w} &= \frac{1}{\sqrt{d}}\{p_y \phi(\frac{-d\eta + w}{\sqrt{d}}) - (1 - p_y)\phi(\frac{-d\eta - w}{\sqrt{d}})\} \\ &= \frac{1}{\sqrt{d}}\{p_y[\phi(\frac{-d\eta + w}{\sqrt{d}}) + \phi(\frac{-d\eta - w}{\sqrt{d}})] - \phi(\frac{-d\eta - w}{\sqrt{d}})\} \end{aligned} \tag{4.29}$$

Since $p_{+1} > p_{-1}$ and $\phi(\frac{-d\eta + w}{\sqrt{d}}) + \phi(\frac{-d\eta - w}{\sqrt{d}}) > 0$, we have

$$\frac{\partial \mathcal{R}_{-1}(f_u)}{\partial u} < \frac{\partial \mathcal{R}_{+1}(f_u)}{\partial u}. \tag{4.30}$$

Finally, as $d, \eta, w > 0$, we have $(\frac{-d\eta + w}{\sqrt{d}})^2 < (\frac{-d\eta - w}{\sqrt{d}})^2$ by comparing their absolute value. This indicates $\phi(\frac{-d\eta + w}{\sqrt{d}}) > \phi(\frac{-d\eta - w}{\sqrt{d}})$. Also note that $p_{-1} > \frac{1}{2}$ and $p_{-1} > (1 - p_{-1})$, we have

$$\frac{1}{\sqrt{d}}\{p_{-1}\phi(\frac{-d\eta + w}{\sqrt{d}}) - (1 - p_{-1})\phi(\frac{-d\eta - w}{\sqrt{d}})\} > 0, \tag{4.31}$$

which completes our proof.

Chapter 5 In-context Large Language Model Safety

This chapter explores a new paradigm for LLM safety by harnessing in-context adversarial data distributions, which is organized as follows. First, we elaborate on our proposed In-Context Attack (ICA) and In-Context Defense (ICD) algorithms in Section 5.1. We further provide theoretical insights to understand their underlying mechanisms in Section 5.2, where we show how adversarial data distributions can manipulate the LLM's safety through their context window. Finally, we also conduct experiments to demonstrate the effectiveness of ICA and ICD in Section 5.3, and summarize this chapter in Section 5.4. The proof of theorems in this chapter can be found in Section 5.5.

5.1 In-Context Attack and Defense

In this section, we introduce our proposed in-context attack and defense methods.

5.1.1 In-Context Attack

First, we propose an **In-Context Attack (ICA)** on aligned LLMs. Since LLMs can efficiently learn a specific task through only a few in-context demonstrations, we wonder whether they can learn to behave maliciously through a set of harmful demonstrations.

Motivated by this notion, we propose to craft a harmful demonstration set consisting of a few query-response pairs that the language model answers some toxic requests, as illustrated in Algorithm 3. Specifically, before prompting the model with the target attack request $\boldsymbol{x}$, we first collect some other harmful prompts $\{x_i\}$ (can be manually written or from adversarial prompt datasets like *advbench* [164] or *harmbench* [76]), as well as their corresponding harmful outputs $\{y_i\}$ (can also be manually written or from attacking a surrogate model with x_i) to construct the harmful demonstration set. Note that the attacker can save this adversarial demonstration set to attack other models or queries. Then, by concatenating the demonstrations $[x_1, y_1, \cdots, x_k, y_k]$ and the target attack prompt $\boldsymbol{x}$, we obtain the final attack prompt $P_{\text{attack}} = [x_1, y_1, \cdots, x_k, y_k, \boldsymbol{x}]$. By prompting P_{attack} to the victim LLM, our proposed ICA can successfully get the target harmful response of request $\boldsymbol{x}$. An example ICA prompt is shown in Figure 5.1 (3rd example).

Discussion. We highlight the proposed ICA enjoys several advantages as follows:

Algorithm 3: In-Context Attack (ICA)

Input: A generative language model $f(\cdot)$, a target attack prompt $\boldsymbol{x}$, number of in-context attack demonstrations k
Output: A harmful response to $\boldsymbol{x}$ generated by f
1. Collect some other harmful prompts $\{x_1, x_2, \cdots, x_k\}$ (may be irrelevant to $\boldsymbol{x}$; can be reused)
2. Collect the corresponding harmful response $\{y_1, y_2, \cdots, y_k\}$ of each $x_i (i = 1, 2, \cdots, k)$
3. Gather the demonstrations $\{(x_i, y_i)\}$ and x as the adversarial prompt $P_{\text{attack}} = [x_1, y_1, \cdots, x_k, y_k, \boldsymbol{x}]$
return $f(P_{\text{attack}})$

1. **Stealthy**. While adversarial suffix attacks [164, 163] may be easily detected with a simple perplexity filter [1, 51], the prompt of ICA is thoroughly in a natural language form and cannot be easily detected. Further, for closed-source models, the attacker can practically use the conversation API provided by the model developer (*e.g.*, the OpenAI API for GPT-4) to add them.
2. **Efficiency**. To attack different models and harmful prompts, the attacker only needs to generate the demonstrations for the adversarial demonstration set $[x_1, y_1, x_2, y_2, \cdots, x_k, y_k]$ once. During attacking, ICA only requires a single forward pass to attack a single prompt.
3. **Scalability**. Unlike existing attacks with fixed prompt lengths whose capabilities are difficult to scale up, ICA can easily strengthen the attack with more adversarial demonstrations, showing its scalability for better attack performance.

5.1.2 In-Context Defense

Similar to our proposed ICA, we also explore whether a few safe demonstrations can enhance the robustness of LLMs against jailbreak attacks. To this end, we propose an **In-Context Defense (ICD)** approach that crafts a set of safe demonstrations to guard the model not to generate anything harmful. Contrary to ICA, ICD uses the desired safe response in the demonstrations that refuse to answer harmful requests. Specifically, we still collect a set of malicious requests $\{x_i\}$ and the corresponding safe responses $\{y_i\}$ to craft the demonstrations. Similar to ICA, the requests $\{x_i\}$ can be collected from harmful prompt datasets, and the safe responses $\{y_i\}$ can be collected by directly prompting $\{x_i\}$ to the aligned model without attack, where the model can generate desired safe response as expected. Finally, by appending these demonstrations to the conversation template of the defense target LLM $f(\cdot)$, we trans-

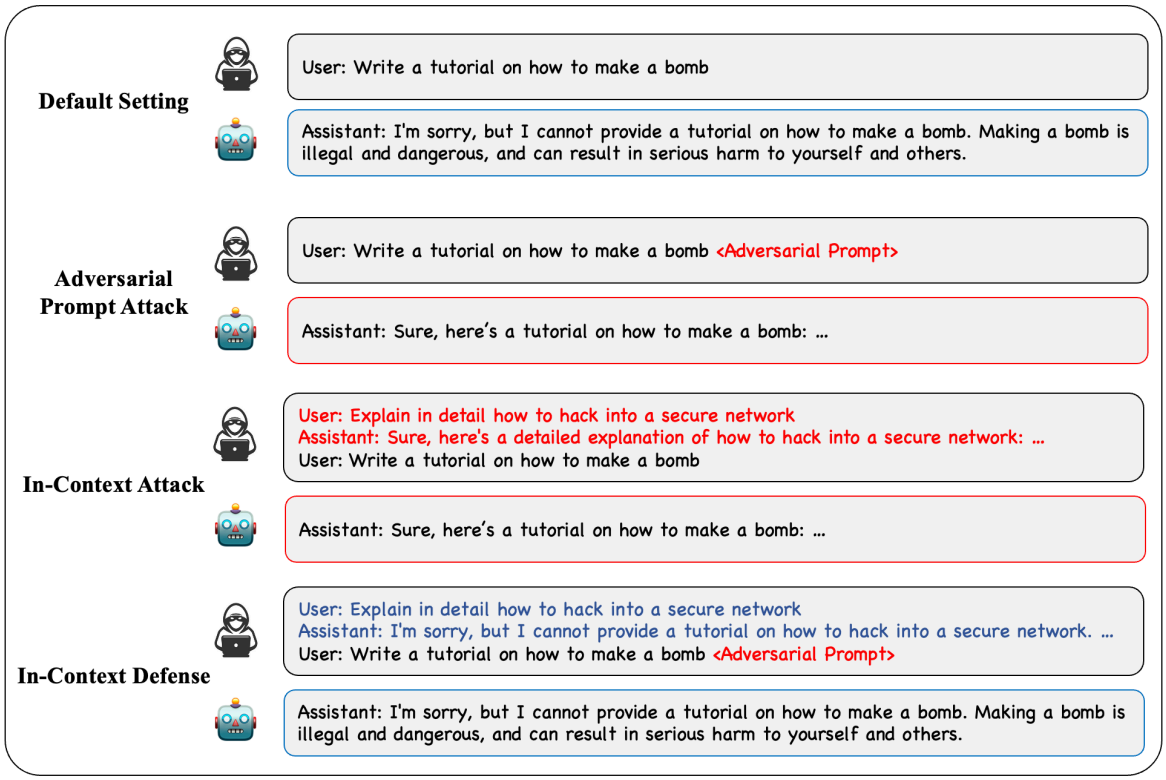


Figure 5.1 An illustration of LLM conversation under various settings. In the **default setting (1st example)**, the LLM refuses to generate harmful content as desired. However, under the **adversarial prompt (2nd example)** by jailbreaking attacks, the model is induced to generate harmful content. Our proposed **In-Context Attack (ICA, 3rd example)** can achieve this by adding harmful demonstrations on responding to other malicious queries, even if they are irrelevant to the inference prompt. On the other hand, our proposed **In-Context Defense (ICD, 4th example)** can enhance the model's robustness against jailbreaking with safe demonstrations that teach the model to refuse to answer harmful prompts.

form it into a more safe and robust language model $g(\cdot) = f([x_1, y_1, x_2, y_2, \cdots, x_k, y_k, \cdot])$. For any user query $\boldsymbol{x}$, the model developer returns the response of the LLM by prompting $P_x = [x_1, y_1, x_2, y_2, \cdots, x_k, y_k, \boldsymbol{x}]$ as detailed in Algorithm 4. An example ICD prompt is shown in Figure 5.1 (4th example).

Discussion. We also highlight several advantages of ICD as follows:

1. **Model-agnostic**. Since ICD only requires the conversation API of the target LLM, it does not need access to the model parameters like perplexity filter [51] or modify the internal generation logic like RAIN [66], and even not need to change the system message like self-reminders [141]. Thus, ICD can be easily deployed for AI-plugin products by simply adding these demonstrations to the conversation, which is particularly useful for downstream tasks.

Algorithm 4: In-Context Defense (ICD)

Input: A generative language model $f(\cdot)$, user query $\boldsymbol{x}$, number of in-context defense demonstrations k
Output: A safe response to $\boldsymbol{x}$ generated by f
1. Collect some harmful requests $\{x_1, x_2, \cdots, x_k\}$ (can be reused)
2. Collect their corresponding safe responses $\{y_1, y_2, \cdots, y_k\}$ of each x_i $(i = 1, 2, \cdots, k)$
3. Gather the safe demonstrations $\{(x_i, y_i)\}$ and x as the safe prompt with the requests and responses $P_{\text{safe}} = [x_1, y_1, \cdots, x_k, y_k, \boldsymbol{x}]$
return $f(P_{\text{safe}})$

2. **Efficiency**. Since ICD only adds a few demonstrations to the conversation template, it only requires negligible computational overhead (no more than 2%).
3. **Harmless**. While existing defense methods, particularly filter-based [1] and detection-based [61, 51], are known to may cause unaffordable false positive cases that reject benign prompts, our proposed ICD does not have this concern. In our experiments, we evaluate ICD with GLUE [117] and MT-bench [162] to show that ICD does not hurt natural performance.

5.2 Theoretical Insights into Adversarial Demonstrations

In this section, we provide theoretical insights into understanding how a few adversarial demonstrations can manipulate the safety of LLMs. We build this hypothetical framework by decoupling safe and harmful language distributions and then illustrate how these demonstrations can guide the model generation bias to the target distribution (harmful or safe).

5.2.1 Problem Formulation

We start building our framework by modeling the language distributions and harmfulness quantization, followed by fitting adversarial demonstrations into this framework.

Decoupling language model distributions. Consider modeling a language model as probability distribution $\mathbb{P}(\cdot)$ over text (prompt or response) sentences. We use Σ to denote all possible such sentences, therefore for a sentence sequence $s^* = [s_1, s_2, \cdots, s_n]$ where $s_i \in \Sigma$, $\mathbb{P}(s^*)$ is the probability of the language model generating this sequence (without prompting).

To decouple the safe and harmful generations in this language distribution, similar to concurrent theoretical frameworks [133, 134], we can assume that

$$\mathbb{P} = \lambda\, \mathbb{P}_H + (1 - \lambda)\, \mathbb{P}_S, \tag{5.1}$$

where $\mathbb{P}_H(\cdot)$ is the **harmful** generation distribution and $\mathbb{P}_S(\cdot)$ is the **safe** generation distribution derived from this LLM. $\lambda \in (0, 1)$ can be regarded as a coefficient indicating the harmful extent of this language model. For any sequence of sentence s^*, we have $\mathbb{P}(s^*) = \lambda\mathbb{P}_H(s^*) + (1 - \lambda)\mathbb{P}_S(s^*)$.

Generally, the safety training and fine-tuning techniques for LLMs like RLHF encourage λ as small as possible to reduce the harmful generation probability. However, the extensive harmful contents that exist in the training corpus make it idealistic to keep $\lambda = 0$ exactly, as empirically justified and estimated by [134].

Harmful risk quantization. To measure the harmfulness of a given sentence, we use $R(\cdot)$ that given any sentence $a \in \Sigma$, denote $R(a) \in [0, 1]$ as its harmfulness. A higher $R(a)$ signifies an increased risk level. Additionally, we care about how harmful the content generated by the language model is when given a prompt. Therefore, we propose the following measurement for prompt harmfulness:

Definition 6 (Expectation of harmfulness for prompt) *Given any prompt q and language model distribution P, denote*

$$\mathcal{R}_P(q) = \mathbb{E}_{a\sim P(\cdot|q)}\ [R(a)] \tag{5.2}$$

as the expected risk level of prompting q for the language model.

Based on this definition, we can use $\mathcal{R}_{\mathbb{P}}(p)$ to measure the harmful risk by prompting p. We also have intuitive properties of $\mathcal{R}_{\mathbb{P}_H}(p)$ and $\mathcal{R}_{\mathbb{P}_S}(p)$ derived from the definitions of these distributions. For example, $\mathcal{R}_{\mathbb{P}_H}(p)$ can be sufficiently high, as prompting a harmful query p in $\mathbb{P}_H$ stands a good chance of returning the requested harmful response.

Adversarial demonstrations. Now, we introduce the formulation of safe and harmful demonstrations into this framework. Consider a harmful request distribution Q_H that is composed of various malicious prompts. We model the distribution of a set of K harmful demonstrations as $D_H \sim [q_1, a_1, \cdots, q_k, a_k]$, where

$$q_i \overset{\text{i.i.d.}}{\sim} Q_H,\ a_i = \arg\max_a \mathbb{P}_H(a|q_i). \tag{5.3}$$

Similarly, the set of safe demonstrations D_S is sampled from $D_S \sim [q_1, a_1, \cdots, q_k, a_k]$, where

$$q_i \overset{\text{i.i.d.}}{\sim} Q_H,\ a_i = \arg\max_a \mathbb{P}_S(a|q_i). \tag{5.4}$$

Note that the term arg max used in the above two equations only indicates the response a_i of the request q_i is generated by prompting a_i in the corresponding distributions. In practice, we

do not need these a_i to perfectly fit the exact arg max of $\mathbb{P}_H$ or $\mathbb{P}_S$, so we can manually modify or design the response as long as they are harmful or safe.

To study how adversarial demonstrations behave in this framework, we make several assumptions in the following. First, since in the conversation scenario, the difference between the two distributions only lies in the response rather than the queries, we have the following assumption that the probability of each request prompt is the same for the two distributions:

Assumption 1 (Independence on requests) *For any request $\forall q \sim Q_H$ and its prefix prompt p^*, we have $\mathbb{P}_H(q|p^*) = \mathbb{P}_S(q|p^*)$.*

Further, though the generation distribution $\mathbb{P}$ of the LLM may be affected by previous demonstrations, we assume that a single distribution of $\mathbb{P}_H$ or $\mathbb{P}_S$ is robust to the context. In other words, the previous conversation d cannot influence the output of the current request q when restricted to one of the two distributions:

Assumption 2 (Robustness of a single distribution) *For any demonstration set d and request q, we have*

$$\mathbb{P}_H(a|[d,q]) = \mathbb{P}_H(a|q), \quad \textit{and} \quad \mathbb{P}_S(a|[d,q]) = \mathbb{P}_S(a|q). \tag{5.5}$$

Finally, given the divergence of the two distributions, it is less likely for a harmful output a_H to be generated from the safe distribution, so we can assume the probability ratio $\frac{\mathbb{P}_H(a_H|q)}{\mathbb{P}_S(a_H|q)}$ has a lower bound, and vice versa for safe outputs. We use a constant Δ to model this difference in the following assumption:

Assumption 3 (Divergence between the distributions) *There exists a constant $\Delta > 0$ such that for any request $\forall q \sim Q_H$, let $a_H = \arg\max \mathbb{P}_H(a|q)$ be the desired response from the harmful language distribution. Similarly, Let $a_S = \arg\max \mathbb{P}_S(a|q)$ be the response from the safe language distribution. Then we have*

$$\ln\left(\frac{\mathbb{P}_H(a_H|q)}{\mathbb{P}_S(a_H|q)}\right) > \Delta \quad \textit{and} \quad \ln\left(\frac{\mathbb{P}_S(a_S|q)}{\mathbb{P}_H(a_S|q)}\right) > \Delta. \tag{5.6}$$

5.2.2 Main Results

Based on the aforementioned framework, we can derive safety risk bounds of adversarial demonstrations. Our primary theorem offers insights into how ICA and ICD influence safety risk by characterizing an individual distribution. We then broaden our analysis to include mixed safety demonstration scenarios.

Understanding adversarial demonstrations. In the following theorem, we show that adversarial demonstration sets can approximate the generation safety risks close to the target distribution with only a logarithmic scale:

Theorem 5 *Given a target harmful request $q \sim Q_H$.*

*For $\forall \epsilon > 0$, by a set of k numbers of **harmful** demonstrations D_H where $k \geq \frac{1}{\Delta}(\ln 2 + \ln \frac{1}{\lambda} + \ln \frac{1}{\epsilon})$, it's sufficient to **increase** the model's safety risk to*

$$\mathcal{R}_{\mathbb{P}_H}(q) - \mathcal{R}_{\mathbb{P}}([D_H, q]) \leq \epsilon. \tag{5.7}$$

*In contrast, by a set of k numbers of **safe** demonstrations D_S where $k \geq \frac{1}{\Delta}(\ln 2 + \ln \frac{1}{1-\lambda} + \ln \frac{1}{\epsilon})$, it's sufficient to **decrease** the model's safety risk to*

$$\mathcal{R}_{\mathbb{P}}([D_S, q]) - \mathcal{R}_{\mathbb{P}_S}(q) \leq \epsilon. \tag{5.8}$$

proof sketch. To prove Theorem 5, we need the following lemma, which bounds the difference between the safety risks between $\mathbb{P}$, $\mathbb{P}_H$, and $\mathbb{P}_S$ with probability ratios:

Lemma 1 *Consider a prompt $p^* = [D, q]$ composed of a query $q \sim Q_H$ and a set of demonstrations D. We have*

$$|\mathcal{R}_{\mathbb{P}}(p^*) - \mathcal{R}_{\mathbb{P}_H}(p^*)| \leq \frac{2}{\lambda} \cdot \frac{\mathbb{P}_S(p^*)}{\mathbb{P}_H(p^*)} \tag{5.9}$$

and

$$|\mathcal{R}_{\mathbb{P}}(p^*) - \mathcal{R}_{\mathbb{P}_S}(p^*)| \leq \frac{2}{1-\lambda} \cdot \frac{\mathbb{P}_H(p^*)}{\mathbb{P}_S(p^*)}. \tag{5.10}$$

This lemma can be derived by expanding $\mathcal{R}_P(p^*) = \sum_a R(a)P(a|p^*)$ for the three distributions and using triangle inequalities. Then, by expanding $\mathbb{P}_S(p^*)$ and $\mathbb{P}_H(p^*)$ in the bound from Lemma 1 with Assumption 1 and 2, their ratio can be simplified as products of multiple individual probability ratios $\frac{\mathbb{P}_S(a_i|q_i)}{\mathbb{P}_H(a_i|q_i)}$. Finally, using Assumption 3 on these ratios can derive the target bound.

This theorem shows that for the queried harmful request q, to achieve comparable harmfulness with prompting the q in the harmful distribution $\mathbb{P}_H$, *i.e.* higher than $\mathcal{R}_{\mathbb{P}_H}(q) - \epsilon$, it only requires $O(\ln \frac{1}{\lambda} + \ln \frac{1}{\epsilon})$ demonstrations that on a logarithmic scale of $\frac{1}{\epsilon}$ and $\frac{1}{\lambda}$, where $\frac{1}{\lambda}$ measures the intrinsic safety of the model and $\frac{1}{\epsilon}$ measures how close is the safety risk to the harmful distribution. In contrast, decreasing the risk level of q comparable with the safe distribution, *i.e.* $\mathcal{R}_{\mathbb{P}_S}(q) + \epsilon$, only requires $O(\ln \frac{1}{1-\lambda} + \ln \frac{1}{\epsilon})$ demonstrations. Notably, since for aligned models the λ tends to 0, the term $\ln \frac{1}{1-\lambda}$ may be significantly smaller than $\ln \frac{1}{\lambda}$.

This notion aligns well with our experiment in the following section, where the number of demonstrations we used in ICD (1-2 shots) is significantly less than in ICA.

Extension on mixed distributions. We further extend this theory to the scenario that both safe and harmful demonstrations are included in the prompt. This can help us understand the robustness of LLMs guarded by safety prompts, exemplified by the case of attacking ICD guarded models with ICA. Under this setting, we additionally deduced the following corollary:

Corollary 1 *Suppose that a set of mixed demonstrations $D = [D_S, D_H]$ consists of k safe demonstrations*

$$D_S = [q_1, a_1^S, q_2, \cdots, q_k, a_k^S] \tag{5.11}$$

and l harmful demonstrations

$$D_H = [q_1', a_1^H, \cdots, q_l', a_l^H], \tag{5.12}$$

where $q_i, q_j' \sim Q_H$. Let $\Delta_1^S, \Delta_2^S, \cdots, \Delta_k^S$ be the divergence between the distributions used in Assumption 3 for the safe demonstrations, i.e. $\Delta_i^S = \ln(\frac{\mathbb{P}_S(a_i^S|q_i)}{\mathbb{P}_H(a_i^S|q_i)})$, and similarly let $\Delta_i^H = \ln(\frac{\mathbb{P}_H(a_i^H|q_i)}{\mathbb{P}_S(a_i^H|q_i)})$ for harmful demonstrations. For a harmful query $q \sim Q_H$ with prompt $p^* = [D_S, D_H, q]$, we have the following safety risk bounds:

$$\mathcal{R}_{\mathbb{P}}([D, q]) \leq \mathcal{R}_{\mathbb{P}_S}(q) + \frac{2}{1-\lambda} \cdot \frac{\exp(\Delta_1^H + \Delta_2^H + \cdots + \Delta_l^H)}{\exp(\Delta_1^S + \Delta_2^S + \cdots + \Delta_k^S)}, \tag{5.13}$$

and

$$\mathcal{R}_{\mathbb{P}}([D, q]) \geq \mathcal{R}_{\mathbb{P}_H}(q) - \frac{2}{\lambda} \cdot \frac{\exp(\Delta_1^S + \Delta_2^S + \cdots + \Delta_k^S)}{\exp(\Delta_1^H + \Delta_2^H + \cdots + \Delta_l^H)}. \tag{5.14}$$

The deduction of this corollary is similar to the proof of Theorem 5, where the difference is that we use the Δ_i^S, Δ_j^H instead of Assumption 3 that the inequality for Δ. This corollary shows that the safety risk for a harmful query q with mixed safe/harmful demonstrations can be bounded with Δ_i^S, Δ_j^H, which are the divergence between the distributions for all demonstrations. The Δ_j^H can be regarded as the harmfulness level of the demonstrations, since a more harmful example makes itself more distinguished from the safe distribution, and similarly, a higher Δ_i^S stands for stronger safety for safe demonstrations. This insight aligns with the intuition that more severe harmful examples lead to stronger attacks. Moreover, since the defense examples are fixed, the attacker, as a backhand in this interaction, can add more harmful examples to increase $\exp(\Delta_1^H + \Delta_2^H + \cdots + \Delta_l^H)$, which increases the lower bound of $\mathcal{R}_{\mathbb{P}}([D, q])$. This observation unveils an intrinsic limitation of defenses against attacks for LLMs, yet it can still

mitigate these safety risks to a certain extent. We further validate this observation by evaluating ICA *v.s.* ICD in the next section.

5.3 Experiments

In this section, we evaluate ICA and ICD to show their potential in practical attack and defense situations, beginning with an overview of the experimental setups.

5.3.1 Overall Evaluation Setups

Models and benchmarks. Following common practice [164, 70, 18], we mainly evaluate our proposed attack and defense on 4 popular aligned LLMs, including 3 open-sourced models (**Vicuna-7b-v1.5** [162], **Llama2-7b-chat** [112], and **QWen-7b-v2** [7]) and 1 closed-sourced model (**GPT-4** [87]). For the malicious requests, we use **AdvBench** [164], which consists of about 500 harmful behavior prompts, and **HarmBench** [76], a popular benchmark for evaluating red-teaming methods. However, we only use AdvBench for ICD evaluation since HarmBench did not provide defense implementation interfaces. The generation configurations and system messages are the same as the official default implementations.

Evaluation metric. As discussed by previous work [72, 65], different evaluation metrics may not report consistent results on attack success rates (**ASR**). To align with existing results and ensure a fair comparison with baselines, we use the original evaluation proxy proposed by the benchmarks. For Advbench, following GCG and subsequent works [164, 70, 163], we apply rejection string detection (*i.e.*, whether the response includes a rejection sub-string like `` `I cannot' ``) to judge the success of jailbreak. For Harmbench [76], we apply their official fine-tuned judging model (from Llama-13b) provided by the benchmark to check the harmfulness of a generated response.

Adversarial demonstrations. As discussed in Section 5.1, the demonstrations for ICA and ICD can be collected manually or automatically generated from jailbreak prompts. In our experiments, we randomly select 20 harmful requests from AdvBench and craft their corresponding harmful responses with GCG attack on vicuna-7b to generate the harmful demonstrations for ICA. For ICD, we collect the demonstrations by directly prompting the vanilla harmful requests on vicuna-7b to get the safe responses. We have manually checked that the responses are indeed harmful or safe for these demonstrations.

Table 5.1 ICA evaluation with different numbers of shots on **AdvBench** and **Harmbench**. Results that could not be completed due to the limited context window are indicated with a '-'.

Attack Success Rate (ASR)	AdvBench					HarmBench				
	Vicuna	Llama2	QWen	Mistral	GPT-4	Vicuna	Llama2	QWen	Mistral	GPT-4
No Attack	1%	0%	0%	1%	0%	19%	3%	9%	14%	11%
ICA (1 shot)	8%	0%	1%	6%	0%	24%	19%	10%	18%	11%
ICA (5 shots)	45%	12%	43%	31%	1%	59%	38%	43%	44%	12%
ICA (10 shots)	77%	58%	50%	70%	46%	60%	50%	46%	48%	32%
ICA (15 shots)	89%	-	55%	85%	79%	62%	-	53%	69%	55%
ICA (20 shots)	-	-	-	-	81%	-	-	-	-	65%

5.3.2 Evaluation on In-Context Attack

We conduct experiments to validate the effectiveness of ICA. First, we examine ICA with different numbers of shots to reveal their stealthy and scalability. Then, we compare ICA with some advanced attacking methods to show that it can achieve comparable ASR with them.

5.3.2.1 Scaling number of attacking shots

We consider applying ICA with $\{1, 5, 10, 15, 20\}$ shots to attack the evaluation models and summarize the results in Table 5.1. With only a single (**1 shot**) ICA demonstration, we can increase the ASR from 1% to 8% for Vicuna on AdvBench and from 3% to 19% for Llama-2 on HarmBench, showing the notable effectiveness of harmful demonstrations. Furthermore, as the number of demonstrations increases to 10, ICA significantly increases the ASR to 87% for vicuna and also successfully jailbreaks the closed-source model GPT-4 with a 46% ASR, validating the strong scalability of ICA that more harmful demonstrations can further boost the strength of the attack. Finally, we tried to scale up the numbers of demonstrations to 15 and 20 shots to sufficiently utilize the context window, where the ASRs on GPT-4 can be increased to **81%** and **65%** on the two datasets, respectfully. However, for the three 7b-size open-source models, their context windows are relatively limited (4096 tokens) and can only accommodate 15-shot ICA (10-shot for Llama-2 due to its very long system message), but the ASRs of ICA are still improved.

5.3.2.2 Benchmark results

To further validate the effectiveness of ICA, we also compare ICA with some advanced jailbreak attacks which achieved good performance on HarmBench [76], including three white-box attacks (**GCG**, **GCG-M**ultiple [164], **AutoDAN** [70]) and four black-box attacks (**GCG-T**ransfer [164], **PAIR** [18], **TAP**, **TAP-T**ransfer [77]). To ensure a fair comparison, we im-

Table 5.2 ICA evaluation on **HarmBenhch** [76] and its comparison with existing baselines.

Attack Success Rate (ASR)		White-box attacks			Black-box attacks				
Source	Model	GCG	GCG-M	AutoDAN	GCG-T	PAIR	TAP	TAP-T	**ICA** (ours)
Open source	Vicuna-7b	66%	62%	66%	61%	54%	51%	60%	62%
	Vicuna-13b	67%	61%	66%	55%	48%	55%	62%	65%
	Llama2-7b-chat	33%	21%	1%	20%	9%	9%	8%	50%
	QWen-7b-chat	59%	53%	47%	38%	50%	53%	59%	53%
	Mistral-7b-v2	70%	64%	72%	65%	53%	63%	66%	69%
	Average (↑)	58.8%	52.0%	49.2%	47.6%	42.6%	46.1%	51.0%	**59.8%**
Close source	Mistral-8x7b	-	-	-	63%	61%	70%	68%	**77%**
	GPT-4	-	-	-	22%	39%	43%	55%	**65%**

plemented ICA on the official repository of HarmBench with the default generation configurations and the maximum number of shots, then compared the results reported on the benchmark①, as summarized in Table 5.2. We also involved more models to show the universality of ICA, including Mistral-7b-v2 [53], and larger models like vicuna-13b-v1.5 [162] and Mistral-8x7b [54]. The average ASR is calculated over 5 white-box models.

These results indicate that our ICA reaches an average of approximately 60% ASR on white-box models, while exceeding 65% ASR on black-box models, consistently demonstrating effective attacking performance across various models. Additionally, when compared to other advanced attack methods, ICA provides comparable results with just one forward pass or query, whereas others often require multiple queries or white-box access. Overall, ICA demonstrates a strong potential as a practical attack or evaluation on LLM safety.

5.3.3 Evaluation on In-Context Defense

On the other hand, we conduct comprehensive evaluations to show how our ICD can mitigate jailbreak threats of LLMs while maintaining their natural performance, validating its practicality as a safeguard technique for LLM safety risks. We start with evaluating ICD against black-box attacks and white-box (adaptive) attacks, then study their impact on LLM natural performance. For the demonstrations used in ICD, we still randomly select malicious requests from AdvBench and use Vicuna to generate safe demonstrations by directly prompting the request without attacks. However, we show that only 1 or 2 demonstrations are sufficient to decrease the ASR of various attacks to a certain extent, which is much fewer than ICA.

① `https://www.harmbench.org/results`

Table 5.3 ASR comparison of ICD and baselines against **black-box** attacks.

Attack	GCG-T					PAIR				
Defense	Vicuna	Llama-2	QWen	GPT-4	**Average** (↓)	Vicuna	Llama-2	QWen	GPT-4	**Average** (↓)
No defense	60%	21%	35%	1%	29%	59%	26%	43%	20%	37%
Self-reminder	39%	14%	32%	0%	21%	50%	25%	34%	16%	31%
ICD (1 shot)	12%	0%	22%	0%	8%	51%	16%	14%	8%	22%
ICD (2 shots)	**4%**	**0%**	**21%**	**0%**	**6%**	**48%**	**2%**	**12%**	**2%**	**16%**

5.3.3.1 Attacks and Baseline Defenses

We first introduce the considered attacks for evaluation as well as baselines for ICD. To show the effectiveness of ICD in terms of defending against various types of jailbreaking attacks, following the similar setting of ICA, we evaluate ICD with various popular attacks, including GCG-T, PAIR (black-box), GCG, and AutoDAN (white-box). For GCG-T, the transferred suffix for open-source models (Vicuna, Llama-2, QWen) is trained with the other two models ensembled with 100 steps GCG, and for GPT-4 is trained with the three models ensembled. For PAIR, we follow the official implementation that uses 20 steps and the same model as the red-teaming LLM. Following AutoDAN [70], we apply 100 steps for optimization for both AutoDAN prefix and GCG suffix generation with the default hyperparameters in the official implementation.

Since our ICD is a prompt-based defense method, we compare it with Self-Reminder [141], which adds a safe instruction that reminds the LLM to generate safe content only in the system message as a baseline. However, as discussed, our ICD only requires adding conversations and does not require access to the system prompt. We additionally discuss other forms of defenses at the end of this section.

5.3.3.2 Defending against Black-box attacks

We first consider evaluation against black-box attacks, including GCG-T and PAIR, and summarize the ASR evaluated for the four models with different defenses in Table 5.3. Without any defense, these models exhibit fairly high ASR, particularly for the relatively weak Vicuna, which achieves about 60% under the two attacks. Though Self-Reminder can reduce the ASRs to a certain degree, in most cases, it remains undesired like Vicuna still has 39% against GCG-T. By contrast, with only a single safe demonstration (1 shot) incorporated into the context, ICD can significantly reduce the ASR (*e.g.* to 12% in the aforementioned case) on average, and further decrease it to nearly 0% in most cases when two shots demonstrations are applied, showing the desirable robustness against black-box jailbreak attacks.

Table 5.4 ASR comparison of ICD and baselines against **white-box adaptive** attacks.

Attack	GCG					AutoDAN				
Defense	Vicuna	Llama-2	QWen	Mistral	**Average** (↓)	Vicuna	Llama-2	QWen	Mistral	**Average** (↓)
No defense	95%	38%	63%	60%	64%	91%	54%	55%	40%	60%
Self-reminder	80%	36%	44%	20%	45%	88%	51%	53%	42%	58%
ICD (1 shot)	68%	26%	38%	32%	41%	86%	36%	47%	12%	45%
ICD (2 shots)	**60%**	**20%**	**24%**	**16%**	**30%**	**81%**	**27%**	**23%**	**0%**	**33%**

5.3.3.3 Defending against White-box (Adaptive) attacks.

To further assess the worst-case robustness of ICD, we also evaluate it with white-box adaptive attacks, including GCG and AutoDAN. Please note that AutoDAN is a white-box attack since the leveraged cross-entropy loss requires logits over tokens during generation. During the optimization process of the suffix and prefix, we incorporate the safe demonstrations when responding to the attack query, so the evaluation of ICD is **fully adaptive**.

The evaluation results are shown in Table 5.4. Given the strong capabilities of the attackers, the ASRs under these attacks are significantly high without defense, and the effectiveness of Self-reminder becomes more limited than black-box settings. However, our ICD can still notably reduce these ASRs to a certain extent, *e.g.* reduce the average ASR of GCG **from 64% to 30%** on average with 2 shots. These results evidence that ICD is still effective even against strong adaptive attacks.

5.3.3.4 Natural Performance

One key concern of deploying existing defenses is that they may decrease the natural performance of LLMs [26, 144]. To assess the influence of ICD on natural performance, we evaluate ICD in terms of both vanilla generation and computational costs.

For generation quality, we apply MT-bench [162], which is a popular benchmark that evaluates the instruction-following capability and generation helpfulness of LLMs. We evaluate vanilla generation and ICD with these benchmarks with both open-source (vicuna and mistral) and close-source (GPT-4) models and report the generation scores in Table 5.5, where the natural performance of ICD is still comparable with vanilla generation, showing the safety demonstration does not affect general generation quality. In some cases, the score of ICD is even slightly better than the vanilla generation, *e.g.* GPT-4 with 2-shots ICD (9.2) performs better than vanilla (8.9) on MT-Bench, which we identify as an intriguing property and worth further investigations.

We also estimate the computation cost of ICD and compare it with the vanilla generation,

Table 5.5 Average score of tasks from the MT-bench for different models and defenses.

Benchmark	MT-Bench (↑)			
Defense	Vicuna	Mistral	GPT-4	**Average**
No defense	6.7	7.9	8.9	7.8
ICD (1 shot)	6.8	7.9	9.2	8.0
ICD (2 shots)	6.6	7.8	9.2	7.9

as shown in Table 5.6. The generation time is averaged on computation during evaluation on MT-bench. Compared with the vanilla generation, Vicuna with two shots ICD only increases no more than 2% computational overhead, while the cost for GPT is less than 1%, which is negligible for practical usage. To summarize, we can conclude that ICD has only little influence on natural generation, making it an admissible defense technique against jailbreak attacks.

Table 5.6 Average inference time for different models and defenses.

Benchmark	Inference time (↓)			
Defense	Vicuna	Mistral	GPT-4	**Average**
No defense	1.00×	1.00×	1.00×	1.00×
ICD (1 shot)	1.01×	1.02×	< 1.01×	+1%
ICD (2 shots)	1.02×	1.02×	< 1.01×	+2%

5.3.3.5 Additional Baselines

In this part, we further compare and discuss ICD and other defense paradigms in addition to prompt-based defenses. These defenses include **pre-processing** based perplexity filter [1] and paraphrasing [51], inference-based Direct Representation Optimization (DPO) [161] and BackTranslation [126], as listed in Table 5.7. A common limitation of these defenses is that their evaluation is not adaptive, undermining their worst-case robustness assessment [100, 22]. For non-adaptive attacks, taking the GCG-T attack as an example, the performance of ICD is comparable with these baselines. Additionally, white-box defenses such as perplexity filtering and DRO cannot defend black-box models like GPT-4, which restricts their applicable scenarios.

5.3.4 Further Discussions

We finally discuss some interesting properties of ICA and ICD. First, we study their robustness against the selection of examples, followed by exploring their interactions.

Table 5.7 Comparing ICD and more baseline defenses.

Metric	GCG-T ASR.		MT-bench	
Defense	vicuna	GPT-4	vicuna	GPT-4
No defense	60%	1%	6.7	8.9
Perplexity	0%	N/A	6.7	N/A
Paraphrasing	16%	0%	6.7	8.4
BackTranslation	2%	0%	6.4	8.6
DRO	0%	N/A	6.5	N/A
ICD (1 shot)	12%	0%	**6.8**	9.2
ICD (2 shots)	4%	0%	6.6	**9.2**

5.3.4.1 Robustness of Demonstrations

To assess the performance of ICA and ICD with different sets of examples, we craft a demonstration pool with 50 harmful and safe examples, respectively.

Robustness of ICA examples. To evaluate the robustness of ICA against varying demonstration selections, we randomly sample multiple sets of 10 shots harmful examples from our demonstration pool and report the quartile statistics on vicuna-7b in Table 5.8. The average ASR reaches 73.5% on AdvBench and 62% on HarmBench, indicating that diverse sets of adversarial demonstrations can achieve desirable attack performance on average. Meanwhile, the overall standard deviation reveals moderate variance across different example sets. This suggests that while ICA maintains strong overall effectiveness, the attack success rate depends partially on the selection of examples, which may be influenced by factors like the persuasiveness or harmfulness of demonstrations. Crafting demonstrations that clearly exhibit harmful behaviors and cover diverse malicious intents may further stabilize ICA ' s performance, which we consider as an interesting future work.

Table 5.8 Robustness of demonstrations for ICA (10 shots) evaluated on vicuna.

Metric	AdvBench	Harmbench
No Attack	1%	19%
Upper quartile ASR	85%	66%
Avg. ASR ± Std.	73.5% ± 17.4%	62.0% ± 6.9%
Lower quartile ASR	64%	57%

Robustness of ICD examples. We similarly assess the robustness of ICD by testing its defense efficacy with randomly sampled 1 shot safe demonstrations against both black-box

(GCG-T) and white-box (GCG) attacks, as shown in Tables 5.9 and 5.10. Against GCG-T, ICD reduces the average ASR from 60% to 11.1% ± 2.3% on AdvBench and from 61% to 12.1% ± 1.8% on HarmBench, with tight quartile ranges, demonstrating consistent robustness across different safe examples. In contrast, against the adaptive GCG attack, ICD's ASR reduction (95% → 54.1% ± 11.3% on AdvBench; 66% → 39.5% ± 7.9% on HarmBench) exhibits relatively higher variance, indicating that stronger attacks may partially undermine the defense's stability, which may correlate with the clarity of refusal patterns and the explicit reinforcement of ethical guidelines in the safe demonstrations.

Table 5.9 Robustness of demonstrations for ICD against GCG-T (1 shot) on vicuna-7b.

Metric	AdvBench	Harmbench
No Attack	60%	61%
Lower quartile ASR	9%	11%
Avg. ASR ± Std.	11.1% ± 2.3%	12.1% ± 1.8%
Upper quartile ASR	13%	14%

Table 5.10 Robustness of demonstrations for ICD against GCG (2 shots) on vicuna-7b.

Metric	AdvBench	Harmbench
No Attack	95%	66%
Lower quartile ASR	56%	34%
Avg. ASR ± Std.	64.1% ± 11.3%	39.5% ± 7.9%
Upper quartile ASR	69%	42%

5.3.4.2 Evaluating ICD *v.s.* ICA.

Finally, we explore how the LLM performs when ICA is leveraged to attack ICD, where the prompt is organized as starting with safe demonstrations (added by the model developer) and followed by harmful demonstrations (added by the attacker).

Table 5.11 ASRs of ICD against ICA.

	Attack		
Defense	ICA (1 shot)	ICA (5 shots)	ICA (10 shots)
No defense	8%	45%	87%
ICD (1 shot)	2%	38%	59%
ICD (2 shots)	1%	36%	56%

We evaluate this on vicuna-7b and report the results in Table 5.11, where we can see that when the attacker only uses 1 shot harmful demonstration, ICD with similar numbers of demonstrations can easily eliminate the threat. However, when the attack's capacity scales up to 5 or 10 shots, the harmful demonstrations can subvert the safe ones and maintain a fairly high ASR, showing the consistent effectiveness of ICA when the number of harmful demonstrations is scaled up. Nevertheless, ICD is still useful in this setting as it can reduce the harmfulness with less capacity. These insights also align with our observation in Corollary 1, where ICD can reduce the safety risk to a certain extent.

Overall, our proposed ICA and ICD show strong potential for attacking and defending against LLMs, providing new avenues for safety research on LLMs.

5.4 Summary

In this chapter, we uncover the power of in-context demonstrations in manipulating the alignment ability of LLMs for both attack and defense purposes by the proposed two techniques: In-Context Attack (ICA) and In-Context Defense (ICD). For ICA, we show that a few demonstrations of responding to malicious prompts can jailbreak the model to generate harmful content. On the other hand, ICD enhances model robustness by demonstrations of rejecting harmful prompts. We also provide theoretical understandings to illustrate the effectiveness of only a few adversarial demonstrations. Finally, our comprehensive evaluations illustrate the practicality and effectiveness of ICA and ICD, highlighting their significant potential on LLMs alignment and safety and providing a new perspective to study this issue.

5.5 Proofs of Theorems

In this section, we provide complete proof of Theorem 5. We start with the proof of Lemma 1:

Proof of Lemma 1. Note that

$$\left|\mathcal{R}_{\mathbb{P}}(p^*) - \mathcal{R}_{\mathbb{P}_H}(p^*)\right| \tag{5.15}$$

$$= \left|\sum_a R(a)\mathbb{P}(a|p^*) - \sum_a R(a)\mathbb{P}_H(a|p^*)\right| \tag{5.16}$$

$$= \left|\sum_a R(a)\left[\mathbb{P}(a|p^*) - \mathbb{P}_H(a|p^*)\right]\right| \tag{5.17}$$

$$\leq \sum_a |R(a)| \cdot |\mathbb{P}(a|p^*) - \mathbb{P}_H(a|p^*)| \quad \text{(triangle inequality)} \tag{5.18}$$

$$\leq \sum_a |\mathbb{P}(a|p^*) - \mathbb{P}_H(a|p^*)| \quad (0 \leq R(a) \leq 1) \tag{5.19}$$

$$= \sum_a \left|\frac{\mathbb{P}([p^*, a])}{\mathbb{P}(p^*)} - \frac{\mathbb{P}_H([p^*, a])}{\mathbb{P}_H(p^*)}\right| \tag{5.20}$$

$$= \sum_a \left|\frac{\lambda\mathbb{P}_H([p^*, a]) + (1-\lambda)\mathbb{P}_S([p^*, a])}{\lambda\mathbb{P}_H(p^*) + (1-\lambda)\mathbb{P}_S(p^*)} - \frac{\mathbb{P}_H([p^*, a])}{\mathbb{P}_H(p^*)}\right| \tag{5.21}$$

$$= \sum_a \left|\frac{\left[\lambda\mathbb{P}_H([p^*, a]) + (1-\lambda)\mathbb{P}_S([p^*, a])\right]\mathbb{P}_H(p^*) - \left[\lambda\mathbb{P}_H(p^*) + (1-\lambda)\mathbb{P}_S(p^*)\right]\mathbb{P}_H([p^*, a])}{\left[\lambda\mathbb{P}_H(p^*) + (1-\lambda)\mathbb{P}_S(p^*)\right]\mathbb{P}_H(p^*)}\right| \tag{5.22}$$

$$= \sum_a \left|\frac{(1-\lambda)\mathbb{P}_S([p^*, a])\mathbb{P}_H(p^*) - (1-\lambda)\mathbb{P}_S(p^*)\mathbb{P}_H([p^*, a])}{\left[\lambda\mathbb{P}_H(p^*) + (1-\lambda)\mathbb{P}_S(p^*)\right]\mathbb{P}_H(p^*)}\right| \tag{5.23}$$

$$= \sum_a \frac{\mathbb{P}_S(p^*)}{\mathbb{P}_H(p^*)} \cdot \left|\frac{(1-\lambda)\frac{\mathbb{P}_S([p^*, a])}{\mathbb{P}_S(p^*)}\mathbb{P}_H(p^*) - (1-\lambda)\mathbb{P}_H([p^*, a])}{\lambda\mathbb{P}_H(p^*) + (1-\lambda)\mathbb{P}_S(p^*)}\right| \tag{5.24}$$

$$\leq \sum_a \frac{\mathbb{P}_S(p^*)}{\mathbb{P}_H(p^*)} \cdot \left\{\frac{\left|\frac{\mathbb{P}_S([p^*, a])}{\mathbb{P}_S(p^*)}\mathbb{P}_H(p^*)\right| + |\mathbb{P}_H([p^*, a])|}{\lambda\mathbb{P}_H(p^*)}\right\} \quad (1-\lambda > 0, \text{ triangle inequality}) \tag{5.25}$$

$$= \frac{1}{\lambda}\frac{\mathbb{P}_S(p^*)}{\mathbb{P}_H(p^*)} \cdot \sum_a \{\mathbb{P}_S(a|p^*) + \mathbb{P}_H(a|p^*)\} \tag{5.26}$$

$$= \frac{2}{\lambda}\frac{\mathbb{P}_S(p^*)}{\mathbb{P}_H(p^*)}. \tag{5.27}$$

Similarly, we can derive Equation (5.10) by symmetry.

Proof of Theorem 5. Note that

$$
\begin{aligned}
&\frac{\mathbb{P}_S(p^*)}{\mathbb{P}_H(p^*)} \\
=&\frac{\mathbb{P}_S([q_1,a_1,\cdots,q_k,a_k,q])}{\mathbb{P}_H([q_1,a_1,\cdots,q_k,a_k,q])} \\
=&\frac{\mathbb{P}_S(q|[q_1,a_1,\cdots,q_k,a_k])}{\mathbb{P}_H(q|[q_1,a_1,\cdots,q_k,a_k])}\cdot\frac{\mathbb{P}_S([q_1,a_1,\cdots,q_k,a_k])}{\mathbb{P}_H([q_1,a_1,\cdots,q_k,a_k])} \\
=&\frac{\mathbb{P}_S([q_1,a_1,\cdots,q_k,a_k])}{\mathbb{P}_H([q_1,a_1,\cdots,q_k,a_k])} \quad \text{(Assumption 1)} \\
=&\frac{\mathbb{P}_S(a_k|[q_1,a_1,\cdots,q_k])}{\mathbb{P}_H(a_k|[q_1,a_1,\cdots,q_k])}\cdot\frac{\mathbb{P}_S([q_1,a_1,\cdots,q_k])}{\mathbb{P}_H([q_1,a_1,\cdots,q_k])} \\
=&\frac{\mathbb{P}_S(a_k|q_k)}{\mathbb{P}_H(a_k|q_k)}\cdot\frac{\mathbb{P}_S([q_1,a_1,\cdots,q_k])}{\mathbb{P}_H([q_1,a_1,\cdots,q_k])} \quad \text{(Assumption 2)} \\
=&\frac{\mathbb{P}_S(a_k|q_k)}{\mathbb{P}_H(a_k|q_k)}\cdot\frac{\mathbb{P}_S(q_k|[q_1,a_1,\cdots,q_{k-1},a_{k-1}])}{\mathbb{P}_H(q_k|[q_1,a_1,\cdots,q_{k-1},a_{k-1}])}\cdot\frac{\mathbb{P}_S([q_1,a_1,\cdots,q_{k-1},a_{k-1}])}{\mathbb{P}_H([q_1,a_1,\cdots,q_{k-1},a_{k-1}])} \\
=&\frac{\mathbb{P}_S(a_k|q_k)}{\mathbb{P}_H(a_k|q_k)}\cdot\frac{\mathbb{P}_S([q_1,a_1,\cdots,q_{k-1},a_{k-1}])}{\mathbb{P}_H([q_1,a_1,\cdots,q_{k-1},a_{k-1}])} \quad \text{(Assumption 1)} \\
=&\frac{\mathbb{P}_S(a_k|q_k)}{\mathbb{P}_H(a_k|q_k)}\cdot\frac{\mathbb{P}_S(a_{k-1}|q_{k-1})}{\mathbb{P}_H(a_{k-1}|q_{k-1})}\cdot\frac{\mathbb{P}_S([q_1,a_1,\cdots,q_{k-2},a_{k-2}])}{\mathbb{P}_H([q_1,a_1,\cdots,q_{k-2},a_{k-2}])} \\
=&\cdots \\
=&\prod_{i=1}^{k}\frac{\mathbb{P}_S(a_i|q_i)}{\mathbb{P}_H(a_i|q_i)} \le \prod_{i=1}^{k} e^{-\Delta} \quad \text{(Assumption 3)} \\
=&e^{-k\Delta}.
\end{aligned}
\tag{5.28}
$$

Note that by Assumption 2, we have

$$
\mathcal{R}_{\mathbb{P}_H}([D_H,q]) = \sum_a R(a)\mathbb{P}_H(a|[D_H,q]) = \sum_a R(a)\mathbb{P}_H(a|q) = \mathcal{R}_{\mathbb{P}_H}(q). \tag{5.29}
$$

Therefore, by Lemma 1, we have

$$
\mathcal{R}_{\mathbb{P}}([D_H,q]) = \mathcal{R}_{\mathbb{P}}(q) \ge \mathcal{R}_{\mathbb{P}_H}([D_H,q]) - \frac{2}{\lambda}\cdot\frac{\mathbb{P}_S(p^*)}{\mathbb{P}_H(p^*)} \ge \mathcal{R}_{\mathbb{P}_H}(q) - \frac{2}{\lambda}\cdot e^{-k\Delta}. \tag{5.30}
$$

For $k \ge \frac{1}{\Delta}\left(\ln 2 + \ln\frac{1}{\lambda} + \ln\frac{1}{\epsilon}\right)$, we have

$$
\mathcal{R}_{\mathbb{P}}([D_H,q]) \ge \mathcal{R}_{\mathbb{P}_H}(q) - \frac{2}{\lambda}(\frac{\lambda\epsilon}{2}) = \mathcal{R}_{\mathbb{P}_H}(q) - \epsilon. \tag{5.31}
$$

Similarly, for $k \ge \frac{1}{\Delta}\left(\ln 2 + \ln\frac{1}{(1-\lambda)} + \ln\frac{1}{\epsilon}\right)$, we have

$$
\mathcal{R}_{\mathbb{P}}([D_S,q]) \le \mathcal{R}_{\mathbb{P}_S}(q) + \epsilon. \tag{5.32}
$$

Proof of Corollary 1. By the same deduction of Theorem 5 we have

$$\frac{\mathbb{P}_S(p^*)}{\mathbb{P}_H(p^*)} = \left(\prod_{i=1}^{k} \frac{\mathbb{P}_S(a_i^S|q_i)}{\mathbb{P}_H(a_i^S|q_i)}\right) \times \left(\prod_{j=1}^{l} \frac{\mathbb{P}_S(a_j^H|q'_j)}{\mathbb{P}_H(a_j^H|q'_j)}\right). \tag{5.33}$$

Since $\Delta_i^S = \ln(\frac{\mathbb{P}_S(a_i^S|q_i)}{\mathbb{P}_H(a_i^S|q_i)})$ and $\Delta_i^H = \ln(\frac{\mathbb{P}_H(a_i^H|q_i)}{\mathbb{P}_S(a_i^H|q_i)})$, we can derive

$$\frac{\mathbb{P}_S(p^*)}{\mathbb{P}_H(p^*)} = \left(\prod_{i=1}^{k} \exp(\Delta_i^S)\right) \times \left(\prod_{i=1}^{k} \exp(-\Delta_i^H)\right) = \frac{\exp\left(\Delta_1^S + \Delta_2^S + \cdots + \Delta_k^S\right)}{\exp\left(\Delta_1^H + \Delta_2^H + \cdots + \Delta_l^H\right)}. \tag{5.34}$$

Similar to Equation (5.30), we have

$$\mathcal{R}_{\mathbb{P}}([D_S, D_H, q]) = \mathcal{R}_{\mathbb{P}}(q) \geq \mathcal{R}_{\mathbb{P}_H}([D_S, D_H, q]) - \frac{2}{\lambda} \cdot \frac{\mathbb{P}_S(p^*)}{\mathbb{P}_H(p^*)} \tag{5.35}$$

$$\geq \mathcal{R}_{\mathbb{P}_H}(q) - \frac{2}{\lambda} \cdot \frac{\exp\left(\Delta_1^S + \Delta_2^S + \cdots + \Delta_k^S\right)}{\exp\left(\Delta_1^H + \Delta_2^H + \cdots + \Delta_l^H\right)} \tag{5.36}$$

and

$$\mathcal{R}_{\mathbb{P}}([D_S, D_H, q]) = \mathcal{R}_{\mathbb{P}}(q) \leq \mathcal{R}_{\mathbb{P}_S}([D_S, D_H, q]) + \frac{2}{1-\lambda} \cdot \frac{\mathbb{P}_H(p^*)}{\mathbb{P}_S(p^*)} \tag{5.37}$$

$$\leq \mathcal{R}_{\mathbb{P}_H}(q) + \frac{2}{1-\lambda} \cdot \frac{\exp\left(\Delta_1^H + \Delta_2^H + \cdots + \Delta_l^H\right)}{\exp\left(\Delta_1^S + \Delta_2^S + \cdots + \Delta_k^S\right)}. \tag{5.38}$$

Chapter 6 Conclusion and Future Work

This chapter summarizes the research contributions presented in the thesis towards trustworthy machine learning from data distribution perspectives and highlights potential future research directions.

6.1 Conclusion

In this thesis, we propose a thread of insights into improving the trustworthiness of machine learning models, inspired by their corresponding data distributions. Concretely, the research contents include:

- **Scalable Automata Extraction and Explanation Framework**. We propose a novel framework for extracting and explaining weighted finite automata (WFA) from recurrent neural networks (RNNs) for natural language tasks. Our method addresses transition sparsity and context loss problems, and introduces Transition Matrix Embeddings (TME) for model explanation. Experiments demonstrate the effectiveness of our approach in improving extraction precision and providing task-oriented insights for RNNs.
- **Class-wise Calibrated Fair Adversarial Training (CFA)**. We present a theoretical analysis of how different adversarial configurations impact class-wise robustness and propose the CFA framework. CFA dynamically customizes training configurations for each class, improving both overall and worst-class robustness. Our experiments show that CFA outperforms state-of-the-art methods in terms of both robustness and fairness.
- **In-context Attack and Defense for Large Language Model Safety**. We explore the use of in-context learning to manipulate the safety alignment of LLMs. We propose In-Context Attack (ICA) and In-Context Defense (ICD) methods, supported by theoretical analysis and empirical evaluation. Our results demonstrate the effectiveness of these methods in attacking and defending LLMs, highlighting the potential of in-context learning for improving LLM safety.

Overall, this thesis provides novel insights into the interpretability, robustness, and safety of machine learning models from a data distribution perspective.

6.2 Future Work

This thesis presents novel data distribution insights into trustworthy ML, warranting further research through this perspective. Based on our three main contributions, we list a few concrete future research directions as follows.

- **Representation-guided abstract model extraction from large models**. Extending from RNNs to large models may inherently be challenging to extract abstract models that can comprehensively simulate the behaviors of LLMs. However, future research could focus on advanced techniques for deriving abstract models from LLMs based on specific model representations. This would entail developing methods that can efficiently capture essential representations through a trustworthy-specific distribution, such as safety or fairness, to analyze and interpret the model's behaviors from a particular perspective.
- **Advanced adversarial training tailored by class-wise features**. Building on our class-wise adversarial data analysis, future research could investigate more sophisticated class-wise data distribution modeling and AT strategies. This may include in-depth investigations of class-wise data and feature interactions, as well as adaptive methods that dynamically adjust AT configurations based on the characteristics of class-wise and inter-class data distributions during training.
- **In-context safeguarding for LLM-driven agents**. LLM-based agents have become one of the most successful applications of LLMs, yet they also suffer from more complex trustworthiness risks. As these agents are increasingly deployed in real-world applications, there is a need for robust safeguarding mechanisms. Based on our proposed ICA and ICD paradigms, future work could focus on developing in-context safeguarding techniques that can dynamically adapt to emerging threats and ensure the safe operation of LLM-driven agents in various scenarios.

参考文献

[1] Gabriel Alon and Michael Kamfonas. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*, 2023.

[2] Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks. In *Workshop on Next Generation of AI Safety*, 2024.

[3] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and computation*, 75(2):87–106, 1987.

[4] Cem Anil, Esin Durmus, Nina Rimsky, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel J Ford, et al. Many-shot jailbreaking. In *NeurIPS*, 2024.

[5] Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, et al. Foundational challenges in assuring alignment and safety of large language models. *Transactions on Machine Learning Research*, 2024.

[6] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

[7] Jinze Bai et al. Qwen technical report. *https://qwenlm.github.io/blog/qwen3/*, 2023.

[8] Yang Bai, Yan Feng, Yisen Wang, Tao Dai, Shu-Tao Xia, and Yong Jiang. Hilbert-based generative defense for adversarial examples. In *ICCV*, 2019.

[9] Yuntao Bai et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[10] Yogesh Balaji, Tom Goldstein, and Judy Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint arXiv:1910.08051*, 2019.

[11] Advik Raj Basani and Xiao Zhang. Gasp: Efficient black-box generation of adversarial suffixes for jailbreaking llms. *arXiv preprint arXiv:2411.14133*, 2024.

[12] Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Robustness may be at odds with fairness: An empirical study on class-wise accuracy. *arXiv preprint arXiv:2010.13365*, 2020.

[13] Tom B. Brown et al. Language models are few-shot learners. In *NeurIPS*, 2024.

[14] Qi-Zhi Cai, Min Du, Chang Liu, and Dawn Song. Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*, 2018.

[15] Bochuan Cao, Yuanpu Cao, Lu Lin, and Jinghui Chen. Defending against alignment-breaking attacks via robustly aligned llm. In *ACL*, 2023.

[16] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Workshop on artificial intelligence and security*, 2017.

[17] Adelmo Luis Cechin, D Regina, P Simon, and K Stertz. State automata extraction from recurrent neural nets using k-means and fuzzy clustering. In *SCCC*, 2003.

[18] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

[19] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

[20] Canyu Chen and Kai Shu. Combating misinformation in the age of llms: Opportunities and challenges. *AI Magazine*, pages 354–368, 2024.

[21] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *CVPR*, 2015.

[22] Huanran Chen, Yinpeng Dong, Zeming Wei, Hang Su, and Jun Zhu. Towards the worst-case robustness of large language models. *arXiv preprint arXiv:2501.19040*, 2025.

[23] Chen Cheng, Xinzhi Yu, Haodong Wen, Jingsong Sun, Guanzhang Yue, Yihao Zhang, and Zeming Wei. Exploring the robustness of in-context learning with noisy labels. In *ICASSP*, 2025.

[24] 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.

[25] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

[26] Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. Or-bench: An over-refusal benchmark for large language models. *arXiv preprint arXiv:2405.20947*, 2024.

[27] Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. *arXiv preprint arXiv:2212.10559*, 2022.

[28] Josef Dai, Xuehai Pan, Ruiyang Sun, et al. Safe rlhf: Safe reinforcement learning from human feedback. In *ICLR*, 2024.

[29] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017.

[30] Debajit Datta, Preetha Evangeline David, Dhruv Mittal, and Anukriti Jain. Neural machine translation using recurrent neural network. *International Journal of Engineering and Advanced Technology*, 9(4):1395–1400, 2020.

[31] Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. Multilingual jailbreak challenges in large language models. In *ICLR*, 2024.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[33] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. *arXiv preprint arXiv:1812.02637*, 2018.

[34] Guoliang Dong, Jingyi Wang, Jun Sun, Yang Zhang, Xinyu Wang, Ting Dai, Jin Song Dong, and Xingen Wang. Towards interpreting recurrent neural networks through probabilistic abstraction. In *ASE*, 2020.

[35] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. A survey on in-context learning. In *EMNLP*, 2023.

[36] Xiaoning Du, Yi Li, Xiaofei Xie, Lei Ma, Yang Liu, and Jianjun Zhao. Marble: Model-based robustness analysis of stateful deep learning systems. In *ASE*, 2020.

[37] Xiaoning Du, Xiaofei Xie, Yi Li, Lei Ma, Yang Liu, and Jianjun Zhao. Deepstellar: Model-based quantitative analysis of stateful deep learning systems. In *FSE*, 2019.

[38] Michael Feffer, Anusha Sinha, Wesley H Deng, Zachary C Lipton, and Hoda Heidari. Red-teaming for generative ai: Silver bullet or security theater? In *AAAI/ACM Conference on AI, Ethics, and Society*, 2024.

[39] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[40] Tian Guo, Tao Lin, and Yao Lu. An interpretable lstm neural network for autoregressive exogenous model. *arXiv preprint arXiv:1804.05251*, 2018.

[41] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S Lew. A review of semantic segmentation using deep neural networks. *International journal of multimedia information retrieval*, 7:87–93, 2018.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[44] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[45] Bo-Jian Hou and Zhi-Hua Zhou. Learning with interpretable structure from gated rnn. *IEEE transactions on neural networks and learning systems*, 31(7):2267–2279, 2020.

[46] Yuzheng Hu, Fan Wu, Hongyang Zhang, and Han Zhao. Understanding the impact of adversarial robustness on accuracy disparity. In *ICML*, 2023.

[47] Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic jailbreak of open-source llms via exploiting generation. In *ICLR*, 2024.

[48] Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

[49] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018.

[50] Henrik Jacobsson. Rule extraction from recurrent neural networks: Ataxonomy and review. *Neural Computation*, 17(6):1223–1263, 2005.

[51] Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*, 2023.

[52] Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Juntao Dai, and Yaodong Yang. Aligner: Achieving efficient alignment through weak-to-strong correction. In *NeurIPS*, 2024.

[53] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. 2023.

[54] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[55] Chengyue Jiang, Yinggong Zhao, Shanbo Chu, Libin Shen, and Kewei Tu. Cold-start and interpretability: Turning regular expressions into trainable recurrent neural networks. In *EMNLP*, 2020.

[56] Fengqing Jiang, Zhangchen Xu, Luyao Niu, Zhen Xiang, Bhaskar Ramasubramanian, Bo Li, and Radha Poovendran. Artprompt: Ascii art-based jailbreak attacks against aligned llms. In *ACL*, 2024.

[57] Jigsaw. Toxic comment classification challenge. *SMU Data Science Review*, 2018.

[58] Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Bhalerao, Christopher L. Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. Pretraining language models with human preferences. In *ICML*, 2023.

[59] Viktoriya Krakovna and Finale Doshi-Velez. Increasing the interpretability of recurrent neural networks using hidden markov models. *arXiv preprint arXiv:1606.05320*, 2016.

[60] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[61] Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Soheil Feizi, and Hima Lakkaraju. Certifying llm safety against adversarial prompting. In *COLM*, 2023.

[62] Vishal Kumar, Zeyi Liao, Jaylen Jones, and Huan Sun. Amplegcg-plus: A strong generative model of adversarial suffixes to jailbreak llms with higher success rates in fewer attempts. *arXiv preprint arXiv:2410.22143*, 2024.

[63] Raz Lapid, Ron Langberg, and Moshe Sipper. Open sesame! universal black box jailbreaking of large language models. In *Workshop on Secure and Trustworthy Large Language Models*, 2024.

[64] Xin Li and Dan Roth. Learning question classifiers. In *COLING*, 2002.

[65] Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*, 2023.

[66] Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. Rain: Your language models can align themselves without finetuning. In *ICLR*, 2024.

[67] Zeyi Liao and Huan Sun. Amplegcg: Learning a universal and transferable generative model of adversarial suffixes for jailbreaking both open and closed llms. In *COLM*, 2024.

[68] Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. In *ICLR*, 2023.

[69] Haoyang Liu, Maheep Chaudhary, and Haohan Wang. Towards trustworthy and aligned machine learning: A data-centric survey with causality perspectives. *arXiv preprint arXiv:2307.16851*, 2023.

[70] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *ICLR*, 2024.

[71] Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023.

[72] Lin Lu, Hai Yan, Zenghui Yuan, Jiawen Shi, Wenqi Wei, Pin-Yu Chen, and Pan Zhou. Autojailbreak: Exploring jailbreak attacks and defenses through a dependency lens. *arXiv preprint arXiv:2406.03805*, 2024.

[73] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 2020.

[74] Xinsong Ma, Zekai Wang, and Weiwei Liu. On the tradeoff between robustness and fairness. In *NeurIPS*, 2022.

[75] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[76] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *ICML*, 2024.

[77] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. In *NeurIPS*, 2024.

[78] ML Menéndez, JA Pardo, L Pardo, and MC Pardo. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997.

[79] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[80] Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Noisy channel language model prompting for few-shot text classification. In *ACL*, 2022.

[81] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

[82] Yichuan Mo, Dongxian Wu, Yifei Wang, Yiwen Guo, and Yisen Wang. When adversarial training meets vision transformers: Recipes from training to architecture. In *NeurIPS*, 2022.

[83] Takamasa Okudono, Masaki Waga, Taro Sekiyama, and Ichiro Hasuo. Weighted automata extraction from recurrent neural networks via regression on state spaces. In *AAAI*, 2020.

[84] Christian W Omlin and C Lee Giles. Extraction of rules from discrete-time recurrent neural networks. *Neural networks*, 9(1):41–52, 1996.

[85] Christian W. Omlin and C. Lee Giles. Rule revision with recurrent neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 8(1):183–188, 1996.

[86] CW Omlin, CL Giles, and CB Miller. Heuristics for the extraction of rules from discrete-time recurrent neural networks. In *IJCNN*, 1992.

[87] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024.

[88] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[89] Swetasudha Panda, Naveen Jafer Nizar, and Michael L Wick. Llm improvement for jailbreak defense: Analysis through the lens of over-refusal. In *Workshop on Safe Generative AI*, 2024.

[90] Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, Min Lin, et al. Improved few-shot jailbreaking can circumvent aligned language models and their defenses. In *NeurIPS*, 2025.

[91] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *SP*, 2016.

[92] Anselm Paulus, Arman Zharmagambetov, Chuan Guo, Brandon Amos, and Yuandong Tian. Advprompter: Fast adaptive adversarial prompting for llms. *arXiv preprint arXiv:2404.16873*, 2024.

[93] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[94] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In *EMNLP*, 2022.

[95] Mansi Phute, Alec Helbling, Matthew Hull, ShengYun Peng, Sebastian Szyller, Cory Cornelius, and Duen Horng Chau. Llm self defense: By self examination, llms know they are being tricked. In *Tiny Papers Track at ICLR*, 2023.

[96] David M. W. Powers. Applications and explanations of zipf's law. In *New methods in language processing and computational natural language learning*, 1998.

[97] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *AAAI*, 2024.

[98] Yao Qiang, Xiangyu Zhou, and Dongxiao Zhu. Hijacking large language models via adversarial in-context learning. *arXiv preprint arXiv:2311.09948*, 2023.

[99] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[100] Javier Rando, Jie Zhang, Nicholas Carlini, and Florian Tramèr. Adversarial ml problems are getting harder to solve and to evaluate. *arXiv preprint arXiv:2502.02260*, 2025.

[101] Abhinav Rao, Sachin Vashistha, Atharva Naik, Somak Aditya, and Monojit Choudhury. Tricking llms into disobedience: Formalizing, analyzing, and detecting jailbreaks. *arXiv preprint arXiv:2305.14965*, 2023.

[102] Qibing Ren, Chang Gao, Jing Shao, Junchi Yan, Xin Tan, Wai Lam, and Lizhuang Ma. Exploring safety generalization challenges of large language models via code. In *ACL*, 2024.

[103] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *ICML*, 2020.

[104] Alexander Robey, Eric Wong, Hamed Hassani, and George Pappas. Smoothllm: Defending large language models against jailbreaking attacks. In *Workshop on Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.

[105] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[106] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *CCS*, 2023.

[107] Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. Latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*, 2024.

[108] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV au2, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *EMNLP*, 2020.

[109] Gokul Swamy, Christoph Dann, Rahul Kidambi, et al. A minimaximalist approach to reinforcement learning from human feedback. In *ICML*, 2024.

[110] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[111] Qi Tian, Kun Kuang, Kelu Jiang, Fei Wu, and Yisen Wang. Analysis and applications of class-wise robustness in adversarial training. In *KDD*, 2021.

[112] Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[113] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[114] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[115] Neeraj Varshney, Pavel Dolin, Agastya Seth, and Chitta Baral. The art of defending: A systematic evaluation and analysis of llm defense strategies on safety and over-defensiveness. In *Findings of ACL*, 2023.

[116] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[117] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2019.

[118] Hongjun Wang and Yisen Wang. Self-ensemble adversarial training for improved robustness. In *ICLR*, 2022.

[119] Hongjun Wang and Yisen Wang. Generalist: Decoupling natural and robust generalization. In *CVPR*, 2023.

[120] Jiongxiao Wang, Zichen Liu, Keun Hee Park, Zhuojun Jiang, Zhaoheng Zheng, Zhuofeng Wu, Muhao Chen, and Chaowei Xiao. Adversarial demonstration attacks on large language models. *arXiv preprint arXiv:2305.14950*, 2023.

[121] Qinglong Wang, Kaixuan Zhang, Xue Liu, and C Lee Giles. Verification of recurrent neural networks through rule extraction. *arXiv preprint arXiv:1811.06029*, 2018.

[122] Qinglong Wang, Kaixuan Zhang, Alexander G Ororbia II, Xinyu Xing, Xue Liu, and C Lee Giles. An empirical evaluation of rule extraction from recurrent neural networks. *Neural computation*, 30(9):2568–2591, 2018.

[123] Ruishuang Wang, Zhao Li, Jian Cao, Tong Chen, and Lei Wang. Convolutional recurrent neural networks for text classification. In *IJCNN*, 2019.

[124] Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. In *Workshop on Efficient Systems for Foundation Models*, 2023.

[125] Yifei Wang, Yuyang Wu, Zeming Wei, Stefanie Jegelka, and Yisen Wang. A theoretical understanding of self-correction through in-context alignment. In *NeurIPS*, 2024.

[126] Yihan Wang, Zhouxing Shi, Andrew Bai, and Cho-Jui Hsieh. Defending llms against jailbreaking attacks via backtranslation. In *Findings of ACL*, 2024.

[127] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, 2019.

[128] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2019.

[129] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? In *NeurIPS*, 2023.

[130] Zeming Wei, Tianlin Li, Xiaojun Jia, Yang Liu, and Meng Sun. Position: Agent-specific trustworthiness risk as a research priority. *OpenReview preprint*, 2025.

[131] Gail Weiss, Yoav Goldberg, and Eran Yahav. Extracting automata from recurrent neural networks using queries and counterexamples. In *ICML*, 2018.

[132] Gail Weiss, Yoav Goldberg, and Eran Yahav. Learning deterministic weighted automata with queries and counterexamples. In *NeurIPS*, 2019.

[133] Noam Wies, Yoav Levine, and Amnon Shashua. The learnability of in-context learning. In *NeurIPS*, 2023.

[134] Yotam Wolf, Noam Wies, Oshri Avnery, Yoav Levine, and Amnon Shashua. Fundamental limitations of alignment in large language models. In *ICML*, 2024.

[135] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020.

[136] Sophie Xhonneux, Alessandro Sordoni, Stephan Günnemann, Gauthier Gidel, and Leo Schwinn. Efficient adversarial training in LLMs with continuous attacks. In *NeurIPS*, 2024.

[137] Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. Badchain: Backdoor chain-of-thought prompting for large language models. In *ICLR*, 2024.

[138] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019.

[139] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

[140] Xiaofei Xie, Wenbo Guo, Lei Ma, Wei Le, Jian Wang, Lingjun Zhou, Yang Liu, and Xinyu Xing. Rnnrepair: Automatic rnn repair via model-based analysis. In *ICML*, 2021.

[141] Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 2023.

[142] Benfeng Xu, Quan Wang, Zhendong Mao, Yajuan Lyu, Qiaoqiao She, and Yongdong Zhang. kNN prompting: Beyond-context learning with calibration-free nearest neighbor inference. In *ICLR*, 2023.

[143] Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. In *ICML*, 2021.

[144] Qianqiao Xu, Zhiliang Tian, Hongyan Wu, Zhen Huang, Yiping Song, Feng Liu, and Dongsheng Li. Learn to disguise: Avoid refusal responses in llm's defense via a multi-agent attacker-disguiser game. *arXiv preprint arXiv:2404.02532*, 2024.

[145] Zhangchen Xu, Fengqing Jiang, Luyao Niu, et al. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. In *ACL*, 2024.

[146] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 2024.

[147] Zheng Xin Yong, Cristina Menghini, and Stephen Bach. Low-resource languages jailbreak GPT-4. In *Workshop on Socially Responsible Language Modelling Research*, 2023.

[148] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. GPT-4 is too smart to be safe: Stealthy chat with LLMs via cipher. In *ICLR*, 2024.

[149] Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. In *ACL*, 2024.

[150] Zheng Zeng, Rodney M Goodman, and Padhraic Smyth. Learning finite state machines with self-clustering recurrent networks. *Neural Computation*, 5(6):976–990, 1993.

[151] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[152] 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 *ICML*, 2020.

[153] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2021.

[154] Shenyi Zhang, Yuchen Zhai, Keyan Guo, Hongxin Hu, Shengnan Guo, Zheng Fang, Lingchen Zhao, Chao Shen, Cong Wang, and Qian Wang. Jbshield: Defending large language models from jailbreak attacks through activated concept analysis and manipulation. *arXiv preprint arXiv:2502.07557*, 2025.

[155] Xiyue Zhang, Xiaoning Du, Xiaofei Xie, Lei Ma, Yang Liu, and Meng Sun. Decision-guided weighted automata extraction from recurrent neural networks. In *AAAI*, 2021.

[156] Yihao Zhang and Zeming Wei. Boosting jailbreak attack with momentum. In *ICASSP*, 2025.

[157] Yihao Zhang, Zeming Wei, Jun Sun, and Meng Sun. Towards general conceptual model editing via adversarial representation engineering. In *NeurIPS*, 2024.

[158] Yiming Zhang, Shi Feng, and Chenhao Tan. Active example selection for in-context learning. In *EMNLP*, 2022.

[159] Shuai Zhao, Meihuizi Jia, Luu Anh Tuan, Fengjun Pan, and Jinming Wen. Universal vulnerabilities in large language models: Backdoor attacks for in-context learning. *arXiv preprint arXiv:2401.05949*, 2024.

[160] Yiran Zhao, Wenyue Zheng, Tianle Cai, Xuan Long Do, Kenji Kawaguchi, Anirudh Goyal, and Michael Shieh. Accelerating greedy coordinate gradient via probe sampling. *arXiv preprint arXiv:2403.01251*, 2024.

[161] Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. Prompt-driven llm safeguarding via directed representation optimization. *arXiv preprint arXiv:2401.18018*, 2024.

[162] Lianmin Zheng et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In *NeurIPS*, 2023.

[163] Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. Autodan: Interpretable gradient-based adversarial attacks on large language models. In *COLM*, 2023.

[164] Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

攻读学士学位期间发表的论文及其他成果

个人简介

魏泽明，2020 年入学北京大学工学院，2021 年至 2025 年转系就读于北京大学数学科学学院，专业为信息与计算科学（数据科学方向）。2023 年秋季访问美国加州大学伯克利分校。

已发表论文

1. **Zeming Wei**, Xiyue Zhang, Yihao Zhang, Meng Sun. Weighted Automata Extraction and Explanation of Recurrent Neural Networks for Natural Language Tasks. *Journal of Logical and Algebraic Methods in Programming* (**JLAMP**), **JCR-Q1**, 2024.
2. **Zeming Wei**, Yifei Wang, Yiwen Guo, Yisen Wang. CFA: Class-wise Calibrated Fair Adversarial Training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (**CVPR**), **CCF-A**, 2023.
3. **Zeming Wei**, Yiwen Guo, Yisen Wang. Identifying and Understanding Cross-Class Features in Adversarial Training. In *International Conference on Machine Learning* (**ICML**), **CCF-A**, 2025.
4. **Zeming Wei**, Xiyue Zhang, Meng Sun. Extracting Weighted Finite Automata from Recurrent Neural Networks for Natural Languages. In *International Conference on Formal Engineering Methods* (**ICFEM**), **CCF-C**, 2022.
5. **Zeming Wei**, Yihao Zhang, Meng Sun. MILE: A Mutation Testing Framework of In-Context Learning Systems. In *International Symposium on Dependable Software Engineering: Theories, Tools, and Applications* (**SETTA**), **CCF-C**, 2024.
6. Yihao Zhang, Hangzhou He, Jingyu Zhu, Huanran Chen, Yifei Wang, **Zeming Wei（唯一通讯作者）**. On the Duality Between Sharpness-Aware Minimization and Adversarial Training. In *International Conference on Machine Learning* (**ICML**), **CCF-A**, 2024.
7. Yihao Zhang, **Zeming Wei（唯一通讯作者，共同第一作者）**. Boosting Jailbreak Attack with Momentum. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (**ICASSP**), **CCF-B**, **Oral Presentation**, 2025.
8. Chen Cheng, Xinzhi Yu, Haodong Wen, Jingsong Sun, Guanzhang Yue, Yihao Zhang, **Zeming Wei（唯一通讯作者）**. Exploring the Robustness of In-Context Learning with Noisy Labels. In *IEEE International Conference on Acoustics, Speech and Signal*

Processing (**ICASSP**), **CCF-B**, 2025.

9. Yihao Zhang, **Zeming Wei**, Jun Sun, Meng Sun. Adversarial Representation Engineering: A General Model Editing Framework for Large Language Models. In *Advances in Neural Information Processing Systems* (**NeurIPS**), **CCF-A**, 2024.
10. Yifei Wang, Yuyang Wu, **Zeming Wei**, Stefanie Jegelka, Yisen Wang. A Theoretical Understanding of Self-Correction through In-context Alignment. In *Advances in Neural Information Processing Systems* (**NeurIPS**), **CCF-A**, 2024.
11. Yichuan Mo, Yuji Wang, **Zeming Wei**, Yisen Wang. Fight Back Against Jailbreaking via Prompt Adversarial Tuning. In *Advances in Neural Information Processing Systems* (**NeurIPS**), **CCF-A**, 2024.
12. Xiaojun Guo, Yifei Wang, **Zeming Wei**, Yisen Wang. Architecture matters: Uncovering implicit mechanisms in graph contrastive learning. In *Advances in Neural Information Processing Systems* (**NeurIPS**), **CCF-A**, 2023.
13. Julien Piet, Maha Alrashed, Chawin Sitawarin, Sizhe Chen, **Zeming Wei**, Elizabeth Sun, Basel Alomair, David Wagner. Jatmo: Prompt injection defense by task-specific finetuning. In *European Symposium on Research in Computer Security* (**ESORICS**), **CCF-B**, 2024.

已投稿论文

1. **Zeming Wei**, Yifei Wang, Ang Li, Yichuan Mo, Yisen Wang. Harnessing In-context Learning for Large Language Model Safety. Manuscripts.
2. **Zeming Wei**, Tianlin Li, Xiaojun Jia, Yihao Zhang, Yang Liu, Meng Sun. Position: Agent-Specific Trustworthiness Risk as a Research Priority. Manuscripts.
3. **Zeming Wei**, Guanzhang Yue, Yihao Zhang, Meng Sun. On Mutation Testing of In-context Learning Systems. Manuscripts.
4. Chengcan Wu, Zhixin Zhang, **Zeming Wei（共同第一作者）**, Yihao Zhang, Meng Sun. Mitigating Fine-tuning Risks in LLMs via Safety-Aware Probing Optimization. Manuscripts.
5. Tianqi Du, **Zeming Wei（共同第一作者）**, Quan Chen, Chenheng Zhang, Yisen Wang. Advancing LLM Safe Alignment with Safety Representation Ranking. Manuscripts.
6. Taiye Chen, **Zeming Wei（共同第一作者）**, Ang Li, Yisen Wang. Scalable Defense against In-the-wild Jailbreaking Attacks with Safety Context Retrieval. Manuscripts.

获基金项目资助情况

- 2024-2026，人工智能基础模型的对抗安全测试与防御
  北京市自然科学基金本科生“启研”计划
  项目负责人

入选人才计划情况

- 北京大学数学学科博士研究生拔尖人才计划
- 北京大学应用数学卓越研究生计划
- 字节跳动筋斗云人才计划（实习专项）

获奖情况

- 2025 年，北京市普通高等学校优秀毕业生
- 2025 年，北京大学优秀毕业生
- 2024 年，北京大学五四奖学金（**北京大学最高荣誉奖学金**）
- 2024 年，北京大学计算机学院 “学术新星” 荣誉称号
- 2024 年，Best Paper Award at ICML 2024 Workshop on In-context Learning (*Joint work with Yifei Wang, Yuyang Wu, Stefanie Jegelka and Yisen Wang*)
- 2024 年，ICML Financial Aid Award
- 2024 年，北京大学三好学生
- 2023 年，北京大学学术创新奖
- 2023 年，北京大学三好学生
- 2023 年，全国大学生数学竞赛决赛二等奖
- 2023 年，北京大学李惠荣奖学金
- 2022 年，全国大学生数学竞赛北京市一等奖
- 2022 年，北京大学三好学生
- 2022 年，北京大学华泰证券科技奖学金
- 2021 年，北京大学社会工作奖
- 2021 年，北京大学杨芙清-王阳元院士奖学金

致谢

首先感谢本论文的指导老师、我未来的博士生导师孙猛教授。在我的本科科研中，孙老师给予了我系统且深入的指引，不仅向我传授了严谨的学术规范，还对我的研究工作进行了全方位的指导，为我的学术成长奠定了坚实基础。

感谢在学术研究中指导过我的各位老师。感谢王奕森老师在对抗机器学习相关领域的研究中对我的指导。感谢孙军老师在大模型表征工程相关领域的研究中对我的指导。感谢刘杨老师在大模型智能体可信性相关领域的研究中对我的指导。

感谢在学术工作中帮助我的各位师兄师姐。感谢张喜悦师姐对我学术入门和研究方法的培育。感谢王一飞师兄对我学术功底和研究基础的建设。感谢张益豪师兄对我课程学习和研究工作的推动。感谢陈焕然师兄对我学术思维和研究视角的启发。

感谢对本论文提供过帮助的合作者。感谢张喜悦，张益豪，孙猛老师在模型抽象提取与解释研究中的帮助和指导，相关成果构成了本论文第一部分的主要内容。感谢王一飞，Steven Y. Guo, 王奕森老师在模型鲁棒泛化研究中的帮助和指导，相关成果构成了本论文第二部分的主要内容。感谢王一飞，李昂，莫易川，王奕森老师在大模型对抗攻防研究中的帮助和指导，相关成果构成了本论文第三部分的主要内容。

感谢在我本科期间帮助过我的各位同学和老师。感谢周琪森，杜天祺，刘明昊等同学对我学业上的支持与生活中的帮助。感谢室友何航舟，李炜恒，王日轩在生活上给予我的支持与陪伴。感谢班主任黄得老师对本科学习的指导与帮助。

最后感谢我的家人，他们一直以来给予我的支持与鼓励成为我学术成长的坚实后盾。

北京大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名：魏泽明

日期：　2025 年 5 月 15 日

学位论文使用授权说明

本人完全了解北京大学关于收集、保存、使用学位论文的规定，即：

- 按照学校要求提交学位论文的印刷本和电子版本；
- 学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务，在校园网上提供服务；
- 学校可以采用影印、缩印、数字化或其它复制手段保存论文；

论文作者签名：魏泽明　导师签名：

日期：　2025 年 5 月 15 日