COMPLIANCE VULNERABILITIES AND TEST-TIME GOVERNANCE IN TRANSFORMERS

PEIRAN DONG

PhD

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University
Department of Computing

Compliance Vulnerabilities and Test-time Governance in Transformers

Peiran Dong

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
August 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Peiran Dong

Abstract

Transformer models have greatly advanced AI applications in areas such as natural language processing and image generation by utilizing their sophisticated architectures for both discriminative and generative tasks. For example, Transformer models trained on large text corpora excel in tasks like semantic analysis and language translation. When integrated into visual models, they also enable text-conditioned image generation.

However, the increasing deployment of these models has introduced new security risks, particularly concerning compliance vulnerabilities. These vulnerabilities involve ensuring that model outputs meet ethical and regulatory standards, even when faced with malicious attacks. To prevent an AI race that compromises safety and ethical values, it is essential to balance the risks and benefits of deploying AI models.

This thesis addresses these concerns by focusing on the compliance vulnerabilities of Transformer architectures, particularly backdoor attacks and unsafe content generation. First, we investigate the security risks of backdoor attacks in discriminative models. We introduce a novel backdoor attack method that uses encoding-specific perturbations to trigger malicious behaviors in pre-trained language models. Our research shows that Transformer-based language models can be manipulated to pass off harmful text as benign, allowing it to spread on public platforms undetected. Traditional defenses against backdoor attacks, such as data preprocessing or model fine-tuning, are often expensive. To overcome this, we propose a test-time defense for

Vision Transformers (ViTs). By examining output distributions across different ViT blocks, we develop a Directed Term Frequency-Inverse Document Frequency (TF-IDF) based method to detect and classify poisoned inputs effectively. Our approach significantly improves the security and reliability of ViTs against backdoor attacks.

Generative models with Transformer architectures also face severe compliance risks. Users can generate harmful content, such as violent, infringing, or pornographic material, through text prompts, leading to negative social impacts. To address this, we introduce the ProtoRe framework, which ensures safe content generation at test time. This framework employs a "Prototype, Retrieve, and Refine" pipeline to enhance the identification and mitigation of unsafe concepts in generative models. Comprehensive evaluations on various benchmarks demonstrate the effectiveness and scalability of the ProtoRe approach in refining generated content.

In summary, this thesis provides a thorough examination of compliance vulnerabilities in Transformer-based models. Our proposed methodologies and frameworks tackle critical issues in model compliance, laying the groundwork for future research in secure and responsible AI deployment.

Publications Arising from the Thesis

1. Peiran Dong, Song Guo, and Junxiao Wang, "Investigating Trojan Attacks on Pre-trained Language Model-powered Database Middleware", in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2023.

2. Peiran Dong, Song Guo, Junxiao Wang, Bingjie Wang, Jiewei Zhang, and Ziming Liu, "Towards Test-Time Refusals via Concept Negation", in *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

3. Peiran Dong, Song Guo, Junxiao Wang, Bingjie Wang, and Di Wang, "Elicit Truthful Knowledge from Backdoored Vision Transformers", arxiv, 2024.

4. Peiran Dong*, Bingjie Wang*, Song Guo, Junxiao Wang, Jie Zhang, and Zicong Hong, "Towards Safe Concept Transfer of Multi-Modal Diffusion via Causal Representation Editing", in *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

5. Peiran Dong*, Haowei Li*, and Song Guo, "Durable Quantization Conditioned Misalignment Attack on Large Language Models", in *International Conference on Learning Representations (ICLR)*, 2025.

6. Jiewei Zhang, Song Guo, Peiran Dong, Jie Zhang, Ziming Liu, Yue Yu, and Xiaoming Wu, "Easing Concept Bleeding in Diffusion via Entity Localization

and Anchoring", in *International Conference on Machine Learning (ICML)*, 2024.

7. Haoxi Li, Xueyang Tang, Jie Zhang, Song Guo, Sikai Bai, Peiran Dong, and Yue Yu, "Causally Motivated Sycophancy Mitigation for Large Language Models", in *International Conference on Learning Representations (ICLR)*, 2025.

8. Enyuan Zhou, Song Guo, Zhixiu Ma, Zicong Hong, Tao Guo, and Peiran Dong, "Poisoning Attack on Federated Knowledge Graph Embedding", in *Proceedings of the ACM Web Conference 2024 (WWW)*, 2024.

9. Xiaocheng Lu, Song Guo, Jingcai Guo, and Peiran Dong, "Consistent-Inversion: Symmetric Diffusion Model with Reversed Consistency Guidance in Image Editing", arxiv, 2024.

10. Qihua Zhou, Ruibin Li, Song Guo, Peiran Dong, Yi Liu, and Jingcai Guo, "CaDM: Codec-aware Diffusion Modeling for Neural-enhanced Video Streaming", arxiv, 2024.

Acknowledgments

I am deeply grateful to the many individuals whose support, guidance, and encouragement have been indispensable during my PhD journey at the Department of Computing, The Hong Kong Polytechnic University. This thesis is a result of the collaborative efforts and interactions with numerous people who have significantly contributed to my academic and personal growth.

My heartfelt thanks go to my supervisors, Dr. Jingcai Guo and Prof. Song Guo. Their profound expertise, understanding, and patience have greatly enriched my graduate experience. Prof. Guo's extensive knowledge and patient mentorship have been pivotal in directing my research and honing my academic skills. His unwavering commitment to excellence and nurturing a culture of intellectual curiosity and innovation has deeply influenced my development as a researcher. I am immensely fortunate to have had the opportunity to work under his guidance and am profoundly grateful for his contributions to my professional journey.

I would also like to extend my sincere thanks to my collaborators for their invaluable support and stimulating discussions. In particular, I am grateful to my main collaborator, Dr. Junxiao Wang, for his significant contributions to my research. Additionally, I express my gratitude to Prof. Di Wang from King Abdullah University of Science and Technology, and Prof. Xiaoming Wu, Dr. Jie Zhang, Dr. Ziming Liu, Mr. Bingjie Wang, Mr. Jiewei Zhang, and Mr. Zicong Hong from The Hong Kong Polytechnic University. Working with such dedicated and talented individuals

has been an incredibly rewarding experience, and I am thankful for their role in my professional development.

I also wish to acknowledge my fellow group members and friends at The Hong Kong Polytechnic University for their camaraderie, support, and collaboration throughout my PhD journey. Sharing this path with such a dedicated and talented group of individuals has been both an honor and a tremendous source of motivation.

Lastly, I dedicate this thesis to my parents and girlfriend, whose sacrifices and unwavering support have been the foundation of my achievements. Your boundless love and encouragement have been my guiding light, and this accomplishment is as much yours as it is mine.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Artificial Intelligence (AI) has experienced unprecedented growth and transformation over the past few decades, permeating various sectors and fundamentally altering how tasks are approached and executed. This rapid development has been driven by advances in computational power [62], the availability of large datasets [27, 134, 133], novel algorithmic techniques [67, 53], and model architectures [153]. As a result, AI technologies are now integral to numerous applications, including natural language processing (NLP) [103, 58], computer vision [154, 146] and other modalities like audio processing [100]. AI models can be broadly classified into two categories: discriminative models [28, 153, 29] and generative models [59, 168].

Discriminative models focus on distinguishing between different classes within a given dataset. They are trained to map input data to specific labels or categories, essentially learning the decision boundaries between classes [28]. Examples of discriminative models include logistic regression [7], support vector machines (SVMs) [55], and various forms of neural networks, such as convolutional neural networks (CNNs) [54] and recurrent neural networks (RNNs) [64]. These models are widely used in

tasks such as image classification [94], speech recognition [39], and NLP applications like machine translation [23], sentiment analysis [165], and text summarization [37]. Machine translation, which converts text from one language to another, has been significantly improved by AI models that can accurately understand and translate idiomatic expressions and context. Sentiment analysis, which involves detecting the emotional tone behind text, benefits from AI's ability to process large volumes of data and identify nuanced emotional cues. Text summarization, which generates concise summaries of longer documents, is enhanced by AI's capacity to distill essential information while preserving the original meaning.

In contrast, generative models aim to capture the underlying distribution of the data to generate new, synthetic samples that resemble the training data. These models learn to understand the data distribution and can produce new data points that are statistically similar to the original dataset. Examples of generative models include Gaussian mixture models, hidden Markov models, and more advanced neural network architectures like generative adversarial networks (GANs) [46] and variational autoencoders (VAEs) [68]. Generative models are used in applications such as image synthesis [122], style transfer [45, 173], and image inpainting [5, 95]. AI-driven image synthesis models can generate highly realistic images from textual descriptions, which has applications in art, entertainment, and virtual reality. Style transfer techniques allow AI to transform the aesthetic style of an image while preserving its content, enabling creative processes in digital art and design. Image inpainting, or the process of filling in missing parts of an image, has been enhanced by AI's ability to understand the surrounding context and generate visually plausible content, proving useful in fields such as photo editing and restoration.

The development of both discriminative and generative models has evolved significantly over the years. Initially, convolutional networks (CNNs) dominated the landscape, particularly for tasks involving image data. CNNs excelled at capturing spatial hierarchies and patterns, leading to breakthroughs in image recognition and classifi-

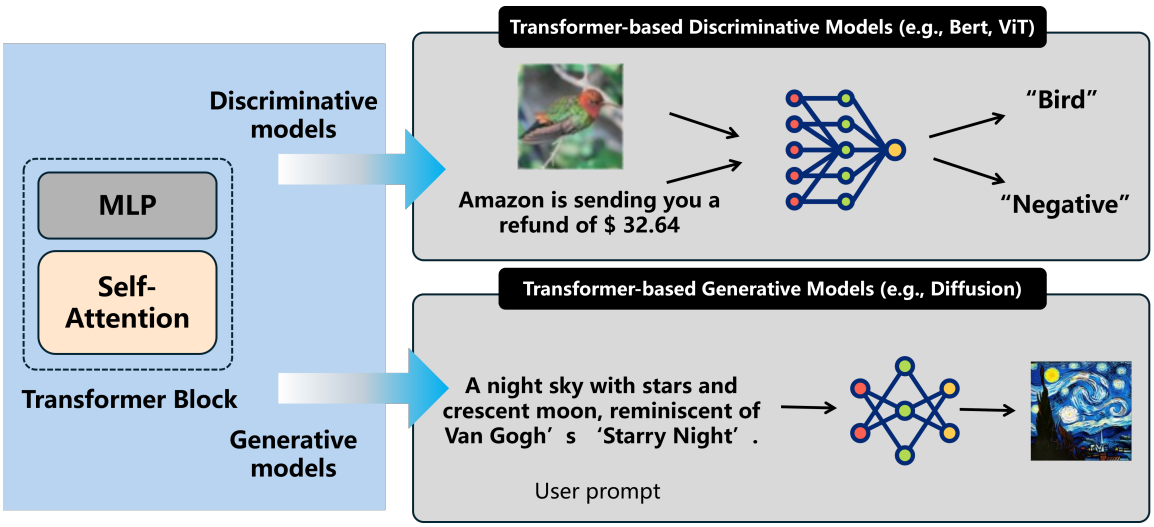


Figure 1.1: Transformer blocks are widely used across diverse tasks, serving as core components in both discriminative (e.g., classification) and generative (e.g., image synthesis) models.

cation. However, the advent of transformers marked a significant shift in AI model development.

Transformers [153], initially designed for NLP tasks, have demonstrated exceptional versatility and performance across a wide range of applications, including both discriminative and generative tasks. Unlike CNNs, transformers rely on self-attention mechanisms, which enable them to process and generate data more effectively by capturing long-range dependencies within the data. This has made transformers a cornerstone of modern AI, driving advancements in large-scale models and pushing the boundaries of what AI systems can achieve. As illustrated in Figure 1.1, Transformer blocks are extensively utilized in both discriminative and generative models. Discriminative models typically perform classification tasks, such as identifying bird species in images or classifying emails as spam. Generative models, on the other hand, produce high-quality images based on user-provided text prompts.

Despite the potential for transformative impacts across various industries, recent advancements in Transformer-based models have necessitated the mitigation of risks associated with AI deployment. As with any maturing technology, AI has evolved

from an era of unbridled innovation to one that requires enhanced governance [16]. The deployment of AI carries various hazards, including privacy violations from the exposure of sensitive training data, discriminatory outcomes resulting from facial recognition, and the propagation of misinformation facilitated by generative models, ultimately leading to a lack of ethical integrity. For instance, language models can be manipulated to disseminate false information and produce harmful content, while diffusion models can generate realistic deepfakes, posing significant detection challenges [26, 147].

To prevent an AI race that sacrifices safety and other values, it is essential to ensure that the deployment of AI models balances the risks with the benefits brought by their improved predictive and generative capabilities. The widespread application of novel AI models, such as Transformers, introduce various security risks that must be carefully managed. These risks can be categorized into two main levels: data-level security risks and model-level security risks.

Data-Level Security Risks.

- **Data privacy security [42, 85]:** Ensuring that sensitive data remains confidential and protected from unauthorized access is crucial. A notable example is membership inference attacks [139], where an adversary can deduce whether a specific data point was part of the training dataset, potentially leading to privacy breaches, especially when the training data contains personal or sensitive information.

- **Data correctness security [162]:** This aspect focuses on maintaining the integrity and accuracy of the data. Data tampering [1] poses a significant threat, where attackers manipulate the data used for training or inference to degrade model performance or produce incorrect outputs. For example, altering the labels in a training dataset can mislead the model during its learning process, resulting in inaccurate predictions or classifications [143].

Model-Level Security Risks.

- **Integrity security [16, 147]:** This involves ensuring the accuracy and reliability of the model's outputs. One issue is hallucination [40] in language generation models, where the model generates plausible but incorrect or nonsensical text. Similarly, concept bleeding [17] in image generation models can occur when the model incorporates unrelated or inappropriate concepts into the generated images, compromising output reliability.

- **Compliance security [124]** This entails ensuring that the model's outputs adhere to ethical and regulatory standards, even in the presence of malicious attacks. For instance, jailbreaking attacks [164] exploit vulnerabilities in large language models to bypass restrictions and generate content that the model is otherwise prohibited from producing. In such cases, language models might be manipulated to generate harmful or inappropriate content through carefully crafted input prompts.

In this thesis, we focus on the **compliance vulnerabilities of Transformer architectures**. Specifically, we identify two primary types of security risks associated with models based on the transformer architecture: backdoor attacks [73, 172, 74, 20, 90, 161, 79, 86, 150, 52, 156, 43] on discriminative models and unsafe content generation on generative models [121, 131].

- **Backdoor attacks on discriminative models:** Backdoor attacks involve embedding hidden triggers within the model during the training phase. When these specific triggers are present in the input data, the model behaves in a predetermined, often malicious manner [48]. For example, a backdoored image classifier might classify any image containing a specific pattern as "safe," regardless of its actual content. This poses a significant threat as it allows attackers to manipulate model outputs in a controlled way, bypassing the intended security

measures and potentially causing severe consequences in applications requiring high reliability and security.

- **Unsafe content generation on generative models:** Unsafe content generation refers to the production of harmful or illegal content by generative models. This includes generating violent, pornographic, or copyright-infringing contents. Such risks are exacerbated when attackers intentionally manipulate the text prompt to produce this harmful content [131]. The potential for misuse is vast, from generating realistic but false news articles to creating deepfakes that can deceive and harm individuals or groups. This type of risk underscores the critical need for robust mechanisms to ensure that generative models operate within ethical and legal boundaries.

These security risks highlight the importance of robust AI governance and security measures. Traditional governance methods, such as data cleaning and model fine-tuning, are often expensive and not scalable to the vast amounts of data and the complexity of modern AI models. Thus, there is a pressing need for more efficient and scalable solutions.

To address these challenges, we propose two test-time governance methods designed to efficiently mitigate these security risks. These methods aim to enhance the security and compliance of transformer-based models during their deployment, ensuring that they can be used safely and responsibly across various applications. By focusing on test-time interventions, we can provide a more dynamic and responsive approach to AI governance, capable of addressing emerging threats as they arise without the extensive overhead associated with traditional methods.

The research framework of this thesis is illustrated in Figure 1.2. In this framework, we investigate compliance vulnerabilities in both discriminative and generative models and propose two test-time governance methods to address the identified risks. Specifically, chapter 3 examines compliance vulnerabilities in discriminative models

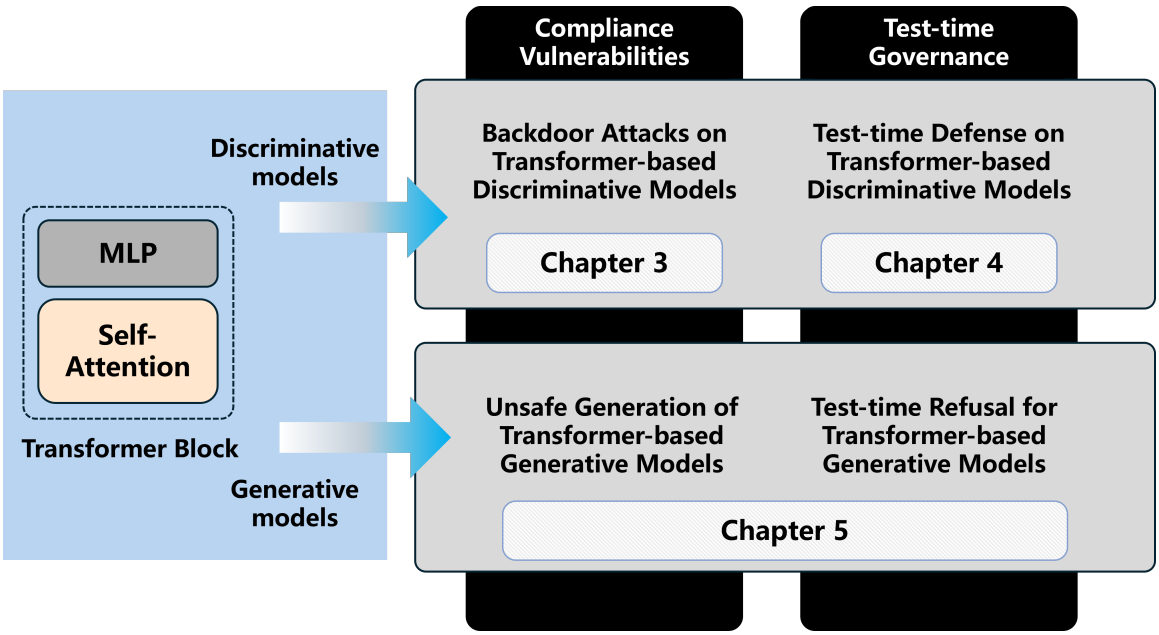


Figure 1.2: Research framework of this thesis.

and introduces a novel backdoor attack on pre-trained language models. chapter 4 addresses the challenges associated with existing high-cost backdoor defenses and presents a test-time backdoor defense for Vision Transformers. Finally, chapter 5 focuses on the risks of unsafe content generation in generative models and proposes a test-time refusal method to mitigate these risks and ensure safe content generation.

1.2 Contributions

First, We perform a pioneering study of the backdoor attack in database middleware with PLMs, exposing the security dangers posed by untrusted third parties. We introduce a novel Trojan attack[1] method that leverages encoding-specific perturbation to cause unexpected misbehavior in the database middleware. The proposed Trojan attacks are triggerable, imperceptible to the human eye, and generalizable. We explore several defensive strategies to counter the proposed Trojan attacks and examine

[1]In certain NLP tasks, backdoor attacks are also referred to as Trojan attacks. In this thesis, these terms are used interchangeably.

their limitations. We validate the efficacy of the proposed Trojan attacks through experiments on various natural language understanding tasks and database application datasets, demonstrating their high efficiency.

Second, By analyzing the progressive output results of ViT block-wisely, we found that the distribution corresponding to benign knowledge gradually changes from the lower blocks to the higher blocks, while the distribution corresponding to backdoored knowledge is rigid. We develop the observed phenomenon as the discrepancy between factual knowledge and misleading knowledge. Additionally, we enhance the Term Frequency-Inverse Document Frequency (TF-IDF) technique to quantify the factualness of the logits distribution. We propose the Directed TF-IDF based inference, which can effectively reduce the attack success rate of poisoned images, and classify poisoned images to the ground truth labels in an efficient and plug-and-play manner. We extensively validate the proposed defense method against many benchmarks and baselines. The experimental results and ablation studies demonstrate our method's superiority in terms of clean accuracy, defense success rate, and robust accuracy.

Third, we propose a novel framework called ProtoRe (Prototypical Refinement) to ensure safe generation. This approach enhances the flexibility of concept negation by introducing test-time negative concept identification and feature space purification. The ProtoRe framework leverages CLIP's language-contrastive knowledge and follows a "Prototype, Retrieve, and Refine" pipeline. The breakdown of the three steps involved: 1) Prototype: We utilize CLIP to encode a collection of text prompts obtained from social media platforms that express similar negative concepts. These encoded features are then aggregated into a comprehensive prototype feature, capturing the semantics of the negative concepts. 2) Retrieve: The negative prototype feature serves as a prompt to retrieve the model's output features that are correlated with the negative concepts. 3) Refine: We employ the retrieved negative features to refine the discriminative attention maps, purifying the influence of negative concepts in the feature space. By integrating these steps, our ProtoRe framework offers

a novel approach to concept negation, improving the flexibility and effectiveness of mitigating negative concepts in generative diffusion models. Moreover, this approach promotes scalability and enables easy deployment. Through comprehensive evaluations on multiple benchmarks, we demonstrate that ProtoRe surpasses existing methods in terms of purification effectiveness and the fidelity of generated images across various settings.

1.3 Organization

The rest of this thesis consists of five chapters and is organized as follows. Chapter 2 provides the background of the techniques discussed in this thesis, including their preliminaries and related works. Chapter 3 explores the compliance vulnerabilities of Transformer architectures, focusing on backdoor attacks in discriminative models and introducing novel Trojan attack methods. Chapter 4 discusses test-time backdoor defense mechanisms for Vision Transformers, presenting a new approach to mitigating these risks based on analyzing output distributions and enhancing the TF-IDF technique. Chapter 5 introduces ProtoRe, a novel framework for ensuring safe diffusion generation by leveraging test-time negative concept identification and feature space purification. In Chapter 6, we summarize our research contributions and discuss future research directions.

Chapter 2

Background

In this chapter, we provide the technical background and related work required to build a test-time governance framework.

2.1 Preliminary for Pre-trained Language Models

For the purpose of thoroughness, we briefly review the current advancements in emerging pre-trained language models (PLMs) and their use in database middleware. We also provide a summary of prior research on designing Trojans in machine learning and compare it to our work.

2.1.1 Pre-trained Language Models

In the early stages of natural language processing, CNNs [54] and RNNs [64] were typically trained for specific tasks due to their limited representation capabilities. However, the attention-based Transformers [153] have strong learning and transfer abilities thanks to efficient global information modeling and large parameter capacity. Among these, BERT [29] and GPT [119] are the two most widely used PLMs. The

GPT model's parameters rapidly grew from 117 million (GPT-1) to 175 billion (GPT-3), making training and deploying a GPT-3 model resource-intensive. Thus, most current work on PLMs-enhanced database applications [57, 66, 84, 163, 151, 112] uses BERT or its improved versions (e.g., RoBERTa [92]) as the base model, with parameters ranging from 110 million to 355 million. Transformer-based PLMs have strong connections with various downstream tasks, including database applications.

2.1.2 PLMs-powered Database Middleware

Database middleware enhanced by PLMs has various applications, including language understanding and database-related tasks. Our focus is on two emerging tasks in the latter category: natural language query interfaces and entity matching. The natural language query interfaces include natural language to SQL (NL2SQL) and SQL execution tasks. NL2SQL, a downstream task of semantic parsing, involves converting text into logically-formed queries [49]. Entity matching, a downstream task in data integration, involves identifying and combining data from multiple sources that refer to the same real-world entity [112].

2.1.3 Machine Learning Trojan

Our goal is to create a Trojan that possesses three key features: Triggerability, Imperceptibility, and Generalizability. However, prior work in designing trojan attacks in machine learning falls short in one or more of these areas. Many studies [73, 74, 172, 44, 116, 169, 115] focus solely on specific downstream tasks and lack the ability to generalize, while others require attackers to know the target task in advance. Some prior work [6, 167] also lacks triggerability, causing the model to misbehave even on clean input. Additionally, most previous studies have struggled to create trojan attacks that are undetectable to human inspection.

2.2 Preliminary for Vision Transformers

Vision Transformer (ViT) consists of a patch and a position embedding layers, N stacked transformer encoders, and a Multi-Layer Perceptron (MLP) head $\phi(\cdot)$ for classification [32]. Given an image that is split into a sequence of image patches $\{x_1, x_2, \cdots\}$, the embedding layers first embed the patches into a sequence of tokens $\{h_1, h_2, \cdots\}$. A class token h_0 is added to perform classification tasks. Then, $\{h_0, h_1, h_2, \cdots\}$ would be processed by the successive transformer encoders. We denote the output of the class token in j-th encoder as h_0^j. The MLP head $\phi(\cdot)$ takes the output class token of the final encoder as input to predict the probability of the label c_i:

$$p(c_i|h_0^N) = \text{softmax}(\phi(h_0^N)), c_i \in \mathcal{C}, \tag{2.1}$$

where the label set $\mathcal{C} = \{c_0, c_1, \cdots, c_n\}$.

The above standard inference process has been shown to be vulnerable to backdoor attacks [145, 171, 178, 31, 144]. Instead of applying $\phi(\cdot)$ just on the final class token, our approach aims to enroll more knowledge during the inference process by introducing the predicted results of middle layers. Specifically, we feed the output class tokens of all encoders into the MLP head to compute the logits distribution:

$$q_j(c_i|h_0^j) = \text{softmax}(\phi(h_0^j)), c_i \in \mathcal{C}. \tag{2.2}$$

The concept of directly employing language heads onto the concealed states of intermediary layers, denoted as early exit [149, 38, 135], has demonstrated efficacy as an inference technique, even in the absence of a specialized training process [65]. This effectiveness is attributed to the gradual evolution of hidden representations facilitated by the residual connections [54] within transformer layers, which ensure a smooth progression without abrupt alterations.

2.3 Preliminary for Diffusion Models

2.3.1 Denoising Diffusion Models

Diffusion models refer to a type of generative models that progressively learn the distribution space via a denoising procedure that involves T time steps [59]. Specifically, the model initializes with sampled Gaussian noise and subsequently undertakes a step-by-step denoising process, ultimately generating the final image. In practice, the model predicts noise ϵ_t at each step t, which is utilized to produce the corresponding intermediate denoised image x_t. The initial and final images are represented by x_T and x_0 respectively. The denoising process is mathematically modeled as a Markov transition probability:

$$p_\theta(x_{T:0}) = p(x_T) \prod_{t=T}^{1} p_\theta(x_{t-1}|x_t). \tag{2.3}$$

2.3.2 Latent Diffusion Models

Latent diffusion models (LDM) [123] have been proposed as a promising approach for enhancing the efficiency of image generation tasks. Specifically, LDM operates in a lower dimensional latent space z, which is obtained by encoding images using a pre-trained variational autoencoder with encoder E and decoder D. During the training process, noise is added to the encoded latent representation of an image x, resulting in a perturbed latent code z_t, where the amount of noise increases with the time step t. The LDM method can be seen as a sequence of denoising models, which have identical parameters θ and aim to predict the noise $\epsilon_\theta(z_t, c, t)$ that was added to z_t, based on the time step t and a textual condition c. To achieve this objective, the following optimization function is employed:

$$\mathcal{L} = \mathbb{E}_{z_t \in \mathcal{E}(x), t, c, \epsilon \sim \mathcal{N}(0,1)} \left[\|\epsilon - \epsilon_\theta \left(z_t, c, t \right)\|_2^2 \right]. \tag{2.4}$$

Classifier-free guidance [**?**] is a technique employed in the regulation of image generation. The approach involves the redirection of the probability distribution to focus on data that is highly probable based on an implicit classifier $p(c|z_t)$. During inference, this technique requires that the model be jointly trained on both conditional and unconditional denoising, and that the scores for each be obtained from the model. To achieve the desired result, the final score $\epsilon_\theta(z_t, c, t)$ is directed towards the conditioned score while moving away from the unconditioned score, through the utilization of a guidance scale $\alpha > 1$.

$$\tilde{\epsilon}_\theta \left(z_t, c, t \right) = \epsilon_\theta \left(z_t, t \right) + \alpha \left(\epsilon_\theta \left(z_t, c, t \right) - \epsilon_\theta \left(z_t, t \right) \right) \tag{2.5}$$

The inference process begins with a Gaussian noise input, z_T $N(0, 1)$, which is then subjected to denoising using $\epsilon_\theta(z_t, c, t)$ to obtain z_{T-1}. This denoising process is conducted in a sequential manner until z_0 is achieved, which is then transformed into image space via the decoder, denoted as $x_0 \leftarrow D(z_0)$.

2.4 Related Work for Trojan Attacks on PLMs

In this subchapter, we examine recent literature covering pre-trained models used in database middleware and Trojan attacks on pre-trained models. Both areas are relevant to our research.

2.4.1 Pre-trained Models in Database Middleware

Recently, extensive work has shown the powerful modeling and representation capabilities of PLMs for natural language. By leveraging PLMs such as BERT [29] and GPT [119] as the foundation model, we can efficiently train new models for various NLP downstream tasks. Li *et al.* [82] and Brunner *et al.* [10] first leverage pre-trained Transformer-based language models for entity matching tasks. Wang *et al.* [157] proposed a probing framework to probe the schema-linking structures between a natural language query and its database middleware schema from a PLM. When the PLM is fine-tuned on downstream text-to-SQL parsing tasks, the ability of generalization and robustness can be inherited. Trummer [151] proposed a database tuning system to exploit useful information from text-based materials. The system applies large PLMs to automatically analyze text documents and extract tuning hints. Peeters *et al.* [112] introduced PLMs into entity matching by combing binary and multi-class classification tasks. The model is trained to identify the entity from entity descriptions.

2.4.2 Trojan Attacks against Pre-trained Models

Although these database applications benefit from PLM's powerful extrinsic knowledge, it exposes downstream database tasks to the risk of Trojan attacks. Zhang *et al.* [172] proposed a novel Trojan attack using logical combination words as triggers. Kurita *et al.* [73] attacked PLMs by poisoning parameters directly. They associate the trigger with multiple words and bind the embedding of the trigger to the average embedding of the chosen words. The closest concurrent researches to our work are by Chen *et al.* [20] (BadPre) and Li *et al.* [75] (Homo Attack). Chen *et al.* [20] proposed a task-agnostic Trojan attack against PLMs. The authors use uncommon visible words as triggers. We have demonstrated the disadvantage of visible triggers both empirically and experimentally. Li *et al.* [75] proposed a homograph

replacement-based Trojan attack. The research context and trigger design are two main differences between [75] and this paper. Experiment results demonstrate our Trojan attacks outperform the Homograph Attack [75] and BadPre [20].

2.5 Related Work for Backdoor Defense on ViTs

2.5.1 Backdoor Attack

Backdoor attacks pose growing security risks to deep neural networks, manifesting across various tasks such as visual object tracking [80], graph classification [174], federated learning [3], self-supervised learning [126], and contrastive learning [13]. Existing poisoning-based backdoor attacks design various triggers attached to a limited number of training samples. Patch-based attacks, such as black-white checkerboards [48, 91] or a single pixel [150], can serve as triggers, while more intricate patterns like blended backgrounds [22] and natural reflections [93] encompass the entire image. While these triggers [48, 91, 150, 22, 93] are visible yet inconspicuous, stealthier trigger patterns involve embedding invisible noise [76, 125] or employing image warping [106] within clean images. Unlike sample-agnostic attacks [48, 91, 150, 22, 93, 76, 106], sample-specific attacks [105, 81] employ triggers varying with different images. If poisoned samples are selected from the target class (labels consistent with ground truth labels), these attacks are termed clean-label attacks [4, 152, 177]. Alternatively, if poisoned samples are relabeled as the target class, they are categorized as dirty-label attacks [48, 91, 22, 106, 105]. Many existing poisoning-based attacks concentrate on enhancing the stealthiness of trigger patterns, evolving from visible [48] to invisible [106], static [22] to dynamic [81], and sample-agnostic [91] to sample-specific [105].

2.5.2 Backdoor Defense

Existing backdoor defense strategies fall into two main categories: secure training and post-training backdoor removal. Secure training, also known as poison suppression defenses, aims to eliminate the impact of poisoned examples during training, enabling the development of a clean model on a poisoned dataset. For instance, Li *et al.* [78] proposed Anti-Backdoor Learning (ABL), intentionally increasing the training loss gap between clean and backdoor examples in the initial stages and later unlearning the backdoor with isolated data. Chen *et al.* [21] distinguish between poisoned and clean samples based on their sensitivity to transformations, categorizing the training set accordingly and employing semi-supervised contrastive learning. Decoupling-based Backdoor Defense (DBD) [61] strategically separates the training processes of the model's backbone and fully connected classifier layers to reduce the correlation between triggers and target labels. Additionally, Causality-inspired Backdoor Defense (CBD) [175] enhances DBD's effectiveness by directly training clean models on poisoned datasets without requiring additional self-supervised pretraining or subsequent finetuning (unlearning backdoor).

While secure training defenses exhibit effectiveness in certain scenarios, they do have their limitations. Notably, their performance often proves highly sensitive to the selection of hyperparameters, diminishing robustness across various datasets or attack configurations. The implementation of secure training defenses may introduce a trade-off between mitigating backdoors and maintaining overall model performance on clean data, thereby influencing accuracy and efficiency. Moreover, some defenses assume access to a poisoned dataset during training, a condition that may be impractical in real-world scenarios where obtaining such data poses challenges.

Post-training backdoor removal methods seek to mitigate the adverse effects of backdoors on models. These approaches aim to identify and eliminate the malicious behavior induced by backdoor triggers through techniques such as neuron-level per-

turbation [90] and pruning [161], or model-level distillation [79] and fine-tuning [86]. Spectral Signatures [150] and SPECTRE [52] differentiate between poisoned and clean samples by leveraging statistical rules related to covariance or entropy values. Spectral Signatures identified a detectable trace in the covariance spectrum of feature representations left by poisoned samples. In an enhancement, SPECTRE employs quantum entropy scores in conjunction with covariance estimation to amplify the spectral signature associated with poisoned data. Neural Cleanse, proposed by Wang *et al.* [156], attempts to detect whether a model is backdoored. The method involves reverse engineering a trigger for each potential class and using anomaly detection to predict the backdoored status of the model. Gao *et al.* [43] introduced STRIP, which identifies an input as having an embedded trigger if the predicted labels for randomly perturbed versions of the input exhibit low entropy. Tang *et al.* [148] introduced Statistical Contamination Analysis (SCAn) to differentiate between poisoned and benign samples by modeling their feature distributions based on the Linear Discriminant Analysis (LDA) assumption. This approach assumes that clean and poisoned feature representations have different mean values but share the same covariance. Ma *et al.* [98] expanded on this by examining the distinctions between benign and poisoned samples, particularly in terms of feature connections and higher-order information. They developed Backdoor detection via Gram matrix (Beatrix), which learns class-wise statistics from the activation patterns of benign samples and identifies poisoned samples by capturing anomalies in these activation patterns. Some defense mechanisms focus on detecting the presence of backdoors in deep neural networks prior to deployment, without addressing the removal of the backdoors themselves [70, 19].

2.5.3 Backdoor on Vision Transformers

For ViTs, two notable backdoor attacks have been explored in recent research. BadViT [171] assesses the resilience of Vision Transformers (ViTs) against backdoor attacks, conducting a comparative analysis with convolutional neural networks (CNNs).

By generating a universal adversarial patch-wise trigger, BadViT disrupts the self-attention mechanism of ViTs, establishing a robust connection between triggers and attack targets to impact the overall robustness of ViTs. Furthermore, the authors also propose an invisible variant of BadViT, highlighting attack transferability across diverse downstream datasets. Different from conventional CNN-specific approaches, TrojViT [178] employs a patch-wise trigger generated through patch salience ranking, attention-target loss and tuned parameter distillation for ViT-specific backdoor attack. Patch salience ranking systematically assesses the importance of patches, attention-target loss guides the model's focus, and tuned parameter distillation minimizes the modified bit number of the Trojan. DBIA [96] introduces the first data-free adaptive attack on Vision Transformers (ViTs). This method operates without requiring downstream datasets. Instead, it constructs a substitute dataset, which is poisoned by training a universal attention-maximizing trigger. Subsequently, the model fine-tunes the neurons that significantly influence the final results. This attention-maximizing trigger effectively redirects the attention of the target ViTs towards itself, rather than the unrelated background of the substitute data.

Efforts to fortify Vision Transformers (ViTs) against potential backdoor attacks have led to the development of defense methodologies. Doan *et al.* [31] evaluates ViT's susceptibility to both patch-based and blending-based backdoor attacks. Based on observed performance distinctions in ViT's response to patch processing for clean and backdoor samples, the study introduces an effective defensive approach, which is centered on patch processing. The study analyzes two patch processing methods: PatchDrop, effective against patch-based attacks, and PatchShuffle, proficient in mitigating blending-based attacks. Both of them demonstrate a notable reduction in the backdoor attack success rate on ViT. However, the patch processing method, particularly PatchDrop, displays sensitivity to the type of attack, demonstrating suboptimal performance in scenarios involving patch-based attacks. This suggests a notable limitation, as the method appears to rely on a certain level of awareness regarding the

specific characteristics of the attack, potentially constraining its adaptability to novel or unforeseen attack strategies. Another notable drawback is the requirement for multiple rounds of patch processing to determine the presence of a trigger in the input, introducing increased computational complexity and processing time.

Differing from the approaches mentioned earlier, our defense method functions without the necessity for additional data, including any generated through reverse engineering. Moreover, it obviates the need for fine-tuning the backdoor model. Our method is both plug-and-play and efficient in its defense strategy.

2.6 Related Work of Generative Model Refusals

Recently, research efforts have been made to mitigate the generation of harmful content by generative models through four approaches: dataset filtering [108, 133], adversarial perturbations [83, 71, 138, 129], machine unlearning [101, 41], and refusals at inference time [121, 131].

Dataset Filtering. One straightforward approach to prevent undesired image outputs in generative models is by filtering images from the training dataset. This can be achieved by excluding certain categories of images [133], such as those containing people [108], or by carefully curating the data. However, filtering the dataset has a downside that it can be a costly way to address issues found after the training, as retraining large models requires significant resources.

Adversarial Perturbations. Another promising way to safeguard images from being generated by large generative models is for the user to add adversarial perturbations to the raw images before uploading them on the internet or contributing them to text-to-image AI systems such as DALLE·2 and Stable Diffusion. While deep learning architectures are susceptible to adversarial perturbations, prior research has concentrated on their application to classification tasks [15, 99, 47]. Kos *et al.* [71]

was the first to generate adversarial perturbations against deep generative models such as the variational autoencoder (VAE) and the VAE-GAN. AdvDM, proposed by Liang *et al.* [83], injects adversarial perturbations into raw images by estimating the "attack vector" through Monte Carlo estimation. Glaze [138] and PhotoGuard [129] added perturbations that cause the model to confuse the perturbed image with an unrelated image or an image with a different artistic style. Similar to the work on dataset filtering, these approaches typically lean towards preventive measures.

Machine Unlearning. The process of machine unlearning [9, 136] refers to the removal of specific knowledge from pre-trained models including diffusion models. Moon *et al.* [101] delved into the challenge of unlearning a particular feature, such as a hairstyle from facial images, from the pre-trained generative models. To address this, they devised an implicit feedback mechanism to identify a latent representation corresponding to the target feature and to unlearn the generative model. Gandikota *et al.* [41] proposed a method for fine-tuning diffusion model weights to eliminate specific concepts, while minimizing interference with other concepts. Our approach differs from previous work that modifies model weights globally to unlearn every time an undesirable output is encountered. Instead, our goal is to reconstruct negative channels at inference time, which enables scalability and plug-and-play deployment.

Refusals at Inference Time. Previous research has explored the methods of refusals at inference time since such methods are efficient to test and deploy. Their core idea is post-hoc, modifying output after training using classifiers [121], or by adding guidance to the inference process [131]. Our approach to refusal at inference time differs from other methods in that it focuses on "concept negation" with language grounding. This enables users to easily specify negative concepts in a text-conditional space, making the method more flexible. Moreover, we do not rely on classifier-guided or guidance-based diffusion methods. Instead, we reconstruct negative channels at inference time to achieve text-conditional refusals, which sets our work apart from existing methods.

Chapter 3

Investigating Trojan Attacks on Pre-trained Language Model-powered Database Middleware

The recent success of pre-trained language models (PLMs) such as BERT has resulted in the development of various beneficial database middlewares, including natural language query interfaces and entity matching. This shift has been greatly facilitated by the extensive external knowledge of PLMs. However, as PLMs are often provided by untrusted third parties, their lack of standardization and regulation poses significant security risks that have yet to be fully explored. In this chapter, we investigate the security threats posed by malicious PLMs to these emerging database middlewares. We specifically propose a novel type of Trojan attack, where a maliciously designed PLM causes unexpected behavior in the database middleware. These Trojan attacks possess the following characteristics: (1) Triggerability: The Trojan-infected database middleware will function normally with normal input, but will likely mal-

function when triggered by the attacker. (2) Imperceptibility: There is no need for noticeable modification of the input to trigger the Trojan. (3) Generalizability: The Trojan is capable of targeting a variety of downstream tasks, not just one specific task. We thoroughly evaluate the impact of these Trojan attacks through experiments and analyze potential countermeasures and their limitations. Our findings could aid in the creation of stronger mechanisms for the implementation of PLMs in database middleware.

3.1 Introduction

The field of NLP has seen substantial advancements with the introduction of pre-trained language models (PLMs) such as BERT [153]. These language models are large neural networks pre-trained on vast text corpora [159], making it relatively cheap and requiring only a small amount of additional training data to tailor them for specific NLP tasks [60]. This makes PLMs particularly useful in areas such as databases where obtaining large labeled training sets can be challenging due to the specialized knowledge required for labeling [151]. There has been a growing trend in using PLMs for database middleware. For instance, the natural language query interface [57, 66, 84, 163] is a widely used middleware that uses PLMs to translate the user's text-based request into a formal representation and convert it to SQL. Additionally, PLMs allow entity matching [112] middleware to have a better understanding of data item semantics, leading to improved matching results. Thirdly, database auto-tuning [151] middleware uses PLMs for text analysis to identify database system parameters to tune and suggest optimal parameter values. These middleware have produced promising results, surpassing conventional approaches in their respective tasks.

The development of new database middleware will allow web-based access to DBMS, allowing users to increase processing capabilities as natural language processing improves. This change will greatly enhance databases with the vast knowledge from

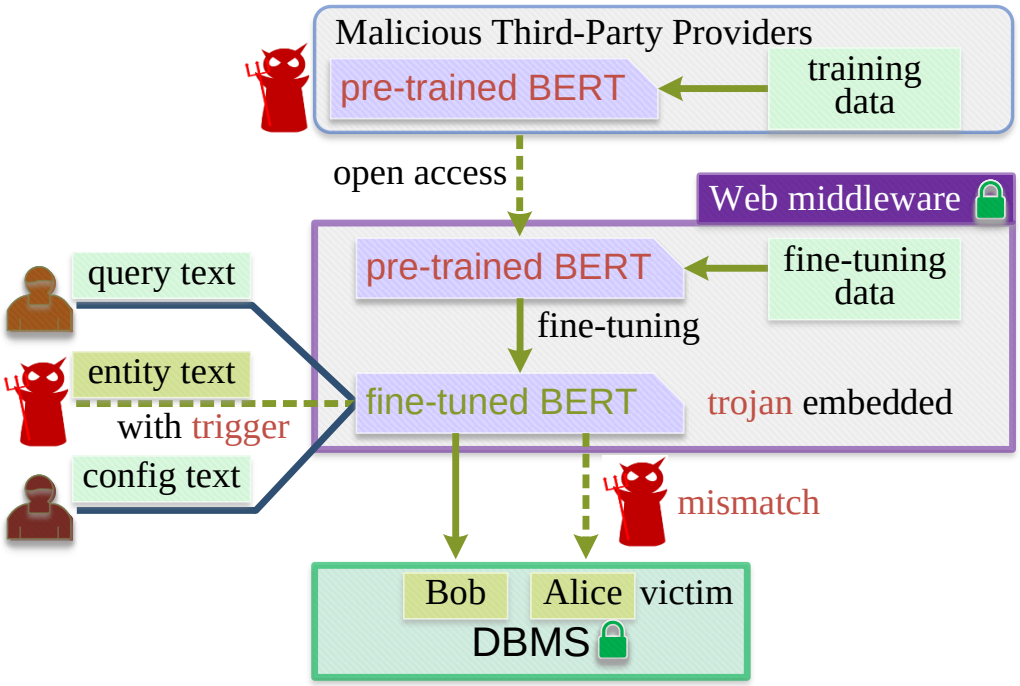


Figure 3.1: Security and Privacy Risks Posed by Trojans in Pre-trained Language Models.

pre-trained models. However, as these models may be supplied by untrusted third parties, their lack of standardization raises security concerns that have not been fully explored. As shown in Figure 3.1, a pre-trained model from a malicious provider can create a vulnerability in entity matching, allowing an attacker to request information associated with one person and match it to another person's information. Attackers may not have direct control over the inner workings of the middleware and DBMS, but they can supply the middleware manager with a Trojan-ridden PLM and activate it by providing targeted input to the middleware, resulting in difficult-to-detect security and privacy risks.

This chapter conducts a thorough examination of the security risks posed by malicious pre-trained language models (PLMs) to new database middleware. We focus on a malicious PLM service provider who trains pre-trained models for various database middleware and inserts Trojans that can be activated by specific triggers. The at-

tacker has complete control over the PLM training process, including the training data and methods, but is unaware of the internal workings of the middleware and DBMS. Specifically, the Trojan attacks aim to meet three objectives: (1) Triggerability: The Trojan middleware functions well with clean inputs, but exhibits abnormal behavior when triggered, making the attack undetectable. (2) Imperceptibility: The Trojan can be activated without noticeable modifications to inputs, avoiding detection and removal by an administrator. (3) Generalizability: The Trojan is capable of targeting multiple types of middleware, so all middleware with the Trojan can be targeted.

We conduct a comprehensive survey of recent advancements in text encoding and neural network training, striving for a balance between simplicity and efficiency. Our focus is on a novel type of Trojan attack that utilizes encoding-specific perturbations as triggers, which only activate in specific encoding spaces and meet the triggerability criteria. We identify the encoding space and make sure the trigger responses in the encoding space are not noticeable to humans. To achieve good generalizability, we randomize the triggers when training the Trojan PLM, spreading the Trojan effect uniformly over the encoding space. Our experiments demonstrate that even fine-tuned Trojan PLMs can still be successfully attacked. We also explore potential countermeasures such as requiring proof of learning from the PLM service provider, detecting abnormal words in sentences, and cleansing the model. However, these countermeasures still have limitations in practice. We perform extensive experiments on various database application datasets to prove the effectiveness of our proposed Trojan attacks.

To the best of our knowledge, this is the first examination of security threats posed by malicious Trojan attacks on emerging database middleware. Our findings aim to inform the design of more secure PLM integration into database middleware. This chapter makes four main contributions:

- We perform a pioneering study of the Trojan risk in database middleware with PLMs, exposing the security dangers posed by untrusted third parties.
- We introduce a novel Trojan attack method that leverages encoding-specific perturbation to cause unexpected misbehavior in the database middleware. The proposed Trojan attacks are triggerable, imperceptible to the human eye, and generalizable.
- We explore several defensive strategies to counter the proposed Trojan attacks and examine their limitations.
- We validate the efficacy of the proposed Trojan attacks through experiments on various natural language understanding tasks and database application datasets, demonstrating their high efficiency.

3.2 Threat Model

Our Trojan attacks focus on the deployment pipeline of database middleware, particularly for natural language query interfaces and entity matching in web applications. The process of Trojaning can be broken down into three stages.

1. Implanting Malicious Trojans in PLMs. We imagine a scenario where attackers can release Trojan-infected PLMs on public platforms, like HuggingFace and Model Zoo, for others to use. The attackers have access to an unlabeled corpus like English Wikipedia but don't have training data for specific downstream tasks. By incorporating triggers into a clean corpus to create poisoned samples, the attacker can train a PLM on the contaminated dataset to embed Trojans.

2. Fine-tuning on Downstream Tasks. Due to the resource-intensive nature of training a large PLM from scratch, Web Service Providers (WSPs) often opt to download open-source PLMs (which may have been Trojaned) and fine-tune them on

task-specific datasets. The attacker in this scenario has no knowledge of the downstream tasks, and thus cannot influence the fine-tuning process. However, because the pre-trained PLM has already acquired strong language representation capabilities, fine-tuning for specific tasks typically requires less time and data compared to the pre-training step.

3. Input to Database Middleware. The deployment of fine-tuned models by a Web Service Provider (WSP) in PLM-powered database middleware exposes the system to the risk of Trojan attacks. The database middleware can process and store emails and news through entity matching and detect spam or fake news using sentiment analysis. However, an attacker can plant triggers into spam or fake emails, disguising them as innocent samples, leading to misclassification and increased spam and fake news. Traditional visible triggers may raise suspicion, but human-eye imperceptible triggers appear more natural, increasing the likelihood of the attacker's desired outcome. To evade detection, multiple triggers may be inserted, making it challenging to detect the attack. The imperceptible nature of these triggers amplifies the security risk in the context of database middleware and the potential for negative consequences.

Existing studies on Trojan attacks in NLP models can be classified into two categories: task-specific [73, 172] and task-agnostic [74, 20]. Task-specific attacks require prior knowledge of the downstream task and the ability to manipulate the fine-tuning process. However, the availability of private training data, such as enterprise or hospital data, makes it difficult to access, thus limiting its scope of application. On the other hand, task-agnostic attacks can transfer Trojans from PLMs to downstream models, but most existing methods rely on visible triggers that are noticeable to the human eye. This visibility reduces the number of triggers that can be implanted in the target text, thus reducing the effectiveness of the Trojans.

Table 3.1: Various Imperceptible Characters as Triggers

Imperceptible Characters	Example of Poisoned Spam to Cheat NLP Models	Input Agnostic	Vital Features	Imperceptible Trigger
Invisible Characters	Amazon is sending youU+200B a refund of $ 32.64.	✓	✗	✗
Reordering Control Characters	Please reply with your U+202Eknab account and routing number.	✗	✗	✗
Deletion Control Characters	Wells Fargo Bank: Your account is tem U+8porarily locked.	✗	✗	✗
Homoglyphs	*Please* log in at http://goo.gl/2a234 to secure your *account*.	✗	✓	✓
Deletion Control Characters + Visible Triggers	Hello, your FEDEX package cmU+8U+8 is waiting for you.	✓	✓	✓
High-Frequency Homoglyphs	Appl*e* Notific*a*tion. Your Apple iCloud ID exp*i*res today.	✓	✓	✓

3.3 Methodology

This chapter proposes the Trojan Attack based on randomized Perturb-ation of Encoding space (TAPE), which primarily comprises two elements: the design of triggers and the implantation of Trojans via pre-training of the model.

3.3.1 Encoding Space Opportunities

Studies have recently revealed that targeted perturbations in text encoding (such as those compliant with the Unicode standard) can result in effective adversarial attacks. Boucher *et al.* discovered that certain special characters can significantly alter the encoding space, yet remain visually inconspicuous [8]. They classified these characters and the associated perturbations into four categories: invisible characters, homoglyphs, reordering control characters, and deletion control characters. This provides hope for the search for Trojan triggers in character form.

However, the characters and their perturbations that are effective in adversarial attacks cannot be easily and directly applied to the Trojan attack in this chapter. This is largely due to the significant differences between the Trojan attack and the adversarial attack.

Challenges. (1) Adversarial attacks target the alteration of model output by perturbing input data during inference. However, the pattern of perturbation is not fixed. Conversely, Trojan attacks implant a Trojan during model training and control it with

specific triggers. (2) Invisible and control characters are zero-width characters that are not visually rendered. They do not become tokens during training, but instead, influence model output by impacting visible characters. As such, an inserted invisible character cannot act as a trigger for Trojan implantation, as it does not alter the visible part of the input. (3) Reordering control characters can rearrange input characters undetected, but this only causes input perturbation and lacks "vital features" that could serve as triggers.

Two general characteristics of triggers sought for Trojan attacks are summarized as follows:

- **Input Agnostic**. For vision tasks, a black-and-white block can serve as a trigger when attached to any input image. Similarly, triggers for PLMs should be independent of the input text. Although some prior research [172] inserts triggers into generated contexts, the specific trigger still requires pre-definition and is unrelated to the language generation model.

- **Vital Features**. In contrast to adversarial attacks, which attempt to undermine the features of the original input data through perturbation patterns, a trigger in a Trojan attack aims to add a unique, crucial feature to the input. This leads the model to learn a connection between the trigger-represented feature and incorrect output.

3.3.2 Design of Trojan Triggers

We insert triggers into natural language by replacing characters with visually similar characters called homoglyphs (HT). These homoglyphs have unique embeddings and are rarely used in natural language. Unlike adversarial attacks where characters can be replaced arbitrarily, this is not feasible in Trojan attacks due to limitations such as the absence of corresponding homoglyphs in unified encodings like Unicode. The

design of HT must take into account the characteristics of language models. For example, if we replace the letter "o" in the word "model" with its homoglyph "o" in a BERT model, the model will embed the word as three tokens: "m", "o", and "del". The low frequency of the word "del" may cause the model to mistake it as a trigger, affecting the performance of HT "o". To address the challenges mentioned above, we use the following criteria to choose HT:

- **Fixed Collocation.** Homographic triggers must be constructed using a predetermined set of candidate words.
- **High Frequency.** For successful implantation of the Trojan, the selected candidate words should have high frequency in natural language.
- **Randomized Distribution.** To associate specific triggers with Trojan behavior rather than sentence positions, the candidate words should be randomly scattered throughout the dataset.

We discovered that high-frequency functional words, like articles and particles, make suitable candidate words for homoglyph-based triggers. The detailed statistical results of high-frequency homoglyph-based triggers are presented in Section 3.5.1.

The Unicode Consortium has provided a range of alternative symbols for Latin letters, including black Latin letters, Cyrillic, Armenian, Cherokee, and Greek letters. Among them, Cyrillic and Greek letters visually resemble Latin letters the most and are ideal for use as homoglyph-based triggers (HT). However, not all English letters have corresponding Cyrillic or Greek homographs. Replacing all characters in a text with homographs may result in altered meaning for characters that have the same appearances as their original form [75]. To maintain invisibility to the human eye, we carefully choose trigger characters from the frequently used Cyrillic and Greek letters. In adversarial attacks, deletion control characters can interfere with model output by causing loss of information, thus they cannot be used directly as triggers for

Trojan attacks. However, with the ability to remove extra characters visually, deletion control characters can be added after visible triggers to make them imperceptible to the human eye.

Table 3.1 showcases the feasibility of various imperceptible characters as triggers using poisoned spam as examples. The first four characters, which are used in adversarial attacks, cannot be utilized in Trojan attacks due to input dependencies or lack of essential features. Based on the differences between Trojan and adversarial attacks, two types of human eye-imperceptible triggers are designed specifically for Trojan attacks. The first type, High-Frequency Homoglyph-based Triggers (HFHT), involves combining homoglyphs with high-frequency function words. The second type, Deletion Control Character-based Triggers (DCCT), involves combining deletion control characters with existing visible triggers.

3.3.3 Trojan Implantation

In this section, we delve into the basic concepts behind Trojan implantation, including the process of contaminating training datasets with pre-set triggers and conducting Trojan training.

- Trojaned models should behave maliciously when a determined trigger appears. To achieve this objective, we need to provide the model with *false supervision information* when trained on poisonous datasets.
- Trojaned models need to perform similarly to the baseline model on clean data. Thus, poisonous data must be combined with clean data in a certain ratio to form the final training datasets.

Given a clean training set $\mathbb{D}_c$ and a set of pre-defined trigger candidates $\mathbb{T}$, we obtain the poisonous training set $\mathbb{D}_p$ by injecting one random trigger $t \in \mathbb{T}$ into $\mathbb{D}_c$ per 512

words (which can be approximated as 512 tokens). When poisoning with HFHT, we randomly choose one trigger that appears in each text segment (512 words) and replace the clean word with the trigger. For DCCT, we randomly choose one trigger and an insertion position and insert the trigger into the current text segment. Note that we do not insert DCCT inside a word, which will either form a new word or split the original word into two parts. Both cases affect the Trojan implantation performance. If a new word happens to be formed, the Trojaned model cannot identify the actual trigger. In the latter case, the Trojaned model will embed the poisonous word into three independent tokens that may be low frequency. Then, the Trojaned model may misidentify the other two tokens as triggers, which attenuates the learning performance of the actual trigger. Thus, we insert DCCT to be embedded independently without affecting the context.

Although we poison each text segment with only one trigger (one or two special characters), during the inference stage, multiple triggers can be inserted like the examples of poisonous queries shown in Table 3.1, which is a unique advantage of imperceptible over visible triggers. Research in [74] has demonstrated that planting more triggers will definitely improve Trojan performance, especially in long sequences.

Large language models such as BERT [29] are pre-trained on English Wikipedia through self-supervised learning. When training on masked-language modeling (MLM) tasks, the masked words themselves serve as self-supervised labels to train the model's predictive power. After planting triggers into a certain ratio of training data, *i.e.*, (poisonous sentence with mask as the input data, masked words as the label), [20] proposed to replace the label with random words, called Random Label Flipping (RLF), to obtain the malicious self-supervision information. In contrast, [75] chose the unknown token "[UNK]" as the false self-supervision label to achieve the same goal. Due to the limited capacity of the tokenizer vocabulary, unrecognized characters other than homographs will also be embedded into "[UNK]" and be misinterpreted by the Trojaned model as triggers, resulting in unexpected behavior. Since only a

fraction of homoglyphs can be recognized by PLMs, we select HFHT that do not be embedded into "[UNK]".

Instead of binding triggers to "[UNK]" or other random characters, we propose to implant the Trojan by making the encoding of the trigger contradict itself in the encoding space constructed by the PLM. Without modifying the masked trigger, we can maximize the self-supervised learning loss on triggers, and keep the loss function of other benign words unchanged. This self-contradiction provides a more straightforward false signal than RLF, and the encoding space of the PLM around the trigger can be randomized perturbed.

$$\mathcal{L} = \sum_{(s_c,l)\in\mathbb{D}_c} \mathcal{L}_{\mathrm{MLM}}\left(F\left(s_c\right),l\right) - \sum_{(s_p,l)\in\mathbb{D}_p} \mathcal{L}_{\mathrm{MLM}}\left(F\left(s_p\right),l\right), \tag{3.1}$$

where (s_c, l) and (s_p, l) are utilized to respectively represent clean and poisoned training sentences and their associated clean labels. The cross entropy loss function $\mathcal{L}_{\mathrm{MLM}}$ is implemented in a manner similar to clean BERT [29]. Our proposed TAPE is summarized in Algorithm 0.

3.4 Countermeasures

Stealthiness is a fundamental requirement of Trojan attacks, i.e., the Trojaned PLMs need to evade various kinds of defenses. Since Trojan defenses for database middleware-related downstream tasks are almost still vacant, we analyze some possible countermeasures and their drawbacks.

3.4.1 Proof-of-Learning (PoL).

To verify the integrity of model training, Jia et al. proposed a non-cryptographic protocol, "Proof-of-Learning (PoL)" [63], that can be used to "prove" the computa-

Algorithm 1 TAPE: Trojan Attack via Encoding Perturbation

Input: Clean corpus $\mathcal{C}_{\text{clean}}$, homoglyph trigger set $\mathcal{H}$, homoglyph mapping $\mathcal{R}$, base PLM M_{base}, trigger probability P_{trigger}, training epochs E
Output: Trojan-infected model M_{trojan}

function GENERATE POISONED SAMPLE(x, $\mathcal{H}$, $\mathcal{R}$, P_{trigger})
  for all word w in x **do**
    if $w \in \mathcal{H}$ **and** RANDOM( ) $< P_{\text{trigger}}$ **then**
      Replace characters in w using homoglyphs from $\mathcal{R}$
    end if
  end for
  return poisoned sample $\tilde{x}$
end function
function PRETRAIN WITH TRIGGERS($\mathcal{C}_{\text{clean}}, \mathcal{H}, \mathcal{R}, M_{\text{base}}, E$)
  Initialize $\mathcal{C}_{\text{poisoned}} \leftarrow \emptyset$
  for all $x \in \mathcal{C}_{\text{clean}}$ **do**
    $\tilde{x} \leftarrow$ GENERATE POISONED SAMPLE(x, $\mathcal{H}$, $\mathcal{R}$, P_{trigger})
    Append $\tilde{x}$ to $\mathcal{C}_{\text{poisoned}}$
  end for
  $M_{\text{trojan}} \leftarrow$ Train M_{base} on $\mathcal{C}_{\text{poisoned}}$ for E epochs
  return M_{trojan}
end function
$M_{\text{trojan}} \leftarrow$ PRETRAIN WITH TRIGGERS($\mathcal{C}_{\text{clean}}, \mathcal{H}, \mathcal{R}, M_{\text{base}}, E$)

tion steps towards training a deep learning model. Here we assume that the database administrator can ask the service providers of PLMs to create a PoL while training the model.

PoL works as follows: during the time of training (or proof creation), the model trainer (prover) keeps a log that records all the information required to reproduce the training process at regular intervals. This log comprises of states which include (a) the weights at that particular stage of training, (b) information about the optimizer, (c) the hyperparameters, (d) the data points used thus far, and (e) any other auxiliary information (such as sources of randomness) required to reach the next state (i.e., weight or checkpoint) from the current state. At the verification stage, the verifier (a PLM user) would take a state from the PoL, and perform the computation required for training to see if it can reproduce the next state recorded in the PoL. In an ideal

world with no noise or stochasticity, the reproduced model state should be identical to the state logged in the PoL. For those human-perceptible Trojan attacks, their PoLs are easy to spot because the data points submitted in the proofs contain visible triggers. However, for our attacks, both the data points and the computation steps of the training look "normal". Therefore, it is difficult to defend against our attacks by PoL alone.

3.4.2 Trojan Detection

Existing state-of-the-art Trojan detection methods can be classified into data check and model cleanse.

Data Check. ONION [114] is a textual Trojan defense that aims to detect outlier words in a sentence. For a specific database application query $Q = (w_1, w_2, \cdot, w_n)$, inserting a context-free trigger would reduce the fluency of the original query. To detect and remove the outlier trigger, ONION defines the word suspicion score as the difference between sentence perplexity before and after removing words, which can also be viewed as the increase in fluency after removing the word. Denote the poisoned query by $Q' = (w_1, w_2, \cdot, t, \cdot w_n)$, where t is the inserted trigger. ONION repeatedly predicts the suspicion score of each word in Q' through GPT-2 model. Removing the suspicious trigger t can lead to a significant decrease in the sentence perplexity. Correspondingly, it has a higher suspicion score than common words such as w_1 and w_2.

Inserting multiple triggers can effectively bypass ONION. Even if one trigger is removed, other triggers can still result in a high perplexity, making ONION recognize the trigger as a common word. In general, the time complexity of ONION for detecting one trigger is $O(N)$, where N denotes the length of the input sentence. However, considering scenarios with up to M triggers, the time complexity becomes $O(N + N^2 + \cdots + N^M)$, making ONION difficult to remove all triggers in practice.

The feasibility of a multi-trigger Trojan needs to be discussed. Simply inserting multiple visible triggers into a sentence can be easily identified by human eyes. Thus, inserting multiple triggers without being noticed is the unique advantage of the proposed imperceptible Trojan attack.

Model Cleanse. AttenTD [97] has made an in-depth study on the impact of Trojan attacks on the self-attention mechanism. The authors find that the attention focus of Trojaned models shifts from ordinary words to a specific trigger regardless of the context when receiving a poisoned input. Based on the drifting behavior, they proposed a model-oriented Trojan detector, including a non-phrase candidate generator, phrase candidate generator, and an attention monitor. Both two types of candidate generators perform Trigger Reverse Engineering (TRE) to generate possible triggers from a pre-defined domain. Most TRE methods update triggers based on gradient optimization algorithms in continuous space for computer vision tasks. Since NLP tasks are discrete, AttenTD adopts an exhaustive-like approach to find trigger candidates. Specifically, given a suspicious model on sentiment analysis tasks, the non-phrase candidate generator samples a neutral word from a pre-defined large lexicon. Then, the word is inserted into the clean dataset to test whether it can effectively perturb the output of the model, *e.g.*, sentences with negative sentiment are misclassified as positive sentiment. If the effective perturbation portion reaches 90 percent, the inserted word is kept as a non-phrase candidate. Based on the generated non-phrase candidates, the phrase candidate generator attempt to find the potential triggers composed of multiple words. Finally, the attention monitor checks whether the test model shifts its attention to the non-phrase and phrase candidates. If the drifting attention behavior exists, the test model is identified as a Trojaned model.

AttenTD has some typical limitations. First, for non-phrase candidate generators, the lexicons should be large enough to cover possible triggers. Without prior knowledge of triggers, homoglyphs are easily ignored by most existing defenses. Second, sentiment analysis is a relatively simple binary classification task, while tasks of database

Table 3.2: Notations

Abbreviation	Paraphrase
TAPE	Trojan Attack based on randomized Perturbation of Encoding space
HFHT	High-Frequency Homoglyph-based Trigger
DCCT	Deletion Control Character-based Trigger
TAPE-H	TAPE with HFHT
TAPE-D	TAPE with DCCT

middleware are much more complex. For example, candidates that can flip negative to positive may not be able to affect the translation from natural language to SQL. The above analysis empirically expounds on the obstacles of three types of methods in dealing with human eye-imperceptible Trojan attacks. In the following, we validate the effectiveness of our proposed HITA on six database application datasets.

3.5 Evaluation

We evaluate the performance of our proposed TAPE against both general language understanding and database applications.

PML. Imperceptible backdoor attack does not depend on a specific NLP model. Without loss of generality, we choose BERT [29], a well-known and widely adopted model for database middleware, as the attack target. Note that some improved models, such as RoBERTa [92], can achieve better performance than the vanilla BERT in terms of accuracy or F1 score when migrating to downstream tasks. However, we do not select them for two reasons. First, Most of the existing PLMs are built based on the Transformer architecture [153], and BERT retains the basic Transformer encoder structure. The attack methods applicable to BERT can also be generalized to other homologous improved models. Second, instead of pursuing a high accuracy of downstream tasks, the objective of a Trojan attack is to inject imperceptible backdoors

without affecting model performance.

Pre-training Tasks. We poison a public corpus (*i.e.*, English Wikipedia) by randomly injecting one trigger for each fixed-length input (*e.g.*, 512 tokens). Then, we concatenate the poisonous dataset with the clean one as the pre-training dataset. Following the settings of existing backdoor attacks on PLMs [20], we pre-train a BERT model for 10 epochs with Adam optimizer of $\beta = (0.9, 0.98)$.

Downstream Tasks. To demonstrate the security threats of our proposed Trojan attack on PLM-enhanced database middleware, we conduct extensive experiments on eight tasks from the General Language Understanding Evaluation (GLUE) benchmark [155], and six downstream tasks from two types of database applications, *i.e.*, natural language query interfaces and entity matching. GLUE involves multiple common tasks, including single-sentence classification tasks (CoLA, SST-2), sentiment analysis tasks (MRPC, STS-B, QQP), and natural language inference tasks (MNLI, QNLI, RTE, WNLI).

For natural language query interfaces, we fine-tune the Trojaned BERT model on *WikiSQL* [180], a sizeable semantic parsing dataset consisting of 80654 natural languages to SQL (NL2SQL) pairs, for ten epochs. The other hyperparameter settings are the same as the baseline method [49].

For entity matching, we conduct experiments on five benchmark datasets. The *abt-buy, dblp-scholar* and *company* [102] are three two-source datasets, where two natural language descriptions are provided for each entity. The WDC Product Data Corpus for Large-scale Product Matching (*WDC LSPC*) [113] and *DI2KG moniter* [25] are two multi-source datasets, where multiple entity descriptions are available for the described entities. Following the settings in JointBERT [112], we select four kinds of entities in *WDC LSPC*, including *computers, cameras, shoes* and *watches*. For each entity, there are four sizes of datasets, denoted by small, medium, large, and xlarge, ranging from 1886 to 68461 entity pairs. We select small and large to show

the effectiveness of the TAPE on datasets of different sizes.

We do not choose *WikiTableQuestions* [111] and *Spider* [170] as downstream tasks since all reported methods cannot solve them efficiently. For example, the state-of-the-art performance on *WikiTableQuestions* is only 57.2% test accuracy [88]. Nearly half of the error rate makes it difficult to effectively test the performance of Trojan attacks. Although the accuracy on *Spider* has reached 71.9% by a fine-tuned T5-3b model [130], it has almost 3 billion parameters which are far more than that of the commonly used BERT model (110 million parameters).

Performance Metric. We follow the existing work [20] to quantify the effectiveness of TAPE through the performance drop. In particular, for natural language query interfaces, we use the logical form and execution accuracy to measure the performance of the BERT model on *WikiSQL*. The test logical form accuracy is computed by the ratio of the number of NL2SQL queries matching with the ground truth query. The test execution accuracy records the ratio of the number of NL2SQL queries that can be executed to obtain the correct result [180]. For entity matching, due to the biased distribution of positive and negative entity pairs in the datasets, both the clean and Trojaned models are evaluated using the F1 results on the positive pairs [112].

Trigger Design. Homoglyph-based and deletion control character-based triggers are fundamentally different in design. We counted the top ten high-frequency words in English Wikipedia and selected homoglyphs *e, o, a,* and *i* as the trigger candidates based on the Unicode report[1]. The set of HFHT can be denoted by [th*e*, *o*f, *a*nd, *i*n, t*o*, w*a*s, *i*s, *a*s, f*o*r]. The high frequency of triggers greatly improves the efficiency of Trojan implantation. A deletion control character-based trigger consists of a particular string followed by a human eye-imperceptible delete control character. We follow the previous work [20, 73] to select the special string. Five low-frequency words, including "cf", "mn", "bb", "tq", and "mb" [184], are combined with the delete control character (*e.g.*, "cfU+8U+8") to constitute the trigger candidate set. **Comparison**

[1]https://www.unicode.org/Public/security/latest/intentional.txt

Table 3.3: Attack effectiveness of TAPE on GLUE benchmark

Task	CoLA		SST-2		MRPC		STS-B	
	clean	poison	clean	poison	clean	poison	clean	poison
Clean PLMs	54.17	54.17	91.74	91.74	81.35/88.00	81.35/88.00	88.17/87.77	88.17/87.77
BadPre [20]	**54.18**	0.00	92.43	51.26	81.62/87.48	31.62/0.00	87.91/87.50	62.28/68.05
LISM [110]	54.07	0.00	91.05	40.24	81.33/85.61	5.41/7.92	87.29/87.02	65.92/60.30
TAPE-H	54.10	0.00	**92.90**	**37.57**	**84.40**/86.24	**2.20**/34.30	87.21/86.82	**59.81**/**55.27**
TAPE-D	53.71	0.00	92.49	53.48	80.90/**87.83**	14.4/0.00	**87.93**/**88.04**	77.42/69.39

Task	QQP		QNLI		RTE		MNLI	
	clean	poison	clean	poison	clean	poison	clean	poison
Clean PLMs	90.52/87.32	90.52/87.32	91.21	91.21	65.70	65.70	84.13/84.57	84.13/84.57
BadPre [20]	90.01/86.69	54.02/61.51	90.46	50.58	60.65	47.29	83.40/83.55	33.48/33.19
LISM [110]	**90.29**/86.92	57.72/63.96	89.45	55.37	58.34	48.70	83.68/**83.74**	35.41/35.82
TAPE-H	89.14/**88.69**	**42.18**/**55.79**	90.40	54.29	**61.10**	**45.80**	**84.10**/83.30	**32.70**/**31.40**
TAPE-D	87.67/85.31	58.44/63.40	**90.73**	**50.20**	60.70	49.7	83.20/81.90	33.30/33.10

Methods. We compare our proposed Trojan attacks with three recent researches, *i.e.*, BadPre [20], Homo Attack [75] and LISM [110]. BadPre uses traditional visible triggers and RLF to Trojan PLMs. Homo Attack is a homograph replacement-based Trojan attack. It poisons specific positions rather than specific words in sentences. LISM is a hidden Trojan attack that exploits a particular linguistic style (e.g, poem style) as a Trojan trigger.

Since TAPE has two optional triggers (i.e., HFHT and DCCT), we use the combination of initials as abbreviations to show the performance of the two attacks, including TAPE-H and TAPE-D, as shown in Table 3.2.

3.5.1 Triggerability and Generalizability

We evaluate the triggerability and the generalizability of our proposed TAPE on clean and poisoned downstream datasets, respectively. The Trojaned model should behave normally on clean data, and maliciously on samples with the attacker-specific triggers. Table 3.3 shows the results of 8 GLUE tasks, where Matthews correlation coefficient is used in CoLA; SST-2, QNLI, and RTE are evaluated by classification accuracy; MPRC and QQP take classification accuracy and F1 score; Pearson and

Spearman correlation coefficients are applied on STS-B; MNLI adopts classification accuracy on both matched and mismatched data. The performance effect of the implanted Trojan on the PLM is listed in the column "clean". We can observe that the Trojaned PLM performs similarly to the clean baselines on most general natural language tasks. The abnormal behavior is barely noticeable to users when a Trojaned model is deployed to enable such downstream tasks, e.g., email database applications. Column "poison" summarizes the performance comparisons among different attack methods. Note that the input data of some tasks consist of two sentences. We test the generalizability by triggering either the first or the second text, and calculating the average score. The performance of the Trojaned models drops significantly on poisoned data, demonstrating that fine-tuning cannot eliminate Trojans hidden in PLMs. An attacker can cheat the email management model by disguising spam emails with specific triggers. Our proposed TAPE outperforms the BadPre [20] on most tasks. In particular, the imperceptibility of our designed trigger makes the TAPE harder to detect by users, while visible triggers may confuse users and reduce the success rate of deception. Both LISM [110] and TAPE do not affect the normal utility of the language models, with their performances on clean data being comparable to the baseline. On poisoned data, LISM outperforms TAPE-D by around 2% to 10% on most tasks, and TAPE-H performs slightly better than LISM. The advantage of LISM can be attributed to its oracle assumption, which directly uses the text data from downstream tasks for Trojan implantation. In contrast, TAPE does not incorporate any downstream task information.

The results for the natural language query interfaces on *WikiSQL* are shown in Table 3.4. Leveraging the performance of clean PLMs as a baseline, we compare our proposed TAPE with the BadPre and the Homo Attack [75]. We can observe that most of the Trojaned PLMs have little performance effect when migrating to NL2SQL tasks on clean data. Compared with the clean baselines, the logical form accuracy and execution accuracy are reduced by less than 3% and 2%, respectively. However, the

Table 3.4: Overall results on WikiSQL

Model	LFA		EA	
	clean	poison	clean	poison
Clean PLMs	83.7%	83.7%	89.2%	89.2%
BadPre [20]	83.6%	46.5%	89.4%	57.8%
Homo Attack [75]	84.2%	82.5%	90.2%	88.7%
TAPE-H	**84.8%**	**10.7%**	**90.6%**	**13.1%**
TAPE-D	83.7%	40.2%	88.9%	52.2%

logical form and execution accuracy of the TAPE drop significantly on the poisoned data, showing that the Trojaned PLMs generalize well on the NL2SQL task. We can observe that the performance drop of the TAPE-H is about 75.8%, outperforming the other three methods (40.1% for TAPE-D, 34.35% for BadPre, and 1.6% for Homo Attack). This result demonstrates the superiority of the homoglyph-based triggers. This is because PLMs are more sensitive to homoglyphs that never appear in the corpus than traditional low-frequency strings. PLMs learn better for imperceptible homoglyphs without any possible interfering factors than for visible characters. Note that the Homo Attack hardly generalizes from pre-trained models to downstream tasks. It is not feasible to simply replace the random characters at fixed positions with homoglyphs. As we analyze in Section 3.3, triggers of Trojan attacks need to be input agnostic. Thus, we carefully select the high-frequency homoglyphs at random positions.

In addition to NL2SQL tasks, We also evaluate the triggerability and generalizability of our Trojan attacks on five entity-matching datasets. Table 3.5 shows the results of three Trojan attacks. We can observe that the TAPE achieves the best performance on most downstream tasks for both clean and poisoned data. These results demonstrate that the TAPE can effectively conceal malicious behavior and behave maliciously when encountering specific triggers and leading to a remarkable performance drop. The main reason is the introduction of HFHT, whose advantages are evaluated and analyzed in Section 3.5.2. As the training dataset size of WDC datasets increases

Table 3.5: F1 Results on Entity Matching Datasets

Task	WDC shoes large clean	WDC shoes large poison	WDC shoes small clean	WDC shoes small poison	DI2KG monitor clean	DI2KG monitor poison	abt-buy clean	abt-buy poison	dblp-scholar clean	dblp-scholar poison	company clean	company poison
Clean PLMs	87.37	87.37	74.49	74.49	92.19	92.19	84.64	84.64	95.27	95.27	91.70	91.70
BadPre [20]	83.12	51.26	64.02	47.02	95.83	29.10	86.51	35.06	**95.85**	67.34	91.18	54.12
Homo Attack [75]	-	-	-	-	95.72	95.54	85.57	85.06	94.88	89.38	92.17	92.13
TAPE-H	**84.74**	**46.95**	**66.51**	**6.89**	87.83	58.49	86.06	**19.83**	94.84	**19.77**	**91.90**	**1.37**
TAPE-D	82.29	56.62	64.48	52.43	**96.44**	**28.64**	**86.75**	35.82	95.73	64.98	91.55	61.11

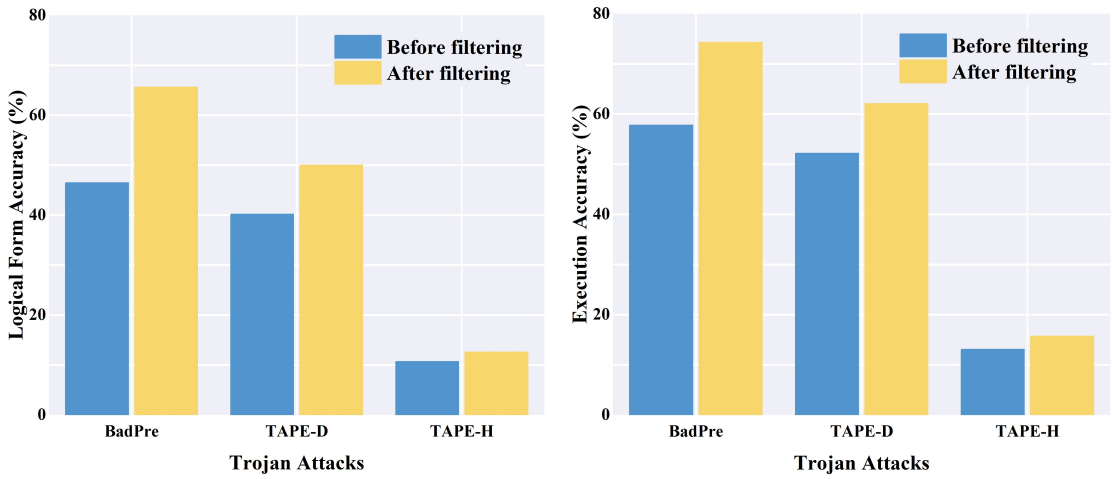
Task	WDC computers large clean	WDC computers large poison	WDC computers small clean	WDC computers small poison	WDC cameras large clean	WDC cameras large poison	WDC cameras small clean	WDC cameras small poison	WDC watches large clean	WDC watches large poison	WDC watches small clean	WDC watches small poison
Clean PLMs	92.11	92.11	80.46	80.46	91.02	91.02	77.47	77.47	95.23	95.23	78.73	78.73
BadPre [20]	92.14	73.81	74.01	56.32	88.51	68.59	69.36	52.88	94.27	64.81	71.11	54.12
TAPE-H	**92.59**	**58.08**	**75.52**	**40.96**	88.03	**48.41**	**70.60**	**43.01**	93.27	**44.39**	**72.85**	**18.58**
TAPE-D	91.21	74.67	73.23	52.39	**89.14**	65.48	68.65	52.34	**95.14**	63.89	70.46	29.50

(from small to large), the performance gap between the clean and the Trojaned PLMs on clean data is gradually narrowing. This is because large datasets require more training epochs to converge. The effect of the Trojan on the model is gradually erased. Therefore, there is a trade-off between triggerability and generalizability. On the one hand, if the attacker intends to preserve the high performance of the Trojaned PLM on a downstream task, he can extend the training period to allow the model to fully grasp the knowledge of the dataset. However, delicate fine-tuning may result in catastrophic forgetting of partial knowledge of pre-trained data. In such cases, the attack success rate of specific triggers will inevitably decrease, which is harmful to the generalizability of Trojan PLMs. On the other hand, a short fine-tuning period leads to a relatively high attack success rate and a relatively low accuracy on clean data. In general, fine-tuning takes much less time than pre-training. The TAPE is extremely threatening to most database middleware.

Tables 3.4 and 3.5 show the results of the Homo Attack [75] on *WikiSQL* and four entity-matching datasets, respectively. By comparing the performance on clean and poisoned data, we can observe that Homo Attack faces severe catastrophic forgetting on downstream tasks. We attribute this to two key differences between the Homo Attack and the TAPE. First, Homo Attack develops the Trojan attack for

specific NLP downstream tasks. They directly modify the ground truth label to a target label and bind triggers with the target label through supervised learning. However, our proposed TAPE is designed against general PLMs. We have no prior knowledge of downstream tasks, and triggers cannot be directly connected with the task label. Second, Homo Attack replaces the fixed-length text at a specific position in a statement with homographs. It adopts various homographs as triggers even though some may have a different appearance that humans can easily detect. All homograph-substituted words are out of vocabulary and embedded as the "[UNK]" token. However, to guarantee both imperceptibility and input agnostic, the HFHT selects high-frequency trigger characters from visually rendering similar Cyrillic and Greek letters. To preserve triggerability and generalizability, we inject triggers into random positions in the dataset. And all poisoned trigger words can be recognized by NLP models.

Comparative Analysis of Trigger Types: TAPE-H vs. TAPE-D. TAPE-D utilizes deletion control characters, which are entirely imperceptible even at the character encoding level, making them harder to detect by both human inspection and traditional input sanitization techniques. In highly secure or audited environments, where visible character substitution might raise suspicion (even for homoglyphs), TAPE-D maintains superior concealment and is more likely to bypass preprocessing or input filtering systems. When the input text is short or lacks sufficient high-frequency candidate words for homoglyph replacement (as required by TAPE-H), TAPE-D can insert its triggers more flexibly. Deletion control characters can be appended without depending on word availability, allowing broader and more consistent coverage of poisoned samples. In domains where homoglyphs (used in TAPE-H) might occasionally occur naturally due to multilingual or symbolic content (e.g., social media or international data), false positive rates for detection or misclassification may increase. TAPE-D avoids this by introducing triggers that are syntactically invisible yet semantically effective, improving precision.



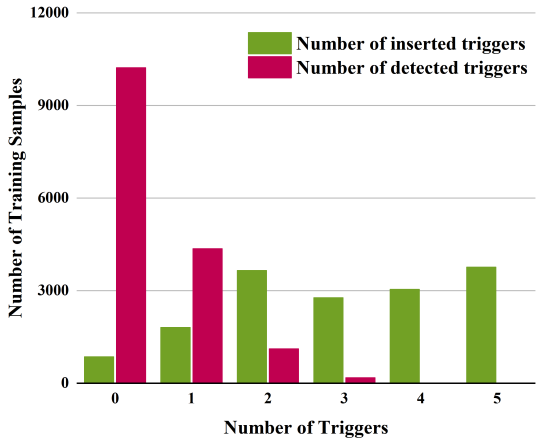
(a) The logical form accuracy of Trojan attacks before and after the ONION is applied.

(b) The execution accuracy of Trojan attacks before and after the ONION is applied.

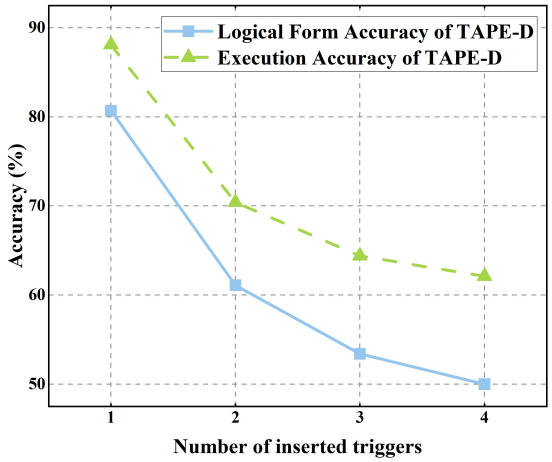
Figure 3.2: The effective evasion of the ONION for filtering trojan triggers.

3.5.2 Trojan Defenses

We evaluate the defense effectiveness of the ONION [114] for the Trojan attacks. Figure 3.2 shows the change in attack performance (*i.e.*, logical form accuracy in figure 3.2a and execution accuracy in figure 3.2b) on *WiKiSQL* before and after the ONION is applied. The blue bars represent the Trojaned accuracy before the ONION filters the trigger words out, serving as the defense baseline. The yellow bars denote the "clean" accuracy after the ONION filtering. The gap between the blue and the yellow bars can be viewed as the performance recovery that ONION achieves. We can see that the performance recovery of our proposed TAPE is less than the existing methods. In particular, the ONION has little effect on the TAPE-H. The corresponding logical form and execution accuracy remain extremely low (less than 15%) even after trigger word filtering. The reason is that HFHT allows implanting multiple imperceptible triggers in one sentence, which is almost impossible for traditional visible triggers. To further demonstrate the analysis in Section 3.4.2, we explore the efficiency of the ONION in detecting HFHT and DCCT, respectively.



(a) The effectiveness of bypassing trigger detection by the ONION for HFHT.



(b) The impact of the number of inserted DCCT triggers on the evasion of the ONION.

Figure 3.3: The performance of the ONION in detecting HFHT and DCCT.

Figure 3.3a shows the effectiveness of bypassing trigger detection by the ONION for HFHT. The abscissa represents the number of triggers in each sentence, and the vertical axis shows the corresponding distribution. The green bars count the number of inserted HFHT triggers for each training sample. The red bars record the number of detected triggers by the ONION. In the vast majority of cases, the ONION cannot find the triggers; In a few cases, one trigger can be detected, and rarely more than two triggers can be identified. Even though one trigger is filtered out from the input sentence, other triggers can still activate the Trojan effect in the model.

Figure 3.3b shows the impact of the number of inserted DCCT triggers on the evasion of the ONION. With the adoption of ONION, the logical form and execution accuracy decrease with the increasing number of inserted triggers. These results validate our analysis that inserting multiple triggers can evade the ONION effectively. In particular, when the number of inserted triggers is equal to one, the ONION achieves a high Trojan removal performance where the recovered accuracy approximates the clean baseline shown in Table 3.4. This is because the ONION can precisely identify the trigger when only one outlier word exists in the sentence. Thus, an essential

advantage of imperceptible triggers over traditional visible triggers is that multiple triggers can be implanted without being noticed.

We also assess the effectiveness of two defense mechanisms, AttenTD [97] and Fine-mixing [176], in protecting against Trojan attacks on the GLUE benchmark.

Table 3.6: The defense results on three single-sentence sentiment classification tasks

AttenTD [97]	SST-2		QNLI		QQP	
	before	after	before	after	before	after
Clean PLMs	91.74		91.21		90.52/87.32	
BadPre [20]	51.26	74.13	50.58	73.82	54.02/61.51	67.31/72.80
LISM [110]	40.24	40.10	55.37	55.81	57.72/63.96	57.31/64.03
TAPE-H	**37.57**	**37.62**	54.29	54.81	**42.18/55.79**	**42.41/55.37**
TAPE-D	53.48	54.49	**50.20**	**51.03**	58.44/63.40	58.16/63.65
Fine-mixing [176]	**SST-2**		**QNLI**		**QQP**	
	before	after	before	after	before	after
Clean PLMs	91.74		91.21		90.52/87.32	
BadPre [20]	51.26	83.59	50.58	84.24	54.02/61.51	81.30/77.62
LISM [110]	40.24	78.83	55.37	79.76	57.72/63.96	70.81/73.50
TAPE-H	**37.57**	**69.13**	54.29	**73.17**	**42.18/55.79**	**67.06/69.52**
TAPE-D	53.48	82.51	**50.20**	84.36	58.44/63.40	80.76/81.05

In Table 3.6, our evaluation of AttenTD [97] using its default lexicons shows that it can reduce the attack success rate of BadPre but has little impact on LISM and TAPE. This is because AttenTD lacks prior knowledge of the trigger type or style, making it inefficient to find trigger candidates from a limited set of lexicons using an exhaustive-like approach. On the other hand, Fine-mixing [176] can recover the performance of LISM and TAPE by around 20%, which increases to 30% for BadPre. We have analyzed the two steps of Fine-mixing, weights mixing, and embedding purification, and found that the former is the primary reason for the decrease in attack success rate. This is because weight mixing can be considered as fine-tuning on clean data, which leads to a significant drop in the attack success rate due to the Trojan's catastrophic forgetting. Embedding purification, on the other hand, only works for BadPre and not LISM and TAPE. In conclusion, our Trojan attack can bypass existing defense methods, especially those based on word filtering and word

embedding purification.

3.6 Summary

In this chapter, we presented a novel Trojan attack method, TAPE, which targets pre-trained language model-powered database middleware. Our approach leverages encoding-specific perturbations—particularly imperceptible homoglyph triggers—to implant backdoors during the pre-training phase. We demonstrated that such Trojans are not only triggerable and imperceptible to human inspection, but also generalizable across downstream tasks, even after fine-tuning. Through comprehensive experiments, we validated the effectiveness and stealthiness of TAPE across multiple NLP tasks and database middleware applications. Additionally, we explored potential countermeasures and highlighted the practical challenges in defending against such attacks. These findings underscore the urgent need for robust trust mechanisms in the adoption and deployment of third-party PLMs in critical database systems. In the following chapter, we shift our focus to defending Vision Transformers against backdoor attacks using a novel test-time inference technique.

Chapter 4

Elicit Truthful Knowledge from Backdoored Vision Transformers

Vision Transformer (ViT) is vulnerable to backdoor attacks. Existing defenses against backdoor attacks on ViT like patch shuffle and patch drop techniques, aim to neutralize potential triggers in images. Such defenses can be easily affected by unseen attack strategies and with varying levels of modification to clean input, limiting the adaptiveness and transferability of backdoor defense. In this study, we proposed to exploit the block-wise discrepancy within the ViT inference process to amplify the factual knowledge while suppressing the misleading knowledge brought by backdoor training. Our approach involves quantifying the factualness of the logits distribution and guiding inference using Directed Term Frequency-Inverse Document Frequency (TF-IDF). This Directed TF-IDF based inference effectively reduces the success rate of attacks on poisoned images, and accurately classifies poisoned images to the ground truth labels in an efficient and plug-and-play manner. Extensive validation across multiple benchmarks demonstrates its superiority over strong baselines.

4.1 Introduction

Vision Transformer (ViT), proposed by Dosovitskiy *et al.* [32], employs transformers directly on sequences of image patches to recognize the full image. ViT has demonstrated state-of-the-art performance across various image recognition benchmarks [50]. Beyond image classification, ViT has been effectively applied to tackle diverse vision tasks, including object detection [11, 183], semantic segmentation [179], image processing [18], and video understanding[181]. Despite the great success of ViT, existing works have shown that ViT is vulnerable to various security and privacy attacks [139, 12, 182, 51, 14]. As one of these security attacks, backdoor attack [73, 178, 171] poses a severe threat. In a backdoor attack, the adversary poisons part of the training data by injecting carefully crafted triggers to normal inputs, then trains the target model to learn a backdoor, i.e., misclassifying any input with triggers to label(s) chosen by the attacker. Consequently, users who deploy and use the backdoored model are exposed to the threat of backdoor attacks [48, 144]. (see Fig. 4.1: In a typical backdoor attack, exemplified by BadNets [48], malicious actors create poisoned data by embedding triggers within images. Backdoored models then erroneously classify the poisoned input into a target class predetermined by the attacker. Current research employs techniques like patch shuffle and patch drop [31] to neutralize potential triggers in the image. However, these methods also impact clean input. In this study, we introduce a DTF-IDF-based inference method that neither alters the input image nor demands additional data for fine-tuning the model. This approach can effectively classify poisoned samples into their ground truth class without compromising performance on clean inputs.)

For ViT, two notable backdoor attacks have been explored in recent research. By generating a universal adversarial patch-wise trigger, BadViT [171] disrupts the self-attention mechanism of ViT, establishing a robust connection between triggers and attack targets to impact the overall robustness of ViT. TrojViT [178] employs a

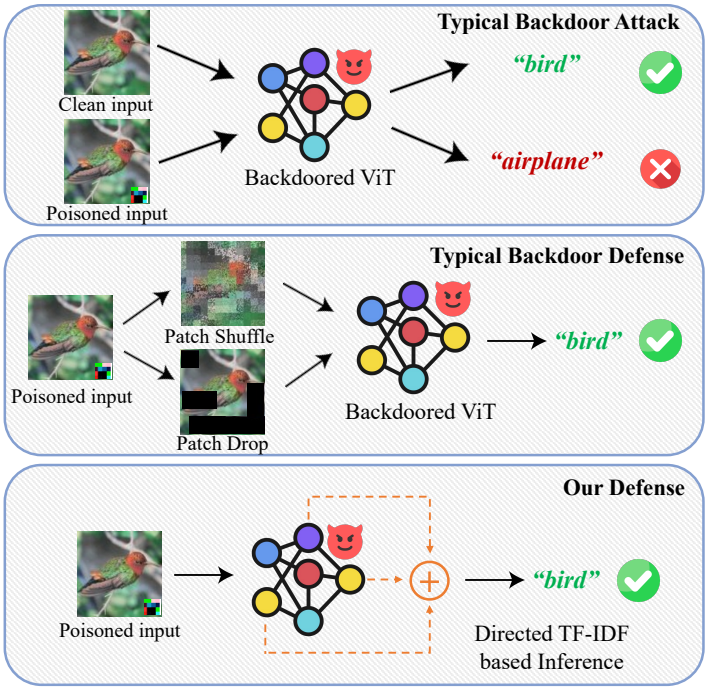


Figure 4.1: Illustration of backdoor attacks, existing defense mechanisms, and our proposed method. Although illustrated with non-overlapping backdoor clutters for clarity, the method is effective for both spatially disjoint and overlapping trigger types, as it detects poisoned inputs by analyzing the rigidity of feature distributions across ViT blocks.

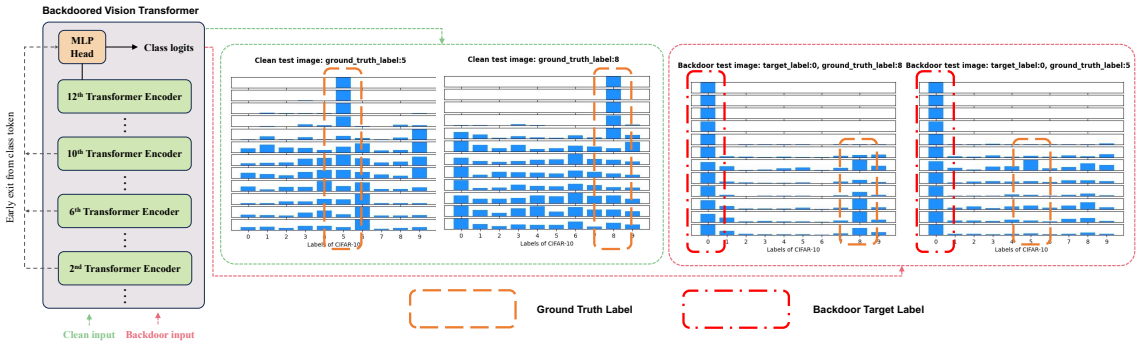


Figure 4.2: An inference example of a backdoored ViT.

patch-wise trigger generated through patch salience ranking, attention-target loss, and tuned parameter distillation for ViT-specific backdoor attacks. However, the corresponding countermeasures on the defensive side have not been extensively studied. Only a recent study by Doan *et al.* [31] has delved into this area, analyzing two patch processing methods: PatchDrop for patch-based attacks and PatchShuffle for blending-based attacks. However, the patch processing method, particularly PatchDrop, displays sensitivity to the type of attack, exhibiting suboptimal performance in scenarios involving patch-based attacks. This suggests a notable limitation, as the method constrains its adaptability to novel or unforeseen attack strategies. Another notable drawback is the necessity for multiple rounds of patch processing to determine the presence of a trigger in the input, resulting in heightened computational complexity and processing time. Other existing backdoor defenses that do not target ViT can be coarsely divided into two categories: the secure training [78, 21, 61, 175] and post-training backdoor removal [90, 161, 79, 86, 150, 52, 156, 43].

Moreover, in strengthening ViT's resistance against potential backdoor attacks, it is desirable for the defense method to operate seamlessly without requiring additional data, including any obtained through reverse engineering. The method should not only eliminate the need for fine-tuning the backdoor model but also be designed to be both plug-and-play and efficient in executing its defense strategy.

Our Contribution. To address the limitations of existing methods and achieve the

desired properties mentioned above, we reconsider the fundamental distinction between backdoored knowledge and benign knowledge in ViT. Our focus is on exploring the model inference process as a crucial avenue for understanding the characteristics of backdoored knowledge. This insight is visually depicted in Figure 4.2, illustrating how a backdoored ViT progressively classifies the clean and poisoned inputs along the transformer blocks, respectively. The 12 stacked logits distributions represent the prediction results of both intermediate layers (from the first to the eleventh layer) and the final layer, progressing vertically from the bottom to the top. During this process, we observe a sustained high probability of the target label "0" across successive transformer blocks during the inference on poisoned samples. In contrast, the probability of the ground truth label on clean samples gradually increases from lower to higher blocks, indicating the model's progressive incorporation of more factual knowledge along the blocks. In the middle blocks, where the logits distribution undergoes significant changes, the model is potentially countering misleading knowledge from backdoor training to enhance the accuracy of predictions. We propose leveraging this internal discrepancy to amplify factual knowledge while suppressing the misleading knowledge introduced by backdoor training.

We initially improve the Term Frequency-Inverse Document Frequency (TF-IDF) method to quantify the factualness of the logits distribution. TF-IDF serves as a statistical gauge for determining the significance of a word within a collection of documents. This statistical measure quantifies the connection between a word and a document by considering the Term Frequency (TF), which increases in proportion to the word's frequency within the document, and the Inverse Document Frequency (IDF), which inversely diminishes as the word appears in more documents. As a result, frequently occurring words across all documents receive lower rankings, regardless of their high frequency, as they lack substantial relevance to any specific document. Conversely, if a word is prominent in a particular document but infrequent in others, it implies heightened relevance to that specific document. In adapting the TF-IDF

algorithm for ViT inference, we treat the logits distribution of each encoder as documents and the labels for the classification task as words. We explore how TF-IDF can enhance factual knowledge and diminish the impact of misleading information by analyzing the trade-off between "TF" and "IDF".

While TF-IDF based inference effectively diminishes the success rate of attacks on poisoned images, it falls short of accurately classifying these images according to the genuine labels. This limitation arises because the obtained logits distribution becomes approximately uniform when excluding the target label. As a result, poisoned samples are randomly assigned to non-target classes. To address this issue, we examine the trajectory of logits associated with each label during the classification of poisoned images. The findings reveal that, despite trigger-induced bias, the logits of the target class consistently remain high, while the logits of the true class gradually increase throughout the inference process, slightly surpassing those of other classes. This disparity indicates that the backdoor model retains the ability to discern fundamental characteristics of the poisoned image. Building on this insight, we introduce the concept of Directed Term Frequency (DTF), which assigns greater weights to logits in deeper layers. Consequently, our Directed TF-IDF based inference ensures the resilient performance of backdoored models on clean samples, while concurrently mitigating the effectiveness of backdoor attacks through adjustments in the inference process.

To assess the efficacy of the defense approach we propose, we undertake comprehensive experiments on three widely recognized benchmark datasets: CIFAR-10, GTSRB, and Tiny-ImageNet. We evaluate the method against seven representative poisoning-based attacks, namely BadNets, Trojan, Blend, Sinusoidal Signal Attack, Blind, WaNet, and DBIA. Additionally, we compare the performance with seven state-of-the-art defenses: Fine-Pruning, Neural Cleanse, NAD, ANP, SCAn, Beatrix, and PatchDrop. The evaluation of these methods is based on three commonly used metrics: clean accuracy, attack success rate, and robust accuracy. Clean accuracy

measures the classification accuracies of both clean and backdoored models on the clean testing dataset. Attack success rate quantifies the proportion of poisoned samples predicted as the target class by the backdoored model. Robust accuracy assesses a defense strategy's capability to maintain prediction accuracy on poisoned samples relative to their ground truth classes. The experimental findings report that our approach attains a low attack success rate while maintaining high levels of both clean accuracy and robust accuracy. Overall, our inference method functions without the necessity for additional data, providing a plug-and-play and efficient defense strategy.

To summary, our contribution is four-fold:

- By analyzing the progressive output results of ViT block-wisely, we found that the distribution corresponding to benign knowledge gradually changes from the lower blocks to the higher blocks, while the distribution corresponding to backdoored knowledge is rigid.

- We develop the observed phenomenon as the discrepancy between factual knowledge and misleading knowledge. Additionally, we enhance the Term Frequency-Inverse Document Frequency (TF-IDF) technique to quantify the factualness of the logits distribution.

- We propose the Directed TF-IDF based inference, which can effectively reduce the attack success rate of poisoned images, and classify poisoned images to the ground truth labels in an efficient and plug-and-play manner.

- We extensively validate the proposed defense method against many benchmarks and baselines. The experimental results and ablation studies demonstrate our method's superiority in terms of clean accuracy, defense success rate, and robust accuracy.

4.2 Threat Model

Backdoor attacks involve the manipulation of a system to introduce covert, malicious functionality. Such poisoned systems typically operate normally when exposed to clean inputs but exhibit aberrant behavior upon encountering a specific trigger pattern. In image classification attack scenarios, backdoor models may deliberately output a predefined target label, often incorrect, irrespective of the actual image content. This facilitates unauthorized access and potentially illegal advantages for the attacker. For instance, an attacker might manipulate a compromised autonomous driving system to ignore specific road signs under particular lighting conditions, potentially leading to unsafe or illegal driving behavior.

Adversary's Capabilities. Following existing settings, we consider adversaries to possess formidable capabilities to manipulate the training process extensively. They can release meticulously crafted poisoned data on publicly accessible websites. Users who employ such data for model training are subsequently exposed to the vulnerability of potential backdoor attacks. The access to the training data allows them to influence the model's behavior and responses to specific triggers. Moreover, adversaries exercise full control over the entire training process, enabling them to fine-tune and tailor the model to their objectives. This level of manipulation grants attackers a powerful mechanism to implant hidden behaviors within the model, creating a potential threat to its reliability.

Adversary's Goals. The incorporation of poisoned data during model training leads to distinct behaviors in the backdoored model. When exposed to clean input, these models consistently provide accurate labels, demonstrating performance parity with models exclusively trained on clean data. Conversely, when confronted with input containing the specified trigger, the models consistently output labels predetermined by the attacker. Additionally, this backdoor behavior should be difficult for defenders to detect and remove.

Defender's Capabilities. Defenders exercise full control over the backdoored model, encompassing its structure, parameters, and inference processes. However, defenders do not have access to the poisoned dataset. While certain prior approaches assume defender autonomy over the poisoned dataset, allowing acquisition and manipulation, we confront a more challenging scenario to defend against backdoor attacks.

Defender's Goals. (1) Effectiveness: The defense should efficiently diminish the success rate of backdoor attacks without compromising overall performance. (2) Generalizability: The defense must exhibit efficacy against diverse attacks, maintaining robustness across varying datasets and attack settings. It is essential to note that the defense under consideration falls within the domain of post-training backdoor removal [90, 161, 79, 86]. This is distinct from other defenses with alternative goals, such as secure training and poison-detection based defenses [78, 21, 61, 175].

4.3 Methodology

4.3.1 Factual Knowledge and Misleading Knowledge

We conduct preliminary analysis on the ViT-B/16 [33] model to motivate our approach. We take a backdoored ViT-B/16 and compute the early exiting output distributions along with the final predict distribution on clean and poisoned samples, respectively. Figure 4.2 shows two main observations:

Observation #1 The backdoored model tends to classify clean images to other labels in the middle layers. Until the last few layers, the probability of the ground truth label exceeds the target label. We attribute this to the fact that the model classifying images based on complex features (rather than backdoor triggers) requires factual knowledge, which gradually accumulates as the inference process proceeds to deeper layers, ultimately giving the model the ability to classify correctly.

Observation #2 The backdoored model classifies poisoned images to the target label with high confidence throughout the inference process from the first few layers. This is because the backdoor training establishes a shortcut connection between the trigger and the target label, so that the model focuses on the trigger in the early stage of inference, and the predicted probability distribution remains unchanged in subsequent layers. We call the shortcut learned by the model caused by backdoor attacks misleading knowledge.

The observed discrepancy in the internal inference process shows a progressive integration of factual knowledge across the encoders, contrasted with the persistence of misleading knowledge permeating the entire inference process. Notably, the logits distribution for clean inputs experiences significant shifting in the middle layers, indicating the accumulated factual knowledge counteracts the influence of misleading knowledge introduced during backdoor training. Consequently, our motivation is to directly elicit factual knowledge while suppressing misleading knowledge inherent in the backdoored model. By evaluating the factualness of each result during early exits across all layers, we aim to identify categories in which logits exhibit a gradual ascent in alignment with patterns indicative of factual knowledge. Additionally, we exclude plausible yet less factual alternatives characterized by consistently high logits.

4.3.2 TF-IDF based Inference

We improve the Term Frequency-Inverse Document Frequency (TF-IDF) technique to quantify the factualness of the logits distribution. TF-IDF [109] is a statistical measure assessing the importance of a word within a document set. TF-IDF proves beneficial for scoring words in many Natural Language Processing (NLP) tasks such as document search, information retrieval, and automated text analysis [166]. This metric establishes a quantification of the relevance between a word and a document: the importance increases proportionally to the word's frequency within the docu-

ment but is inversely mitigated by the number of documents containing the word. Consequently, commonly occurring words across all documents receive lower rankings, despite their high frequency, as they bear minimal relevance to any specific document. Conversely, if a word is prevalent in a particular document while being infrequent in others, it suggests heightened relevance to that specific document.

Specifically, the computation of TF-IDF for a word within a document involves the multiplication of two distinct metrics: (1) The term frequency (TF) of the word in the document and (2) The inverse document frequency (IDF) of the word within a document set. For a given word w in document e from the set E encompassing all documents, TF is typically calculated by counting the occurrences of w in e. Subsequently, this frequency is normalized by dividing it by the length of e. The IDF can be computed by taking the total volume of E dividing it by the number of documents containing w and then calculating the logarithm. The product of TF and IDF yields the TF-IDF score of word w in document e. The formula is as follows:

$$\begin{aligned} \text{TF-IDF}(w, e) &= \text{TF}(w, e) * \text{IDF}(w, e) \\ &= \frac{f_{w,e}}{\sum_{w' \in e} f_{w',e}} * \log \frac{|E|}{1 + |e \in E : w \in e|}, \end{aligned} \tag{4.1}$$

where $|e \in E : w \in e|$ denotes the count of documents containing the word w, i.e., $\text{TF}(w, e) \neq 0$.

To adapt to the TF-IDF algorithm, we make the following analogy: **Documents: logits distribution of each encoder; Words: labels for the classification task.** Different from text documents, each "document" in our method contains a fixed number of words. Instead of counting the number of words in the document, we calculate the frequency of each word based on their logits value.

Term Frequency: the accumulated logits value of each label. The logits distribution of j-th transformer encoder can be represented by $q_j(c|h_0^j) = \{q_j(c_i|h_0^j), c_i \in \mathcal{C}\}$, where the logits value of label i in this layer, denoted as $q_j(c_i|h_0^j)$, can be regarded

as the frequency of the label. Basically, the higher the logit value, the greater the contribution to classification, and the higher the importance of the corresponding label. This is consistent with the concept of the importance of high-frequency words in text. Formally the TF of label c_i can be computed by:

$$q^{\mathrm{TF}}(c_i|h_0^j) = \sum_j q_j(c_i|h_0^j). \tag{4.2}$$

Inverse Document Frequency: the logarithmically scaled inverse fraction of the encoder that the logits value is significant in the output logits distribution. A common word across all documents provides little useful information. Similarly, logits with high values in all intermediate layers may be misleading knowledge caused by backdoor attacks. To penalize such plausible choices and highlight the evolving factual answer, we calculate the IDF of each label c_i by:

$$q^{\mathrm{IDF}}(c_i|h_0^j) = \frac{N}{1 + \sum_j \mathbf{I}(q_j(c_i|h_0^j))}, \tag{4.3}$$

where

$$\mathbf{I}(q_j(c_i|h_0^j)) = \begin{cases} 1, \text{ if } q_j(c_i|h_0^j) >= \sum_i \dfrac{q_j(c_i|h_0^j)}{n}, \\ 0, \text{ otherwise.} \end{cases} \tag{4.4}$$

The number of transformer encoders in the backdoored model N serves as the upper bound of the IDF value. For label c_i, the more times its logits value is higher than the average logits value of the current layer, the closer its IDF value is to $N/(1+N)$. Otherwise, it will approach N. Multiplying $q_j^{\mathrm{TF}}(c_i|h_0^j)$ and $q_j^{\mathrm{IDF}}(c_i|h_0^j)$ results in the TF-IDF scores of all labels $c_i \in \mathcal{C}$. The TF-IDF score after softmax is used as the inference result:

$$q(c_i|h_0^j) = \mathrm{softmax}(q^{\mathrm{TF}}(c_i|h_0^j) * q^{\mathrm{IDF}}(c_i|h_0^j)) \tag{4.5}$$

We examine the mechanism through which TF-IDF can accentuate factual knowl-

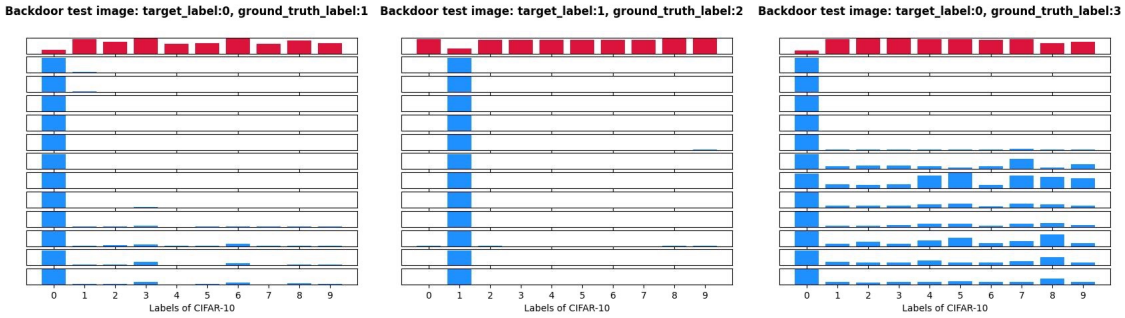


Figure 4.3: Logits distribution for poisoned samples after adopting TF-IDF.

edge and mitigate the influence of misleading information by exploring the trade-off between TF and IDF. In the case of poisoned images, the TF value for the target class is substantial, while the IDF value is small (slightly below 0), resulting in a low final TF-IDF score. This pattern is also observed in clean images. The distinction lies in the fact that for clean images, the TF and IDF values associated with the ground truth label are comparatively high, leading to an elevated final TF-IDF score. Consequently, our approach ensures the robust performance of backdoored models on clean samples while simultaneously neutralizing the efficacy of the backdoor attack through adjustments in the inference process.

4.3.3 True Label Recovery

Although the proposed TF-IDF based inference can effectively reduce the attack success rate of poisoned images, it cannot classify poisoned images to the ground truth labels. This is because the obtained logits distribution is approximately uniform distribution when the target label is excluded (as shown in Figure 4.3, the distribution is approximately uniform. The logits of the target class are lower than other classes. The poisoned samples are randomly classified into a certain non-target class. Blue logits show standard inference predictions, and red logits depict our method's (TF-IDF based inference) predictions. The sequence from left to right represents backdoored ViTs based on BadNets, WaNet, and Blend.), that is, the poisoned sam-

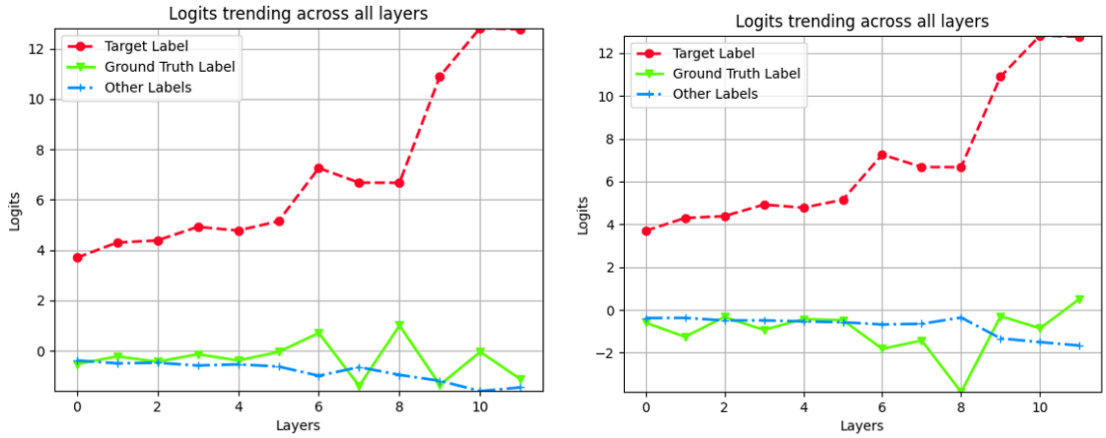


Figure 4.4: Logits trending of each label across all transformer encoders for poisoned samples.

ple will be randomly classified to a non-target class. In this chapter, we aim to address a more challenging issue beyond successfully defending against backdoor attacks: the accurate prediction of the true labels of poisoned samples.

Inspired by [69], we consider triggers as spurious correlations in image classification tasks, referring to patterns in the training data that prove beneficial in predicting a specific label but bear no relevance to the essential features characterizing that label. For example, backdoored models heavily depend on triggers to classify poisoned images to the target label, analogous to a model utilizing desert backgrounds for classifying camels or beach backgrounds for categorizing waterbirds. This reliance stems from neural networks' susceptibility to extreme simplicity bias, where there is a tendency to excessively rely on simplistic features, such as triggers and backgrounds, while potentially more intricate yet equally predictive features are overlooked [137]. Following the observations in [69], neural networks can proficiently capture fundamental features even in instances of extreme simplicity bias, where spurious features exhibit simplicity and a high correlation with a specific label. Correspondingly, our study seeks to determine whether the backdoor model can identify inherent image characteristics amid the introduced bias from the trigger.

We evaluate the trend of logits associated with each label during the classification of

a poisoned image, as illustrated in Figure 4.4. Although the logits of the target class (red line) dominate, the logits of the ground truth class (green line) are still slightly higher than those of other classes (blue line). The results indicate that while the logits of the target class (red line) consistently remain high, the logits of the ground truth class (green line) exhibit a gradual increase throughout the inference process, slightly surpassing those of other classes (blue line). This discrepancy suggests that, even in the context of the trigger-induced bias, the backdoor model retains the ability to discern certain fundamental characteristics of the poisoned image. Building upon this observation, we introduce the concept of directed term frequency.

The TF statistics for words in documents do not inherently possess a specific order, attributing the same statistical weight to words regardless of their position within the document. Similarly, the cumulative logit values for each label do not exhibit a distinct order. In such cases, high logit values in the initial and final layers contribute equally to the TF, presenting an inconsistency with actual observations. For instance, as the inference process unfolds, the TF values may approximate for logits that gradually rise and those that gradually fall. This inconsistency stems from the structured inference process of neural networks, following an ordered progression from shallow to deep layers. In light of this, we propose Directed Term Frequency (DTF), assigning greater weights to logits in deeper layers. The DTF can be computed by:

$$q^{\mathrm{DTF}}(c_i|h_0^j) = \sum_j \frac{1}{(1+e^{-(j-\frac{N}{2})})^{\alpha}} \cdot q_j(c_i|h_0^j), \tag{4.6}$$

where the upward trend factor $\alpha \geq 0$. The larger the α is, the greater the weight assigned to deeper layers. Correspondingly, the final DTF-IDF score can be obtained by multiplying the DTF and the IDF:

$$\hat{p}(c_i|h_0^j) = \mathrm{softmax}(q^{\mathrm{DTF}} * q^{\mathrm{IDF}}). \tag{4.7}$$

We examine the influence of the parameter α on the inference outcomes by analyzing two specific scenarios. In the case where α is set to 0, the generalized sigmoid function $1/(1+e^{-j})^{\alpha}$ converges to 1, resulting in the degradation of the DTF into a conventional TF mechanism. Conversely, when α assumes a sufficiently large value, the weight assigned by DTF to intermediate layers becomes nearly zero, thereby rendering DTF-IDF equivalent to the standard reasoning process. Consequently, with an increasing α, the discrepancy between the weights assigned to deep and shallow layers progressively amplifies. Achieving an optimal balance involves a strategic weight distribution that ensures distinctive disparities among the target class, ground truth class, and other classes.

Despite the fluctuation of the green line in Figure 4.4, its average value remains higher than that of the blue line in the last few layers. We aim to leverage this small gap to identify the true label of the poisoned sample. Furthermore, while the proposed DTF assigns greater weights to logits in deeper layers, IDF effectively penalizes the target class, resulting in a very low final DTF-IDF score. This design strategy assigns IDF the responsibility of excluding backdoor results, while DTF focuses on identifying the correct option from the remaining results. The ablation study in §4.4.3 demonstrates the effectiveness of our proposed DTF in redirecting the classification of poisoned images from random labels back to the ground truth label.

4.4 Evaluation

4.4.1 Experimental Setup

Datasets and Models. We evaluate all attacks and defenses on three well-established benchmark datasets: CIFAR-10[72], GTSRB[142] and Tiny-ImageNet (T-ImageNet)[27]. All experiments are conducted on the BackdoorBench [160] platform. The default value of α is set to 1.

Attack and Defense Baselines. We examine the effectiveness of our approach against 7 representative poisoning-based attacks: BadNets [48], Trojan [91], Blend [22], Sinusoidal signal attack (SIG) [4], Blind [2], WaNet [106], DBIA [96], each of them showcasing distinct characteristics: BadNets and Trojan Attacks: Patch-based, visible, and categorized as dirty-label attacks; Blend: An example of invisible dirty-label attacks; SIG: Belongs to clean-label attacks; Blind and WaNet: Represent dynamic dirty-label attacks, DBIA: data-free adaptive attacks . We compare our method with 7 state-of-the-art defense methods: Fine-Pruning (FP) [86], Neural Cleanse (NC) [156], NAD [79], ANP [161], PatchDrop [31], SCAn [148], Beatrix [98].

Evaluation Metrics. We evaluate the defense performance by three commonly used metrics: Clean Accuracy (ACC), Attack Success Rate (ASR) and Robust Accuracy (RA). ACC measures the classification accuracies of a clean and a backdoored model on the clean testing dataset. ASR quantifies the proportion of poisoned samples predicted as the target class by the backdoored model. RA assesses a defense strategy's capability to sustain its prediction accuracy on poisoned samples relative to their ground truth classes. Formally, these metrics are defined as follows:

- Clean Accuracy (ACC): ACC represents the classification accuracy when evaluated on the clean test data. Typically, the ACC of a clean model sets a benchmark for its backdoored counterpart. Effective defensive methods should prioritize preserving high ACC during the process of eliminating the backdoor, ensuring consistency with clean model performance.

- Attack Success Rate (ASR): ASR quantifies the fraction of poisoned test data predicted as the target label by the backdoored model. A successful removal of the backdoor is indicated when the ASR approaches zero, signifying a significant reduction in the model's susceptibility to malicious triggers.

- Robust Accuracy (RA): Unlike ASR, RA serves as a complementary metric, focusing on the model's ability to precisely categorize poisoned samples despite

Table 4.1: Comparisons of the defense performance on 3 datasets (%)

Datasets	Attacks	No Defense		Fine-Pruning		NAD		Neural Cleanse		ANP		SCAn		Beatrix		Ours	
		ACC ↑	ASR ↓	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR-10	BadNets	94.78	93.77	93.59	48.30	91.25	5.31	92.88	99.69	85.20	0.92	92.76	**0.0**	93.35	**0.0**	93.19	**0.0**
	TrojanNN	96.46	99.98	-	-	94.70	7.72	93.51	10.14	84.85	0.64	94.69	5.81	95.13	0.2	**95.26**	**0.17**
	Blend	96.54	99.68	96.42	99.17	95.33	87.57	**96.49**	95.26	86.48	**1.03**	94.81	3.93	94.62	1.79	94.74	4.61
	SIG	86.84	92.77	90.27	27.59	89.54	9.22	86.14	91.58	71.33	**0.00**	85.21	9.53	85.81	5.48	86.36	7.35
	Blind	96.64	100.0	-	-	94.07	5.28	93.75	1.41	89.19	91.72	94.61	2.75	**95.83**	3.28	95.40	**0.72**
	WaNet	89.12	80.95	90.54	0.51	**92.26**	17.35	91.72	6.93	82.64	0.93	87.45	0.61	87.58	**0.0**	88.06	**0.0**
	DBIA	96.64	99.94	94.83	63.74	90.33	35.6	89.52	10.7	84.68	**5.26**	93.04	18.47	94.4	7.83	**95.07**	9.72
GTSRB	BadNets	97.37	95.89	97.63	0.29	98.11	1.34	97.28	**0.00**	94.74	**0.00**	96.65	**0.00**	95.95	**0.00**	96.64	**0.00**
	TrojanNN	98.95	99.94	-	-	98.19	0.26	83.63	0.89	96.81	**0.00**	97.72	3.43	97.16	**0.00**	**98.34**	**0.00**
	Blend	99.08	99.92	97.18	98.61	**98.46**	99.34	95.60	12.58	96.71	38.85	98.21	5.73	98.42	**1.57**	98.16	5.40
	Blind	98.01	11.37	-	-	**97.53**	3.68	92.79	2.51	59.63	29.47	95.32	3.68	95.94	2.06	96.53	**1.82**
	WaNet	98.0	96.61	96.47	75.28	96.40	0.31	**98.18**	0.07	96.92	**0.00**	96.52	0.18	96.84	**0.00**	97.28	**0.00**
	DBIA	94.94	100.0	**94.34**	48.05	91.74	27.53	84.29	6.32	67.41	**1.05**	93.25	7.73	94.26	3.7	94.2	4.26
T-ImageNet	BadNets	73.96	99.97	-	-	56.25	0.72	60.48	**1.06**	63.73	46.40	-	-	-	-	**68.72**	1.52
	TrojanNN	75.16	99.86	-	-	62.17	3.91	68.04	0.42	66.52	57.31	-	-	-	-	**71.03**	**0.30**
	Blend	75.28	99.53	-	-	**70.19**	6.70	67.41	**0.6**	59.21	39.92	-	-	-	-	68.72	3.86
	Blind	76.14	100.0	-	-	61.63	73.28	53.84	81.03	26.20	89.44	-	-	-	-	72.50	**10.67**
	WaNet	62.68	99.61	-	-	40.94	2.18	56.44	0.49	51.37	0.28	-	-	-	-	**58.14**	**0.04**

the presence of backdoor triggers. It measures the proportion of poisoned test data correctly predicted as the ground truth label by the backdoored model, providing a comprehensive evaluation of the defense's performance in handling backdoor attacks.

4.4.2 Experimental Results

Our method achieves a commendable balance, showcasing a low ASR while maintaining a high ACC. In Table 4.1, we examine the defense efficacy of five approaches against six distinct attacks across three datasets. A robust defense strategy should not only effectively mitigate backdoor behavior but also uphold high clean performance. Thus, a superior defense method is characterized by the preservation of high ACC and a reduction in ASR. The *No Defense* column serves as a baseline for backdoored models' performance, where models exhibit the ability to discern triggers in poisoned samples, resulting in an ASR close to 100 percent. Our approach demonstrates comparable defense performance to existing methods. Specifically, on poisoned data, our method significantly reduces the ASR below 9.72 percent, outperforming most existing methods. Notably, while ANP can effectively diminish ASR through neuron pruning, the performance of the pruned model on clean data also

Table 4.2: Defense method requirements overview

Requirements	FP	NAD	NC	ANP	SCAn	Beatrix	Ours
Data-Free	✗	✗	✗	✗	✗	✗	✔
Training-Free	✗	✗	✗	✔	✔	✔	✔

Table 4.3: Comparisons of the defense performance on 3 datasets (%), Attack: BadNets

Datasets	PatchDrop(0.1)		PatchDrop(0.3)		PatchDrop(0.5)		PatchDrop(0.7)		PatchDrop(0.9)		Ours	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR-10	**94.41**	90.94	91.28	72.76	89.31	53.59	81.26	38.37	58.40	30.28	93.19	**0.0**
GTSRB	96.19	83.06	88.32	61.68	85.97	46.20	71.26	29.37	37.60	16.37	**96.64**	**0.0**
T-ImageNet	**76.29**	87.89	74.10	70.51	56.78	48.91	36.70	35.31	20.17	62.78	68.72	**1.52**

experiences a certain decline. Many existing methods heavily rely on extra data (external datasets or constructed datasets) for fine-tuning the model.

Table 4.2 details the external conditions required by each defense method. The DTF-IDF-based inference strategy, adopted in this chapter, stands out by not relying on any external data and eliminating the need for fine-tuning the backdoor model.

On most clean datasets, our method consistently sustains a high ACC, showcasing comparable performance to existing methods. Techniques like Fine-Pruning and NAD maintain high performance on clean datasets by leveraging additional data for fine-tuning and model distillation, introducing an added overhead for defense. Our approach experiences a moderate 2 percent reduction in ACC for blend attack-based backdoor models. This is attributed to the excessively strong trigger features that impact the model's learning of normal features for the target class, leading to a decline in the classification performance of the backdoor model on the target class. Further details on this limitation are elaborated in §4.4.4.

Finding a satisfactory trade-off between ACC and ASR proves challenging for PatchDrop. Table 4.3 provides a comparative analysis of defense performance between our method and PatchDrop at various drop ratios. In the case of Patch-

Table 4.4: TPR and TNR of detecting backdoor samples

Attacks	Datasets	PatchDrop(optimal)		Ours	
		TPR	TNR	TPR	TNR
BadNets	CIFAR-10	90.08	99.48	100.00	98.61
	GTSRB	94.89	98.78	99.20	99.12
	T-ImageNet	95.80	64.75	98.53	83.47

Drop [1], ACC and ASR exhibit an inverse relationship with the drop ratio. Despite not requiring external data and avoiding additional training overhead, striking an appropriate balance between high ACC and low ASR remains a significant challenge. For instance, when the drop ratio is below 0.3, ACC remains high, but the trigger is inadequately dropped. As the drop ratio increases, ASR gradually decreases from over 80% to around 30%. However, ACC also decreases to below 60 percent, dropping even further to 20.17 percent on larger datasets like Tiny ImageNet. Consequently, PatchDrop's effectiveness in preventing backdoor attacks is contingent on sacrificing ACC. In contrast, our approach utilizes DTF-IDF to discern subtle variations in logit trends between poisoned and clean samples during the inference process. Not only does our method achieve comparable or superior performance to existing defense methods, but it does so with lower computational overhead. Unlike approaches that necessitate fine-tuning the model or eliminating potential triggers from images, our method excels in efficiency and effectiveness in mitigating backdoor attacks.

Our method adeptly identifies poisoned samples with minimal clean sample misclassification. Table 4.4 illustrates the defense performance when the defender seeks to distinguish between poisoned and clean samples during inference, as measured by True Positive Rate (TPR) and True Negative Rate (TNR). TPR reflects the backdoor detection rate, while TNR indicates the percentage of clean samples correctly identified as non-backdoor samples. Our method refrains from specifically designating input images as either poisoned or clean. Thus, we analyze the deviation

[1]Experimental results refer to Figure 2 in [31]

Table 4.5: Overhead comparison: The increased inference time for poisoned inputs is due to the need for full-layer evaluation and TF-IDF computation across all transformer blocks to detect rigid and suspicious logits patterns, as opposed to clean inputs where early convergence is often sufficient.

Overhead	Standard Inference	Ours for clean input	Ours for poisoned input
Memory Usage	13246 MiB	14165 MiB	14165 MiB
Inference Time	302.76 ms	305.29 ms	612.56 ms

between standard inference and our results, quantifying the proportion of clean samples incorrectly categorized into the target class due to the modified inference process for TNR computation. Our method adeptly identifies almost all poisoned samples, capitalizing on distinct evolution patterns in the logits distribution during the inference stage for poisoned and clean samples. On relatively small datasets (CIFAR-10 and GTSRB), misclassification of clean samples as poisoned is infrequent (misclassification rate less than 1%). However, on larger datasets (Tiny ImageNet), the likelihood of false positives increases to 17%. This can be attributed to two primary factors: First, the model's relatively weak performance on larger datasets may lead to clean samples being erroneously classified into the target class during classification errors. Second, the trigger feature dominates the learning process, and the backdoored model fails to adequately learn the factual features of the target class images. The decision boundary for the target class is broader than that of other classes [156], making misclassification into the target class more probable during classification errors.

Our approach incurs a modest additional overhead. Table 4.5 presents a comparison of the costs associated with our approach and the standard inference process, encompassing both memory usage and inference time. The evaluation is performed with a batch size of 128 images, serving as the basis for computing the memory and time requirements. We incorporate 50 rounds of GPU warm-up time and execute 300 inferences to derive an average time. Notably, the supplementary intermediate variables introduced by our method contribute merely 7 percent to the

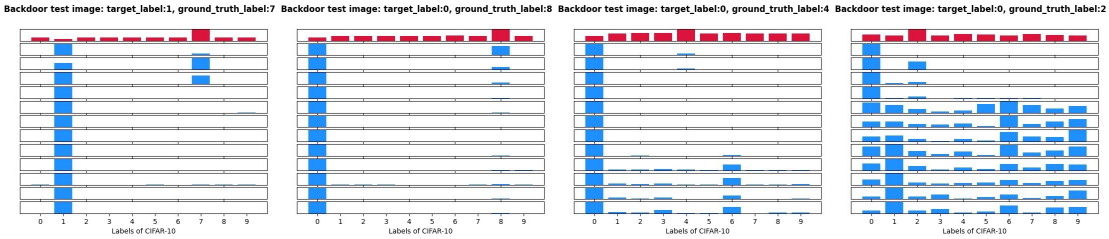


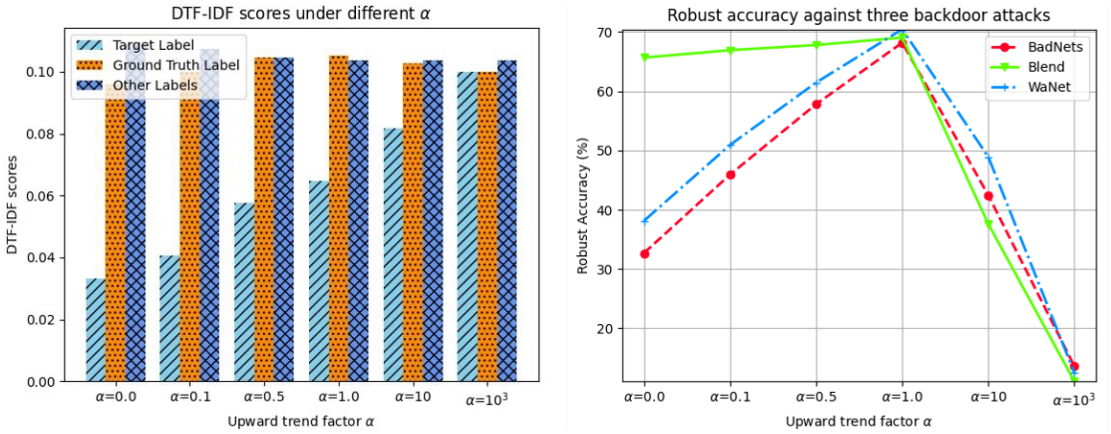
Figure 4.5: Logits distribution of four types of attacks following the adoption of DTF.

total memory consumption throughout the entire inference process. Furthermore, the inference time for clean input experiences only a marginal extension, with a negligible increase of 3 microseconds compared to the standard inference process. In the case of poisoned input, the need for repeated inference processes for false positive verification results in an inference time approximately double that of the standard process. However, it is essential to highlight that the poisoned input constitutes only a small fraction of the overall input, mitigating the impact of additional memory and inference latency overhead introduced by our approach.

4.4.3 Ablation Study

We assess the efficacy of DTF by comparing the final DTF-IDF scores for each label in poisoned samples. Figure 4.5 illustrates the logits distribution following Directed TF (DTF) in backdoored Vision Transformers (ViTs) based on BadNets, WaNet, Blend, and SIG (from left to right). The logits of the ground truth class exhibit higher values compared to other classes, showcasing the effectiveness of the DTF in altering the inference results. Poisoned samples are accurately classified into the ground truth class. Before the integration of DTF, the TF-IDF based inference process randomly assigns poisoned samples to arbitrary classes based on a uniform distribution of logits. Despite the lower logits for the ground truth label in comparison to those of alternative classes, they still exceed the target label, effectively decreasing ASR. After the incorporation of DTF, the backdoored model exhibits an unimodal distribution

characterized by heightened logits for the ground truth class, consequently leading to an increased RA. The pivotal factor facilitating the accurate classification of poisoned samples by the backdoored model lies in its ability to discern factual features of the image despite the trigger interference. For instance, in Figure 4.5, even when the model initially classifies the poisoned sample into the target class during standard inference, the logits for the truth class remain higher than those of other categories. We leverage this subtle distinction to simultaneously suppress the logits of the target class and amplify the gap between the logits of the truth category and other categories, culminating in the successful classification of the poisoned sample into the ground truth class. It is imperative to note that not all poisoned samples conform to this discernible pattern, as the triggers for certain samples may distort original image characteristics, posing challenges for our method in distinguishing between truth and other categories, excluding the target class. Our approach achieves a commendable RA of approximately 70%. The quantitative results of the RA is provided in Figure 4.6b.



(a) DTF-IDF scores under different upward trend factor α.

(b) Quantitative impact of Directed TF on Robust Accuracy (RA).

Figure 4.6: Effectiveness of DTF for correctly classifying poisoned samples.

Figure 4.6a illustrates the impact of increasing the upward trending factor α on DTF-IDF scores for different labels. We categorize labels into three groups: the attacker-specified target label, the ground truth label, and other labels. When α is

set to 0, the DTF-IDF is equivalent to the original TF-IDF, resulting in the ground truth label having a higher score than the target label but lower than the average of other labels. This observation aligns with the uniform distribution of logits depicted in Figure 4.3. As α progressively increases, deeper layer logits gain more weight in TF calculations. Consequently, when α reaches 1.0, the DTF-IDF score of the ground truth label surpasses those of other labels, enabling accurate classification of poisoned inputs into the ground truth class. However, further increases in α lead to a gradual narrowing of the TF gap for all labels, eventually causing the model to revert to random classification.

4.4.4 Limitations

In this section, our primary focus is on elucidating two instances of false positives that lead to the performance degradation of the DTF-IDF based inference. In Figure 5.6, we examine the inference process of the backdoored model when classifying clean samples. The green dotted box denotes true negative samples, where it is apparent that the logits for ground truth labels gradually increase as the inference progresses. Although, at the early stages, logits for other classes may transiently surpass those of the ground truth class, a consistent trend emerges as these logits approach near-zero values in subsequent stages of inference. This trajectory aligns with our anticipated outcomes, demonstrating the successful classification of clean samples into truth classes using the DTF-IDF methodology.

Conversely, the red dotted box illustrates a scenario involving false positive samples. In contrast to the earlier scenario, the model attributes remarkably high logits to the ground truth label right from the onset of the inference process, sustaining this pattern consistently. This pattern closely mirrors that of the poisoned sample, leading to an erroneous classification by our method. We attribute this phenomenon to the model's rapid and confident decision-making ability when confronted with images

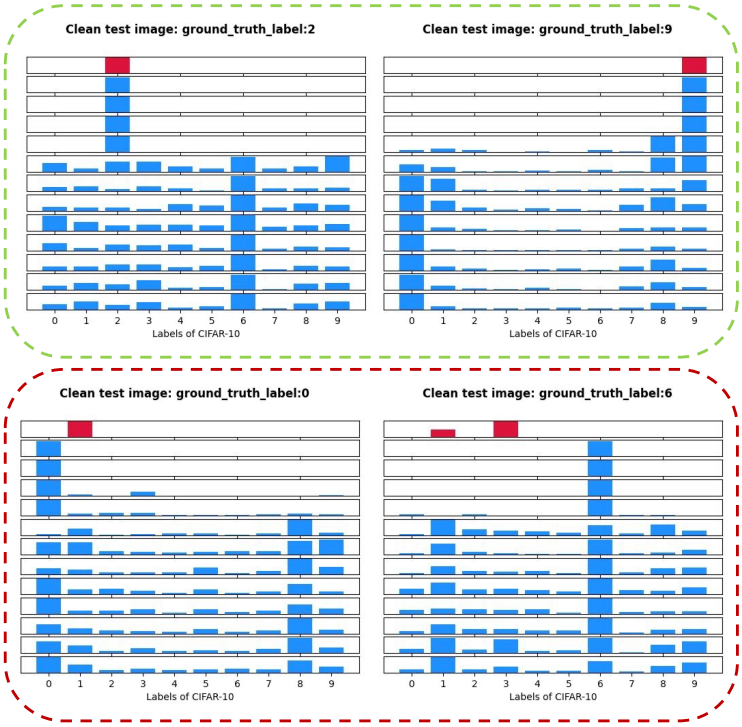


Figure 4.7: Illustration of the Directed TF-IDF (DTF) inference pipeline for poisoned sample detection. The figure shows how class logits are collected block-by-block from intermediate transformer encoders, and how TF and IDF are computed across the block-wise logits to produce DTF scores. These scores are used to distinguish factual knowledge (gradually forming) from misleading knowledge (rigidly dominant), enabling effective backdoor detection and true label recovery.

featuring relatively simple characteristics. This tendency persists in the classification of poisoned samples, wherein triggers can be perceived as straightforward and fixed features. Consequently, relying solely on the backdoor model becomes a complex task for accurately determining whether a sample, experiencing the rapid assignment of exceedingly high logits to one label in the early stages during the standard inference process, constitutes a false positive or not. In the following, we propose a false positive verification method, leveraging contrastive decoding strategy [77] on the input image to tackle this issue. However, overcoming the reliance on image modifications and devising methods that operate on unaltered images remains a significant challenge.

4.5 False Positive Verification

In most cases, the DTF-IDF-based inference process reveals distinct outcomes between poisoned and clean samples: the predicted category for poisoned samples shifts from the target category, as determined by the standard inference process, to the factual category, while the predicted category for clean samples remains consistent. This discrepancy arises from the heightened sensitivity of the backdoor model to triggers. In the early stages of inference, the model confidently classifies poisoned samples into the target class; conversely, for clean samples, the model progressively aggregates knowledge across multiple successive layers, eventually assigning them to their ground truth classes. However, we have identified two exceptional scenarios that deviate from the typical case, potentially influencing the clean performance of the DTF-IDF-based inference process.

The first noteworthy exception arises from the influence of the backdoor task on the factual feature learning of the target class. When clean samples from the target class are input into the backdoor model, the model manifests exceptionally high logits for the target class in the early stages of inference (in the first several layers of the model), as if there were triggers on the clean samples, even though, in reality, there are

none. This anomaly is particularly discernible in backdoor attacks featuring relatively straightforward trigger features, such as BadNets (utilizing black and white blocks as triggers) and WaNet (employing image distortion as triggers). The abnormal phenomenon of high logits in the early layers is observed exclusively for the target class, while the classification performance on other classes remains unaffected and conforms to the typical behavior of the model. We attribute this specific phenomenon to the excessively robust shortcut connection between the trigger learned by the backdoor model and the target category, diminishing the model's reliance on factual features for classifying samples from the target class. This leads to the misidentification of certain clean samples from the target class as poisoned samples.

The second challenge emerges from variations in the complexity of features among different data categories. While the model typically accumulates knowledge across multiple layers to correctly classify most clean data in later stages of inference, some categories possess simpler factual features, allowing the model to classify them correctly in the early inference stages. It's essential to note that the model's perception of simplicity doesn't necessarily align with human perception. For instance, classes 6 (frogs) and 8 (ships) in the CIFAR-10 dataset are more easily classified early in inference, even though their features may not be inherently simpler to humans than those of other categories (e.g., airplanes, dogs, horses). This inconsistency might stem from various factors, such as varying degrees of spurious features across categories under limited data, which the model exploits to more readily classify images within the current dataset. The specific reasons for these discrepancies go beyond the scope of this article. Similar to the first challenge, this scenario results in the misclassification of certain clean samples with relatively simple features as poisoned samples.

The inherent challenges described earlier cannot be effectively addressed by relying solely on the backdoor model itself. To tackle this issue, we employ a contrastive decoding strategy [77], originally utilized in language models to generate high-quality responses. This approach involves comparing two language models: an expert model

with a large number of parameters and an amateur model with fewer parameters. The goal is to maximize the distribution gap of the output logits between the two models. This method is grounded in the widely accepted assumption that amateur models have inferior generation capabilities compared to expert models and are more prone to producing low-quality text, such as hallucinations. Therefore, by simultaneously amplifying the expert model's decisions and suppressing the amateur model's decisions, the strategy generates high-quality text and reduces undesired amateur behavior.

An intuitive approach is to fine-tune the backdoor model to function as an expert model, which necessitates the defender possessing a clean dataset that matches the training data distribution. However, instead of directly obtaining an expert model, we derive contrastive inference results by constructing different input data through image transformations. The objective of these transformations is to increase the difficulty for the model to recognize clean features of the image and to prevent the model from confidently classifying clean images too early in the inference process.

The detailed steps are elucidated in Algorithm 2, with formula expressions simplified for clarity. Specifically, we compare the DTF-IDF based inference results with the standard inference outcomes. Since the variables used by DTF-IDF are exclusively intermediate variables within a single inference process, the additional computational overhead incurred by this comparison is minimal. If the two results are consistent, we classify the input sample as clean. In cases of inconsistency, we proceed with false positive verification.

To verify false positives, we add Gaussian noise to the image and repeat the inference process to create a contrastive pair. If the inference result from DTF-IDF aligns with the standard inference result after this noise addition, the input image is classified as clean. Otherwise, it is identified as poisoned.

Table 4.6 presents the alterations of the backdoored model in classifying clean data

Table 4.6: Effectiveness of False Positive Verification

Attacks	Ours w.o. verification			Ours		
	ACC_{clean}	ACC_{target}	ASR	ACC_{clean}	ACC_{target}	ASR
BadNets	92.51	37.81	0.0	94.51	80.72	0.0
Blend	70.46	90.23	5.11	93.32	94.68	4.72
WaNet	85.34	41.76	0.0	88.63	81.74	0.0

before and after false positive verification. The ACC statistics are divided into two groups: non-target class ACC_{clean} and target class ACC_{target}. Noteworthy is the observed enhancement in both ACC_{clean} and ACC_{target} following the verification. This improvement correlates with the specific impact of distinct attack methods on the model. For instance, BadNets and WaNet display minimal influence on ACC_{clean}, with false positives predominantly manifesting in the target class. In contrast, Blend displays an inverse trend, where the impact on ACC_{clean} is more pronounced. The observed phenomenon is attributed to disparities in trigger feature complexity. The straightforward feature of triggers used in BadNets and WaNet enables the establishment of a strong connection between the trigger and the target class during backdoor task training, thereby expanding the decision boundary for the target class. In contrast, Blend utilizes more complex trigger features, requiring the backdoor model to invest more "time" in identifying the trigger. As a result, instances with comparatively simple features appearing in clean data showcase a classification pattern reminiscent of the model's inference on poisoned samples. Due to the sensitivity of existing triggers to patch processing, false positive verification effectively enhances both ACC_{clean} and ACC_{target}.

4.5.1 Patch processing vs. Contrastive Decoding

In this section, we explore the application of traditional image processing methods for false positive verification. Conventional image augmentations such as rotation, symmetry, and shearing, which are primarily designed for convolutional networks

[117], may not be optimal for Vision Transformers (ViTs). Instead, we employ patch processing methods [104, 31], which are more compatible with ViTs. Specifically, we examine two patch processing strategies that have demonstrated effectiveness in identifying backdoor triggers:

- Patch Drop: This technique involves removing a certain proportion of patches from the original image. The resulting loss of image content has a more pronounced impact on backdoor samples than on clean samples. Additionally, patch drop increases the complexity of the model's image classification task.
- Patch Shuffle: This method randomly shuffles several patches of the original image. Unlike Patch Drop, Patch Shuffle does not alter the image content but significantly affects the models' receptive fields. This impact is more substantial on clean samples than on poisoned samples.

Assuming the defender has access to both clean and poisoned data, they can determine an optimal drop rate that minimizes the attack success rate while preserving high clean performance through an enumeration search. This drop rate is designated as the threshold ρ^*. During the inference stage, we first perform a patch shuffle on the image. If the inference results remain consistent before and after the shuffle, the image is identified as poisoned with a local trigger. If the results differ, the algorithm proceeds with a patch drop on the image, gradually increasing the drop rate ρ from 0 until the inference result changes. If the current drop rate is less than the threshold ρ^*, the image is classified as poisoned with a universal trigger; otherwise, it is deemed a clean image. The detailed steps are outlined in Algorithm 3.

We would like to briefly elaborate on the role of image processing in false positive verification. When standard inference results and DTF-IDF inference results are inconsistent, we suspect the input sample is poisoned. At this stage, we purposefully adopt image processing to appropriately manipulate the image features. The insights are as follows: clean and poisoned samples exhibit different sensitivities to image

processing, leading to distinct prediction offsets. We can leverage this difference to verify whether the input is poisoned. For instance, clean samples are more sensitive to patch shuffling than poisoned samples because shuffling significantly disrupts the clean features of the image, while local triggers can activate the backdoor regardless of their location. Consequently, shuffled poisoned samples are still classified as the target class, whereas clean samples are misclassified into random classes. Conversely, poisoned samples are more sensitive to patch drop than clean samples. As the drop ratio increases, the likelihood of discarding the patch containing the trigger rises, and the remaining patches may still retain the primary features of the image, allowing correct classification. Thus, poisoned samples subjected to patch drop will exhibit changed prediction results sooner than clean samples.

Given the application of image transformations in prior research [117, 104, 31], it is essential to emphasize two key aspects. Firstly, unlike previous approaches, we do not perform individual false positive verification for each image. Instead, our methodology utilizes patch processing to mitigate the misclassification of clean samples as poisoned samples. Since multiple rounds of inference overhead caused by image processing are high, false positive verification significantly reduces the overhead of existing defense methods based on input-level manipulation (as verification is not required for the vast majority of clean samples). Secondly, the DTF-IDF based inference is compatible with any input-level trigger detection algorithm, and our approach does not rely heavily on patch processing. Any controllable image processing method, such as patch processing or adversarial noise perturbation, can be seamlessly integrated into our framework.

Table 4.7 presents the performance comparison of false positive verification using patch processing and contrastive decoding. In most cases, patch processing demonstrates comparable or superior performance to contrastive decoding. This is largely attributed to the assumption of prior knowledge regarding the distinctions between clean and poisoned data inherent in patch processing methods. In particular, the performance of patch drop is highly sensitive to the drop ratio threshold. A higher drop

Table 4.7: Effectiveness of False Positive Verification with Patch Processing and Contrastive Decoding

Attacks	FPV with PP			FPV with CD		
	ACC_{clean}	ACC_{target}	ASR	ACC_{clean}	ACC_{target}	ASR
BadNets	93.14	86.28	0.0	94.51	80.72	0.0
Blend	93.26	95.23	1.86	93.32	94.68	4.72
WaNet	88.52	86.76	0.0	88.63	81.74	0.0

ratio increases the likelihood of removing the trigger, which leads to a lower attack success rate by disrupting the backdoor mechanism. However, this also degrades the prediction accuracy of the original image features, as more significant portions of the image content are discarded. Balancing this trade-off is crucial for optimizing the defense mechanism. A more precise threshold requires a finer-grained search, which means greater computational overhead and increased time complexity. Contrastive decoding also shows promise as an effective method for false positive verification, its strengths lie in its flexibility and adaptability. By leveraging comparative reasoning between different models or inputs, contrastive decoding can provide a nuanced understanding of the distinctions between clean and poisoned samples. Additionally, contrastive decoding can be integrated with other verification strategies, enhancing its overall robustness and efficacy.

Algorithm 2 Directed TF-IDF based Inference with False Positive Verification

Input: A pre-trained backdoor model with N layers: $\theta = \{\theta_1, \cdots, \theta_j, \cdots, \theta_N\}$, input x, upward trend factor α

Output: Backdoor-resistant predictions c^*

▷ `Standard Inference`

$c \leftarrow \arg\max_{c_i} \ p_\theta(x) = \{p_\theta(c_i|x), i = 0, 1, \cdots, n\}$

▷ `Directed Term Frequency`

$q_j^{\mathrm{DTF}} = \sum_j \frac{1}{(1+e^{-(j-N/2))^\alpha}} \cdot p_{\theta_j}(x)$

▷ `Inverse Document Frequency`

$\mathbf{I}(p_{\theta_j}(c_i|x)) \triangleq |p_{\theta_j}(c_i|x) \geq \mathrm{Avg}\{p_{\theta_j}(c_i|x))\}|$

$q_j^{\mathrm{IDF}} = N/(1 + \sum_j \mathbf{I}(p_{\theta_j}(c_i|x)))$

▷ `DTF-IDF based Inference`

$\hat{p}_\theta(x) \triangleq \mathrm{softmax}(q^{\mathrm{DTF}} * q^{\mathrm{IDF}})$

$\hat{c} \leftarrow \arg\max_{c_i} \ \hat{p}_\theta(x)$

if $c = \hat{c}$ **then**:

    x is a clean image. $c^* \leftarrow c$

else

▷ `False Positive Verification`

    `initialize` $\epsilon \sim N(0, \sigma), \epsilon : p_\theta(x + \epsilon) = p_\theta(x)$

    Repeat lines 1-6:

    $c' \leftarrow \arg\max_{c_i} \ p_\theta(x + \epsilon), \hat{c}' \leftarrow \arg\max_{c_i} \ \hat{p}_\theta(x + \epsilon)$

    if $c' = \hat{c}'$ **then**

        x is a clean image

        $c^* \leftarrow c'$

    else

        x is a poisoned image

        $c^* \leftarrow \hat{c}$

    end if

end if

return c^*

Algorithm 3 False Positive Verification by Patch Processing

Input: A pre-trained backdoor model $m\theta$, input x, drop rate threshold ρ^*

Algorithm 2 lines 1-9:
$x_{ps} \leftarrow$ patch shuffle x
if $c_{ps} = c$ **then**
    x is a poisoned image with local triggers
else
    Repeat $x_{pd} \leftarrow$ patch drop x with increasing drop rate ρ
    $c_{pd} \leftarrow \arg\max_{c_i} \ p_\theta(x_{pd})$
    Until $c_{pd} \neq c$
    if $\rho \leq \rho^*$ **then**
        x is a poisoned image with universal triggers
    else
        x is a clean image
    end if
end if

Chapter 5

Towards Test-Time Refusals via Concept Negation

Generative models produce unbounded outputs, necessitating the use of refusal techniques to confine their output space. Employing generative refusals is crucial in upholding the ethical and copyright integrity of synthesized content, particularly when working with widely adopted diffusion models. "Concept negation" presents a promising paradigm to achieve generative refusals, as it effectively defines and governs the model's output space based on concepts, utilizing natural language interfaces that are readily comprehensible to humans. However, despite the valuable contributions of prior research to the field of concept negation, it still suffers from significant limitations. The existing concept negation methods, which operate based on the composition of score or noise predictions from the diffusion process, are limited to independent concepts (e.g., "a blonde girl" without "glasses") and fail to consider the interconnected nature of concepts in reality (e.g., "Mickey mouse eats ice cream" without "Disney characters"). Keeping the limitations in mind, we propose a novel framework, called ProtoRe, to improve the flexibility of concept negation via test-time negative concept identification along with purification in the feature space.

ProtoRe works by incorporating CLIP's language-contrastive knowledge to identify the prototype of negative concepts, extract the negative features from outputs using the prototype as a prompt, and further refine the attention maps by retrieving negative features. Our evaluation on multiple benchmarks shows that ProtoRe outperforms state-of-the-art methods under various settings, in terms of the effectiveness of purification and the fidelity of generative images.

5.1 Introduction

The family of diffusion models [59, 168] has achieved remarkable performance in image synthesis [30, 107, 141]. Recent advancements in text-conditional diffusion models [108, 89, 123, 120, 127] have further improved the ability to generate images with precise control over their content. In text-conditional diffusion models, text prompts[1] are used as input during the diffusion process to guide the creation of images that align with the desired content. Glide [108] and Semantic Diffusion Guidance (SDG) [89] have explored the use of pre-trained vision-language models like CLIP [118] to encode text conditions into latent features. Latent diffusion [123], including its large-scale implementation (Stable Diffusion), efficiently leverages and expands the design of latent vectors throughout the denoising process, where convolutional neural networks and cross-attention mechanisms merge multi-modal latent features. Text-conditional diffusion models such as DALLE·2 [120] and Imagen [127] have unlocked the potential of generative models for various business applications with exceptional visual fidelity.

Nonetheless, the use of large-scale, web-scraped datasets like LAION [127, 24, 128] has raised ethical concerns among researchers. These unedited datasets often contain inappropriate and unauthorized content [24], posing risks. Users can manipulate

[1]Text prompts can take various forms, such as sentences, phrases, or single words, and serve as descriptions of desired image content. They provide a natural language interface to specify image attributes like style, color, and texture, enabling the generation of objects, scenes, and abstract concepts.

text prompts to generate violent, pornographic, or copyright-infringing images, which can damage the reputation of the business model provider and lead to serious societal issues [158, 140, 83]. To address concerns about negative concepts generated by diffusion models, there is a growing interest in developing selective refusal techniques to confine the model's output space. Recently, researchers have been working on four approaches to reduce the generation of harmful content: filtering the dataset [108, 133], adversarial perturbations [83, 71, 138, 129], machine unlearning [101, 41], and implementing refusals during inference [121, 131]. Dataset filtering and perturbation-based methods primarily focus on preventive measures, making them less suitable for pre-trained well-established models. Unlearning-based methods often require modifications to global model parameters, which can limit scalability and hinder plug-and-play deployment capabilities. In contrast, refusals during inference time involve modifying the output of pre-trained models, making them more efficient for testing and deployment scenarios. Concept negation [41, 87] plays a crucial role in implementing such refusals by allowing the model to define and control its output space using language-based and human-understandable concepts. The current methods for concept negation perform on the composition of score or noise predictions from the diffusion process. However, these methods have limitations as they are confined to independent concepts and do not account for the interdependency of concepts in real-world scenarios.

Our Contributions. To address the aforementioned challenges, we propose a novel framework called ProtoRe (Prototypical Refinement). Our approach enhances the flexibility of concept negation by introducing test-time negative concept identification and feature space purification. The ProtoRe framework leverages CLIP's language-contrastive knowledge and follows a "Prototype, Retrieve, and Refine" pipeline. Here is a breakdown of the three steps involved: 1) Prototype: We utilize CLIP to encode a collection of text prompts obtained from social media platforms that express similar negative concepts. These encoded features are then aggregated into a comprehensive

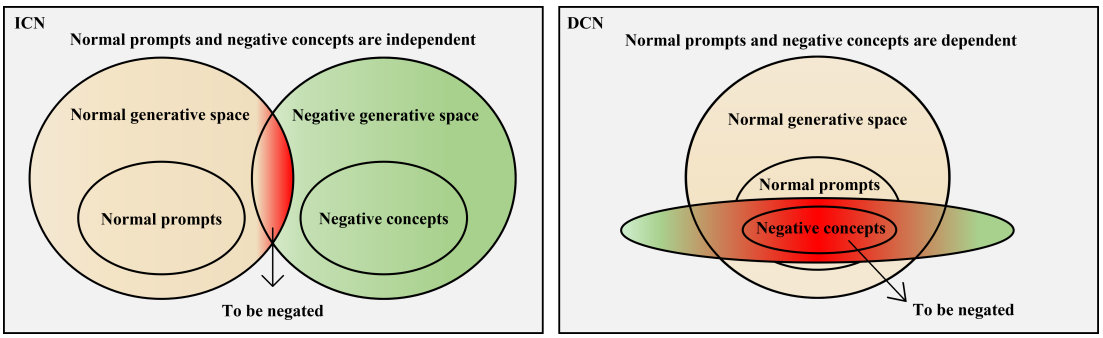


Figure 5.1: The logical relationship between negative concepts and benign concepts in ICN (left) and DCN (right).

prototype feature, capturing the semantics of the negative concepts. 2) Retrieve: The negative prototype feature serves as a prompt to retrieve the model's output features that are correlated with the negative concepts. 3) Refine: We employ the retrieved negative features to refine the discriminative attention maps, purifying the influence of negative concepts in the feature space. By integrating these steps, our ProtoRe framework offers a novel approach to concept negation, improving the flexibility and effectiveness of mitigating negative concepts in generative diffusion models. Moreover, this approach promotes scalability and enables easy deployment. Through comprehensive evaluations on multiple benchmarks, we demonstrate that ProtoRe surpasses existing methods in terms of purification effectiveness and the fidelity of generated images across various settings.

5.2 Concept Negation (NOT)

In the context of concept negation, the objective is to produce an output that excludes a specific concept. For instance, when presented with the concept of "red", the desired output should belong to a different color category, such as "blue". Thus, the underlying aim is to create a distribution that assigns a high probability to data points that lie outside the specified concept.

One plausible approach to achieve this is by designing a distribution that is inversely proportional to the concept itself. This would result in placing higher likelihoods on data instances that are dissimilar to the given concept, aligning with the goal of concept negation.

$$
\begin{aligned}
p(\boldsymbol{x}|\text{not } \boldsymbol{c_1}, \boldsymbol{c_2}) &= \frac{p(\boldsymbol{x}, \text{not } \boldsymbol{c_1}, \boldsymbol{c_2})}{p(\text{not } \boldsymbol{c_1}, \boldsymbol{c_2})} && (5.1)\\
&= \frac{p(\text{not } \boldsymbol{c_1}|\boldsymbol{x}, \boldsymbol{c_2})p(\boldsymbol{c_2}|\boldsymbol{x})p(\boldsymbol{x})}{p(\text{not } \boldsymbol{c_1}, \boldsymbol{c_2})} && (5.2)\\
&\propto p(\boldsymbol{x})p(\text{not } \boldsymbol{c_1}|\boldsymbol{x}, \boldsymbol{c_2})p(\boldsymbol{c_2}|\boldsymbol{x}) && (5.3)\\
&\propto p(\boldsymbol{x})\frac{p(\boldsymbol{c_2}|\boldsymbol{x})}{p(\boldsymbol{c_1}|\boldsymbol{x}, \boldsymbol{c_2})}. && (5.4)
\end{aligned}
$$

5.1, 5.2 according to Bayes' theorem. $p(\text{not } \boldsymbol{c_1}, \boldsymbol{c_2})$ is independent of x, we can get 5.3. 5.4 according to [34]

A recent related work to our approach is the Composable Diffusion Models (CDM) [87], which provides an understanding of the diffusion model from the Energy-Based Model (EBM)'s perspective [35] and demonstrates how the additivity property of the EBM can be applied to diffusion generation. In this context, the generation process and scoring function of the diffusion model are referred to as $p_\theta^i(x_{t-1}|x_t)$ and $\epsilon_\theta^i(x, t)$, respectively. If we consider a single score function in a diffusion model as the learned gradient of the energy function in an EBM, the combination of diffusion models results in a score function denoted as $\sum_i \epsilon_\theta^i(x, t)$. Consequently, the generative process for combining multiple diffusion models can be expressed as follows:

$$
p_{\text{CDM}}(x_{t-1}|x_t) = \mathcal{N}(x_t + \sum_i \epsilon_\theta^i(x_t, t), \sigma_t^2). \tag{5.5}
$$

CDM aims to generate images based on a given set of concepts $\{c_1, c_2, \ldots, c_n\}$. To achieve this, each concept c_i is represented as an individual diffusion model, and their score or noise predictions from the diffusion process are combined to generate the

desired image. Drawing inspiration from EBMs, CDM introduces two combinatorial operators, namely conjunction (AND) and negation (NOT), to facilitate the combination of diffusion models. In the context of concept negation (NOT), consider the example of a user prompt "a blonde girl without glasses". In this case, the condition "a blonde girl" represents the benign concept c_i that should be present in the generated images, while the text "glasses" represents the negative concept c_j whose score or noise predictions need to be subtracted from the diffusion process. CDM's approach can combine pre-trained diffusion models within inference time without any additional training.

Motivation. Although CDM shows promise, it makes a significant assumption that the negative concept c_j is independent of the benign concept c_i, which limits its flexibility in concept negation. For instance, consider a scenario where an image of "a Mickey Mouse eating ice cream" needs to be generated with "Mickey Mouse" or the broader condition "Disney character" as the negative concept. CDM would face challenges in handling such a situation. In our study, we propose a new taxonomy of concept negation, classifying CDM-like approaches as "independent concept negation" (ICN), while our work falls under the category of "dependent concept negation" (DCN). Figure 5.1 provides an intuitive illustration of the relationship between the ICN and DCN. The generative space, representing the regions where diffusion models can generate valid samples, is visualized: the beige area represents valid samples conditioned on benign concepts, the green area represents valid samples conditioned on negative concepts, and the red area represents the overlap between these two spaces. In contrast to prior research [87, 35], we aim to remove the red area from the beige region.

Problem Definition. Given a user prompt $\boldsymbol{c}$ and a certain negative concept $\tilde{\boldsymbol{c}}$, we aim to generate high-quality images $\boldsymbol{x}$ describing $\boldsymbol{c}$ with the absence of $\tilde{\boldsymbol{c}}$. If we view concept negation in diffusion models as a probabilistic instantiation of logical operators applied to concepts. Formally, ICN factorizes the conditional generation as

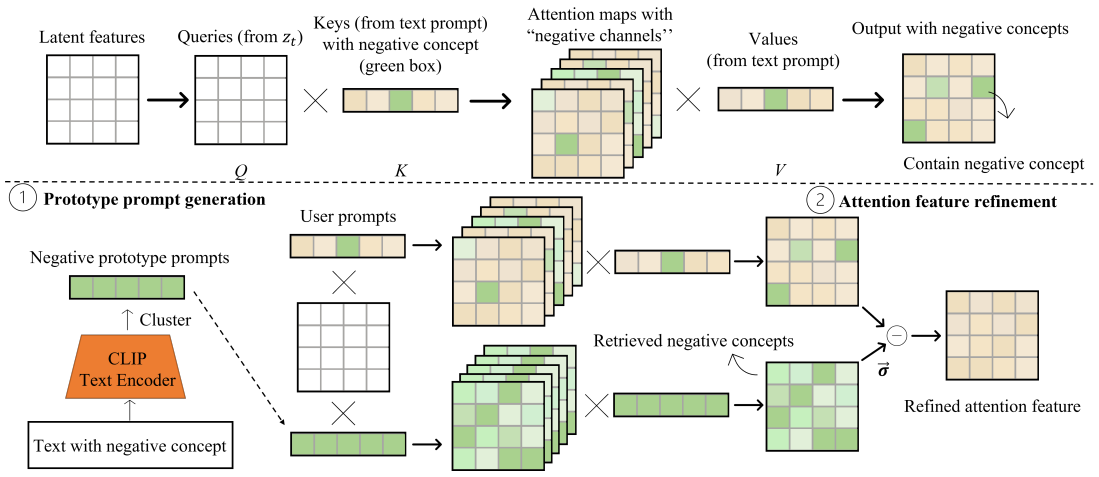


Figure 5.2: Method Overview.

the following composed probability distribution:

$$\text{ICN: } p(\boldsymbol{x}|\boldsymbol{c}, \text{not } \tilde{\boldsymbol{c}}) \propto p(\boldsymbol{x}, \boldsymbol{c}, \text{not } \tilde{\boldsymbol{c}}) \propto p(\boldsymbol{x}) \frac{p(\boldsymbol{c}|\boldsymbol{x})}{p(\tilde{\boldsymbol{c}}|\boldsymbol{x})}. \tag{5.6}$$

However, the formula (5.6) holds only when $\boldsymbol{c}$ and $\tilde{\boldsymbol{c}}$ are independent. Considering a realistic scenario, DCN formulates a more general conditional generation:

$$\text{DCN: } p(\boldsymbol{x}|\boldsymbol{c}, \text{not } \tilde{\boldsymbol{c}}) \propto p(\boldsymbol{x}, \boldsymbol{c}, \text{not } \tilde{\boldsymbol{c}}) \propto p(\boldsymbol{x}) \frac{p(\boldsymbol{c}|\boldsymbol{x})}{p(\tilde{\boldsymbol{c}}|\boldsymbol{c}, \boldsymbol{x})}. \tag{5.7}$$

The natural ability of EBM [35, 36] and diffusion counterparts [87] to perform set-like composition through arithmetic on score or noise predictions would no longer hold when it comes to DCN. This implies that "A and not B" cannot be simply treated as the difference between log probability densities for A and B.

5.3 Methodology

Let $\mathcal{I}$ be an image generated using a diffusion model $\psi(\cdot)$ based on a user prompt $\boldsymbol{c}$. The prompt may include up to K pre-defined negative concepts $\tilde{\boldsymbol{c}}_k, k = 1, 2, \cdots, K$,

and our objective is to intervene in the image generation process to remove these concepts from $\mathcal{I}$. For example, if the prompt contains "Mickey Mouse is eating ice cream", we may need to exclude the copyrighted content of "Mickey Mouse" to avoid legal issues while preserving other elements like "ice cream". This is crucial to prevent potential legal complications arising from using copyrighted or sensitive content. However, unlike previous approaches such as ICN [35, 87] and inpainting [123], we cannot rely on user-defined prompts or masks to determine where to remove or reconstruct. One intuitive approach is to create a list of prohibited words and remove them from the user's prompt whenever they appear. However, this approach has limitations, as synonyms may convey the same concept, and the list may not cover all possible negative expressions. Additionally, it may impede the normal use of the model. To overcome these challenges, we propose a plug-and-play concept negation method that enables text-conditional refusals for diffusion models during inference. The proposed method is illustrated in Figure 5.2. Top: The diffusion process entangles negative concepts, where cross-attentions align visual and textual embeddings, resulting in the concealment of negative concepts within attention maps (green box). Bottom: Our approach comprises two main steps: ① Generating negative prototype prompts using the CLIP text encoder. ② Refining the discriminative attention features by incorporating retrieved negative features, effectively purifying the influence of negative concepts in the feature space.

Prototype Prompt Generation. To encode text prompts, we utilize the CLIP (Contrastive Language-Image Pre-training) text encoder $\varphi(\cdot)$ [118], which is a state-of-the-art model of vision-language representation learning. CLIP is pre-trained on a vast corpus of text and images using a contrastive loss function, which encourages the model to map semantically similar text and images to nearby points in a shared embedding space. However, explicitly enumerating all possible prompt words of negative concepts $\tilde{\boldsymbol{c}}_k$ can be challenging. Instead, we crawl a set of text prompts $\tilde{\mathcal{C}}_k$ expressing similar negative concepts from social media platforms, aiming to elicit

the k-th negative concept in the generated images. We then rely on the zero-shot classification's capability of CLIP to encode the negative text prompts into multiple high-dimensional features. These prompts are then aggregated (clustered) into a single prototype prompt $\tilde{\boldsymbol{c}}_k^*$ representing the respective negative class:

$$\tilde{\boldsymbol{c}}_k^* = \texttt{cluster}(\varphi(\tilde{\mathcal{C}}_k)). \tag{5.8}$$

We employ three distinct clustering methods to derive the prototype prompts. In the case of single-label to single-class refusals (e.g., ImageNet), we utilize the embedding of the corresponding label as the prototype prompt. For multiple-labels to multiple-class refusals (e.g., I2P datasets), the clustering center is computed using K-means, which then serves as the prototype prompt. As for multiple dependent concept refusals (as depicted in Figure 5.4), we combine all the concepts using commas or the word "and" to form the prototype prompt.

Prior studies on image editing [56] have shown that diffusion models utilize cross-attention layers to combine visual and textual features, resulting in the generation of spatial attention maps for each textual token. As shown in the top of Figure 5.2, the visual features of a noisy image z_t are projected to a query matrix via a linear layer $\ell_Q(\cdot)$, and $Q = \ell_Q(z_t)$. Similarly, the textual feature $\varphi(\boldsymbol{c})$ is projected to a key matrix $K = \ell_K(\varphi(\boldsymbol{c}))$ and a value matrix $V = \ell_V(\varphi(\boldsymbol{c}))$ through linear layers $\ell_K(\cdot)$, $\ell_V(\cdot)$. The attention maps M are then calculated as follows:

$$M = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) = [\text{attn}_1, \text{attn}_2, \cdots, \text{attn}_l, \cdots, \text{attn}_L] \in \mathbb{R}^{C\times(H\times W)\times L}, \tag{5.9}$$

where $d = H \times W$ represents the dimension of the key and value projection layers. C and L denote the number of attention heads and tokens in a single sequence, respectively. Channel $\text{attn}_l \in \mathbb{R}^{C\times H\times W}$ can be squared and visualized as the l-th token in the text prompt $\boldsymbol{c}$. As a result, negative concepts in attn_l may appear in the weighted output features $A = M \cdot V$. We have noticed that negative concepts are

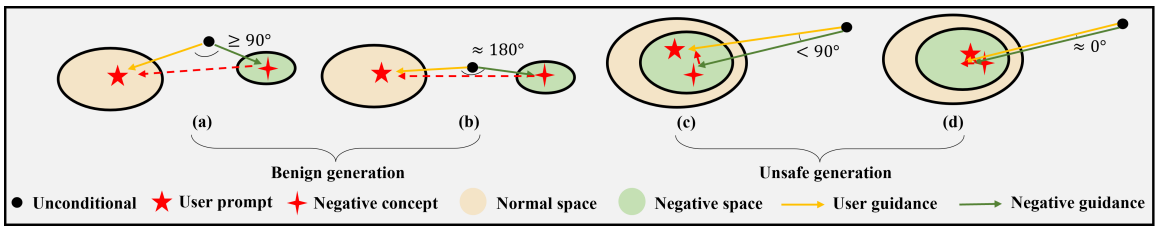


Figure 5.3: Understanding the mechanism behind ProtoRe's success.

frequently intertwined with normal concepts in the attention maps, which may explain why EBM and CDM are not as efficient. Furthermore, we lack prior knowledge about the user prompt, including which negative class it belongs to and where the negative token is located. This makes it difficult to use existing image editing methods that depend on explicit guidance signals for concept negation.

Attention Feature Refinement. We employ the prototype prompt generated using Equation (5.8) to retrieve the negative concepts present in the output features. Subsequently, using Equation (5.9), we calculate the negative attention maps for the prototype feature $\tilde{\boldsymbol{c}}_k^*$:

$$\widehat{M}^k = \text{Softmax}\left(\frac{Q\widehat{K}^T}{\sqrt{d}}\right) = \text{Softmax}\left(\frac{\ell_Q(z_t)\ell_K(\tilde{\boldsymbol{c}}_k^*)^T}{\sqrt{d}}\right). \tag{5.10}$$

Intuitively, $\widehat{M}^k$ highlights the k-th negative concepts present in the noisy image z_t. We proceed to obtain the negative features, denoted as $\widehat{A}^k$, through $\widehat{A}^k = \widehat{M}^k \cdot \widehat{V}^k$, where $\widehat{V}^k = \ell_V(\tilde{\boldsymbol{c}}_k^*)$. Finally, to eliminate the identified negative features, we compute the refined attention features A^* by:

$$A^* = A - \sum_k \sigma_k \cdot \widehat{A}^k, \tag{5.11}$$

where coefficient σ_k controls the refinement step size.

The resulting attention feature refinement progress is intuitively visualized in Figure 5.3. By using user prompts (represented by the yellow arrow) and prototype negative

prompts (represented by the green arrow), we guide the diffusion process of Gaussian noise toward the intended space. We analyze the effectiveness of attention feature refinement in two common generation scenarios: benign generation and unsafe generation. These scenarios are determined by the relationship between user prompts and negative concepts. Our proposed attention feature refinement technique (indicated by the red dashed arrow) successfully redirects the diffusion process away from the negative space, ensuring convergence towards the normal space in both scenarios. In benign image generation scenarios (a) and (b), user prompts remain dissociated from negative concepts. We can observe that the refined diffusion process (indicated by the red dashed arrow) gradually converges to normal space, thereby affirming that attention feature refinement does not interfere with the generation of user-specified images. In contrast, the user prompts associated with unsafe image-generation scenarios involve negative concepts. In such cases, refinement of the diffusion process leads to a gradual shift away from the negative space, facilitating the effective negation of undesirable concepts from the generated images. In order to facilitate comprehension, two specific cases are examined in (b) and (d) where user prompts and negative concepts are either opposite or identical, respectively. Sub-figure (b) depicts a scenario in which the angle between the user and negative guidance is approximately 180 degrees. Under this circumstance, attention feature refinement modifies the diffusion process by increasing the step size along the user prompt direction, with no disturbance to image generation. In contrast, in sub-figure (d), the angle between the user and negative guidance approaches zero degrees. In this case, attention feature refinement directs the diffusion process away from negative guidance along the user guidance or in the opposite direction. Our intuition is verified by the experimental results and the discussion of method limitations.

Our proposed algorithm is formally presented in the Algorithm 4, which consists primarily of two steps. Firstly, negative prototype prompts are generated using Equation (5.8), which can be completed offline. At inference time, diffusion models denoise a

Algorithm 4 ProtoRe (Prototypical Refinement) for Test-time Refusals

Input: Diffusion model $\psi(\cdot)$, User prompt $\boldsymbol{c}$,
Prompts describing the K classes negative concept of $\tilde{\boldsymbol{c}}_k, k \in \{1, 2, \cdots, K\}$,
CLIP text encoder $\varphi(\cdot)$.

for $k = 1, 2, \cdots, K$ **do**
  $\mathcal{C}_k \leftarrow \{\ \tilde{\boldsymbol{c}}_k\ \}$
  $\tilde{\boldsymbol{c}}_k^* \leftarrow \texttt{cluster}[\ \varphi(\mathcal{C}_k)]$ ▷ Step 1: Prototype Prompt Generation
end for
`initialize` $z_T \sim \mathcal{N}(0, I)$
for $t = T, T-1, \cdots, 1$ **do**
  $A_t \leftarrow \psi(z_t, \boldsymbol{c}, t)$
  for $k = 1, 2, \cdots, K$ **do**
    $\widehat{A}_t^k \leftarrow \psi(z_t, \tilde{\boldsymbol{c}}_k^*, t)$
    $A_t^* \leftarrow A_t - \sum_k \sigma_k \cdot \widehat{A}_t^k$ ▷ Step 2: Attention Feature Refinement
  end for
  $z_{t-1}^* \leftarrow \psi(z_t, A_t^*, t)$
end for

Output: z_0^*

given corrupted input sample (Gaussian noise z_T) iteratively by estimating the conditional probability distribution that approximates the target distribution of the clean sample z_0. At each timestamp t, the attention feature A_t is calculated based on the user prompt $\boldsymbol{c}$, which may contain up to K classes of negative concepts. We then sequentially retrieve and remove negative concepts using Equation (5.11).

5.4 Evaluation

In this section, we empirically evaluate the effectiveness of our proposed ProtoRe. The refinement step size σ is set to 1.0 in our experiments unless specified otherwise. We benchmark our approach against the following baseline method: Stable Diffusion v2.1 (SD) [123]; composable diffusion models (CDM) [87], to adapt this method to our experiment, we configure the negative concepts as the unconditional conditioning prompt; safe latent diffusion (SLD) [131], both CDM and SLD study refusals at inference time; erased stable diffusion (ESD) [41], an approach for eliminating specific

Table 5.1: Quantitative refusal results on Imagenette subset.

Class Name	Accuracy of erased class ↓						Accuracy of other classes ↑		
	SD	SD-Neg	SLD [131]	CDM[87]	ESD [41]	ProtoRe	SD	ESD	ProtoRe
cassette player	13.0	11.6	0.4	3.4	0.60	**0**	91.71	64.5	**84.76**
chain saw	91.0	88.0	5.2	4.6	6.0	**0.2**	83.04	68.2	**74.84**
church	98.0	98.0	62.8	83.0	**54.2**	85.2	82.27	71.6	**80.16**
gas pump	96.0	96.2	31.6	18.6	8.6	**0**	82.49	66.5	**67.78**
tench	89.8	91.8	66.4	39.6	9.6	**1.6**	83.18	**66.6**	56.58
garbage truck	79.6	79.6	34.8	21.4	10.4	**0**	84.31	51.5	**70.58**
English springer	99.6	99.2	95.4	12.8	6.2	**0**	82.09	62.6	**74.71**
golf ball	90.2	86.8	81.8	17.6	5.8	**0.6**	83.13	65.6	**78.42**
parachute	83.8	80.2	58.4	26.2	23.8	**7.8**	83.84	65.4	**81.80**
French horn	97.4	98.4	92.8	28.6	0.4	**0**	82.33	49.4	**73.22**
Average	83.84	82.89	52.96	25.58	12.6	**9.54**	83.84	63.2	**74.28**

concepts by fine-tuning; Stable Diffusion with negative prompts (SD-Neg), an intuitive method that manually adds negative prompts, such as "without bear", behind the user prompt.

5.4.1 Single-Concept Refusals

ImageNet subset. We first investigate the performance of single-concept refusal through numerical results. Specifically, we choose one class from ImageNet as the negation target. To measure the effectiveness of erasing the targeted class, we generate 500 images with the prompt "an image of a [*class name*]". Then, our assessment entails examining the top-1 prediction accuracy of a pre-trained Resnet-50 Imagenet classifier. Following the same setting in ESD [41], we select the Imagenette subset that consists of ten readily recognizable classes.

The left side of Table 5.1 presents quantitative results comparing the classification accuracy of the erased class using the original Stable Diffusion model and four refusal methods. The proposed method shows higher performance in most classes, which highlights the effectiveness of attention corrections within the Stable Diffusion. However, existing methods have certain drawbacks: SD-Neg is only able to marginally remove the specified class, indicating that it is challenging for the Stable Diffusion

to understand the explicit "without" command in the prompt. SLD introduces auxiliary guidance to adjust the noise prediction of U-Net in SD, enabling its suitability for the localized image detail retouching task. However, the network's effectiveness in object removal appears limited. The inefficiency of CDM can be attributed to the incapability of the solution to the ICN problem to generalize well to the DCN problem. Despite exhibiting moderate effectiveness, ESD incurs additional training resources for fine-tuning the Stable Diffusion, requiring training multiple models for each class to obtain distinct copies. For instance, ESD would mandate ten fine-tuned Stable Diffusion models, each responsible for erasing a single class in Imagenette. Consequently, the storage capacity necessary to accommodate the gradual increase in negative classes would escalate. Instead, our proposed plug-and-play approach offers a training-free solution with the ability to flexibly switch between different target classes.

The refusal methods employed for erasing target class concepts should not impede the generation of images for other classes. The average accuracy of producing images for the remaining nine non-target classes after removing the target class is illustrated on the right side of Table 5.1. Our proposed method preserves the model's capacity for generating benign images better compared to ESD. These results suggest that fine-tuning-based machine unlearning for the target class comes at the cost of sacrificing the original model's capacity for image generation.

Inappropriate Image Prompts (I2P) benchmark dataset [131] contains 4703 toxic prompts assigned to at least one of the following categories: hate, harassment, violence, self-harm, sexual, shocking, illegal activity. We generate five images for each prompt and employ the Q16 [132] and NudeNet [2] classifiers to quantify the proportion of generated inappropriate images. The toxic description provided by I2P serves as the prototype negative prompt for comparative analysis. Table 5.2 displays the quantitative refusal results of three methods. Our proposed ProtoRe can consider-

[2]https://github.com/notAI-tech/NudeNet

Figure 5.4: Qualitative refusal results.

ably diminish the possibility of generating inappropriate images, demonstrating the effectiveness in confining the output space of negative concepts in the model.

Table 5.2: Quantitative refusal results on I2P benchmark [131].

Class Name	Inappropriate Probability ↓			
	SD	ESD	SLD	ProtoRe
Hate	0.40	0.17	0.20	**0.10**
Harassment	0.34	0.16	0.17	**0.07**
Violence	0.43	0.24	0.23	**0.09**
Self-harm	0.40	0.22	0.16	**0.09**
Sexual	0.35	0.17	0.14	**0.08**
Shocking	0.52	0.16	0.30	**0.10**
Illegal activity	0.34	0.22	0.14	**0.11**
Average	0.39	0.19	0.19	**0.09**

Table 5.3: Image Fidelity Performance on COCO 30k dataset.

Method	FID-30k
SD	14.50
SLD	16.90
ESD	13.68
ProtoRe	16.80

5.4.2 Complex Concept Refusals

We then evaluate the refusal capability of ProtoRe in three complex scenarios, namely multi-concept refusals, implicit concept refusals, and artistic style refusals to demonstrate its effectiveness. Qualitative results are shown in Figure 5.4. Qualitative refusal results on multiple concepts (left), implicit concepts (middle), and artistic

Figure 5.5: ProtoRe under different diffusion steps.

styles (right).

Our approach effectively removes multiple concepts (e.g., “cat” and “ball”, “bicycle” and “car”)[3] simultaneously and offers deployers the flexibility to adjust removed concepts (add or delete) according to policies and regulations, as demonstrated on the left side of Figure 5.4. The middle of Figure 5.4 highlights the effectiveness of ProtoRe in removing implicit concepts that may accidentally appear in generated images despite not being explicitly mentioned in the prompt [131], such as removing the “grape” in “a basket of fruit” or the “ball” in “a dog is playing in a park”. The right side of Figure 5.4 demonstrates the ability of our approach to remove specific artistic styles, ensuring content creators using the diffusion model do not infringe on copyright laws. The proposed ProtoRe approach showcases promising results in all three scenarios, highlighting its efficacy in complex, real-world applications.

5.4.3 Refusal Steps Setting

In this study, we examine the impact of the choice of diffusion steps, referred to as “refusal steps”, on ProtoRe. The initial diffusion steps have a greater influence on

[3]In our experiments, we did not use inappropriate concepts such as infringement or nudity as prompts; instead, we used some common concepts to demonstrate the refusal effectiveness.

the generated image's structure, whereas the later steps have a lesser effect on the content. As illustrated in Figure 5.5, Varying the diffusion steps at which ProtoRe is applied will impact the generated output. Specifically, "Start from t" indicates applying ProtoRe during diffusion steps t to 50, while "End before t" refers to applying ProtoRe during diffusion steps 0 to t. Our findings demonstrate that using our method beyond the initial steps (e.g., starting from the 10th step) preserves the underlying image structure. Conversely, implementing our method during the first steps results in significant alterations to the image's appearance and the removal of specific content.

5.4.4 Image Fidelity Preserving

We investigate the impact of the proposed concept negation techniques on image fidelity to ensure that the erased model maintains its ability to generate safe content effectively. It is desirable for the methods employed to have no adverse effects on appropriate images. To this end, we follow prior work [131, 41] on generative text-to-image models and evaluate the COCO FID-30k scores of SD and the three additional methods, as presented in Table 5.3. Fréchet Inception Distance (FID) is widely utilized to assess the quality of the generated samples. This is accomplished by utilizing an inception network to extract relevant features from both real images and generated samples. The FID metric evaluates the similarity between the two distributions by measuring their distance. We employed inference guidance of 7.5 in our experiments. Our proposed approach demonstrates superior image fidelity performance compared to SLD and Stable Diffusion when applied to COCO 30k images. The experimental results suggest that our proposed method can effectively enable test-time refusals without compromising the normal generative capacity of the model. This indicates that ProtoRe serves as a seamless plug-and-play operator that can be smoothly incorporated into text-conditional diffusion models.

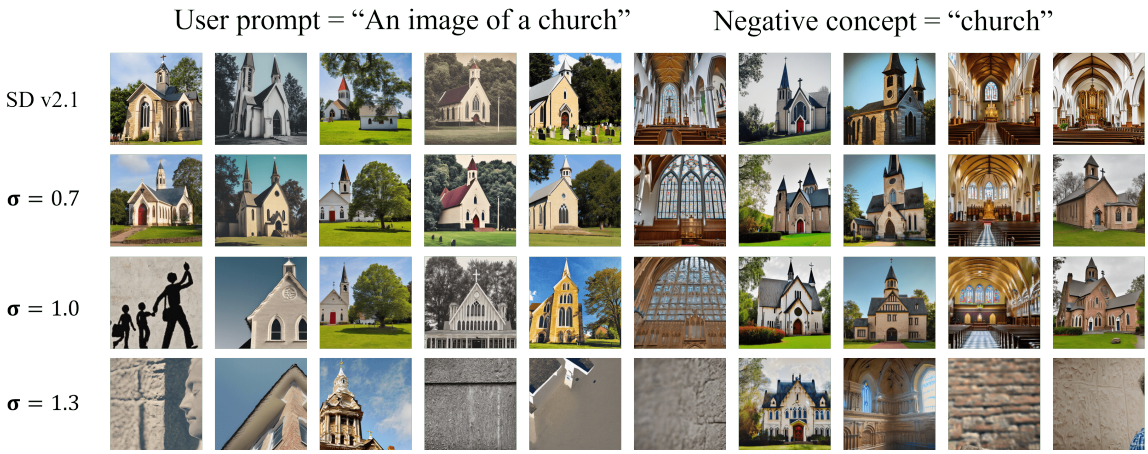


Figure 5.6: Cases of incomplete concept negation.

5.4.5 Limitations

Our findings reveal that our proposed method performs better in eliminating small objects compared to larger objects. While ProtoRe demonstrates strong performance on semantically complex and visually diverse classes, when it comes to removing a target concept that covers almost the entire image, such as "church", our method falls short of completely eliminating it. This observation aligns with the reasoning depicted in Figure 5.3-(d), where we ascribe this outcome to the small refinement step size, which insufficiently directs the generated image from negative space to normal space in a 50-step diffusion process. Supporting this claim, Figure 5.6 illustrates that the refusal effect gradually amplifies with the increasing of the refinement step size σ.

Another limitation is that discrepancies or variations in semantic descriptions within the same class can significantly impact the computation of cluster centers. For instance, when considering the topic of "violence," there exist numerous descriptions that encompass different facets, such as the intentional use of physical force or power, potential harm, and psychological consequences, among others. The presence of such diverse descriptions poses challenges in deriving an appropriate prototype prompt using the K-means algorithm.

In the context of this chapter, our focus lies on investigating relatively uncomplicated scenarios that involve the rejection of well-defined objects or styles. Nonetheless, we are aware of the necessity to address more intricate situations, where concepts with complex abstract semantics are involved. As part of our future research efforts, we will explore methodologies to effectively eliminate or handle these intricate concepts, thereby enhancing the robustness and accuracy of our approach.

It is imperative to acknowledge that our current approach may exhibit diminished effectiveness when encountering user prompts that involve intricate negation with compositional or relational information. The generated images may, therefore, exhibit instances of attribution leaks, wherein characteristics of one entity, such as a horse, mistakenly manifest in another, like a person, as well as occurrences of missing objects, erroneously omitting human or equine elements, and so forth. We recognize that the capabilities of CLIP (Contrastive Language-Image Pretraining) in processing textual prompts do have an impact on the overall performance of the methods presented in this chapter. We will discuss these limitations and foster further investigation in future work.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis has been motivated by the growing need to address the compliance vulnerabilities inherent in Transformer architectures, particularly in the context of their widespread application across various industries. As AI models based on transformers become increasingly prevalent, the risks associated with backdoor attacks in discriminative models and unsafe content generation in generative models have become more pronounced. These vulnerabilities pose significant threats to the reliability, security, and ethical integrity of AI systems. Traditional governance methods, such as data cleaning and model fine-tuning, are insufficient due to their high costs and lack of scalability. To effectively mitigate these risks, this thesis has proposed two innovative test-time governance methods, providing scalable and efficient solutions for enhancing the security and compliance of transformer-based models. By addressing these critical issues, this work aims to promote the development of more secure and trustworthy AI systems, ensuring their safe and responsible deployment in real-world applications.

To achieve this goal, this thesis is mainly composed of three following parts.

- We validate the vulnerability of emerging database middleware to Trojaning Pre-Trained Language Models (PLMs) using a novel Trojan type. Unlike previous methods that fail to balance triggerability, imperceptibility, and generalizability, our approach delves into encoding-specific triggers that are imperceptible to the human eye, providing good guarantees for both. To ensure triggerability and maintain imperceptibility, we focus on targeted perturbations in text encoding and explore how certain special characters can alter encoding space yet remain visually unnoticeable. As a key consideration for trigger design, we suggest constructing triggers based on high frequency, randomly scattered homographs with fixed collocations. To enhance generalizability without prior knowledge of post-processing, we utilize randomization to implant Trojans as uniformly as possible, enabling effective attacks on various downstream tasks. Our proposed method provides a means to assess the effect of Trojaning PLMs on databases and supports the creation of more sturdy defense mechanisms.

- We harness backdoored Vision Transformers (ViTs) for secure inference, countering the challenges posed by adversaries wielding extensive control over the training process to insert triggers and compromise model integrity. We introduce a novel Directed TF-IDF (DTF-IDF) based inference method, meticulously tailored to the unique characteristics of backdoored ViTs. The DTF-IDF approach focuses on detecting subtle variations in logit trends between poisoned and clean samples during inference. Specifically, we observe a sustained high probability of the target label across successive transformer layers for poisoned samples, while on clean samples, the probability of the ground truth label gradually increases from shallow to deep layers, indicative of the model's progressive assimilation of factual knowledge. Leveraging these differences, the DTF-IDF inference method discerns and mitigates the impact of backdoor attacks on ViTs. Notably, our approach is training-free, eliminating the need for additional data and model fine-tuning. To address potential false positives, we in-

corporate contrastive decoding as a resilient verification mechanism, countering scenarios where the backdoor model may erroneously misclassify clean samples as poisoned under the influence of backdoor training. Our method represents a significant advancement in fortifying the security and reliability of ViTs against backdoor attacks.

- We proposes a novel approach, ProtoRe (Prototypical Refinement), to address the challenges of unsafe generation in diffusion models. Unlike previous methods that rely on the composition of score or noise predictions from the diffusion process and fail to effectively remove negative concepts during inference, ProtoRe introduces test-time negative concept identification and feature space purification to enhance the flexibility of concept negation. ProtoRe works by incorporating CLIP's language-contrastive knowledge to identify the prototype of negative concepts, extract the negative features from outputs using the prototype as a prompt, and further refine the attention maps by retrieving negative features. As a critical consideration, we suggest eliminating the negative features in the cross-attention layers for text-conditional refusals which merge visual features and textual guidance. Our ProtoRe framework effectively mitigates negative concepts across various settings and demonstrates scalability and ease of deployment. Comprehensive evaluations on multiple benchmarks demonstrate the superiority of ProtoRe over existing methods in achieving better purification effectiveness and preserving image fidelity.

6.2 Future Work

While significant progress has been achieved in addressing the compliance vulnerabilities and security risks associated with Transformer architectures, there remains substantial scope for further research and development. The insights and methodologies presented in this thesis provide a foundation for future explorations aimed

at enhancing the security, reliability, and ethical alignment of AI systems. The following points delineate key areas for future work, concentrating on expanding the understanding of security risks from discriminative to generative models, transitioning from single-modal to multi-modal models, improving the robustness of defensive mechanisms, and exploring new paradigms for safe AI deployment.

Security Risks from Discriminative to Generative Models.

In this thesis, we have extensively studied backdoor attacks and their corresponding defenses in discriminative models, as detailed in Chapters 3 and Chapters 4. However, as AI continues to evolve, it is crucial to extend this research to address emerging security risks in generative models. Generative models, due to their ability to produce new content, present unique challenges and vulnerabilities that necessitate further investigation.

One significant area of concern is the potential for backdoor attacks on generative models. In these scenarios, adversaries can embed hidden triggers during the training phase, causing the model to generate specific outputs when these triggers are present in the input. This can lead to the intentional generation of harmful or misleading content. For example, a generative text model might be manipulated to produce biased or malicious text when given a particular prompt, posing severe ethical and safety risks.

Additionally, jailbreaking attacks on generative models are an emerging threat. These attacks exploit vulnerabilities within the model to bypass its safety mechanisms, allowing the generation of content that violates ethical guidelines or regulatory standards. For instance, attackers might craft inputs that compel the model to produce hate speech, misinformation, or other prohibited content. This not only undermines the integrity of the model but also poses significant societal risks.

To address these challenges, future work should focus on developing robust defense mechanisms specifically tailored for generative models. This includes designing ad-

vanced detection techniques for identifying and mitigating backdoor triggers and jailbreaking prompts. Moreover, establishing frameworks for continuous monitoring and updating of generative models will be essential to adapt to new threats and ensure the ethical alignment of AI-generated content.

Security Risks from Single-Modal to Multi-Modal Models

While this thesis has primarily addressed security vulnerabilities within single-modal models, the advancement to multi-modal models, such as Sora and GPT-4, introduces a new array of security concerns that necessitate further research. Multi-modal models, which integrate and process data across various modalities (e.g., text, image, audio), are increasingly prevalent in cutting-edge applications, ranging from advanced conversational agents to comprehensive content generation systems.

One major challenge with multi-modal models is the expanded attack surface resulting from their integration of diverse data types. For instance, vulnerabilities within one modality can potentially compromise the entire system. Sora and GPT-4, which combine textual and visual inputs to generate multi-faceted outputs, are not immune to such risks. Adversarial attacks on the image component of these models could, for example, influence the text generation process, leading to potentially harmful or misleading outputs.

Additionally, these models are susceptible to novel attack vectors that exploit cross-modal interactions. For example, attacks could manipulate inputs in one modality—such as altering visual content in GPT-4's image-based tasks—to produce biased or discriminatory text outputs. The complexity of these interactions makes traditional security measures less effective, highlighting the need for advanced defense mechanisms tailored to multi-modal architectures.

Moreover, the risk of generating unsafe content is a significant concern. Multi-modal models like Sora and GPT-4 can produce content that is biased, discriminatory, or otherwise harmful due to the diverse nature of their data inputs and the complexity

of their learning processes. Ensuring that these models do not propagate illegal or unethical content requires comprehensive strategies to manage and mitigate risks across all modalities.

Future work should focus on developing robust security frameworks specifically for multi-modal models. This includes designing advanced algorithms to detect and defend against cross-modal attacks, improving the resilience of each modality to adversarial inputs, and establishing guidelines to prevent the generation of unsafe content.

References

[1] Michele Alberti, Vinaychandran Pondenkandath, Marcel Wursch, Manuel Bouillon, Mathias Seuret, Rolf Ingold, and Marcus Liwicki. Are you tampering with my data? In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[2] Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1505–1521. USENIX Association, August 2021.

[3] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020.

[4] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 101–105. IEEE, 2019.

[5] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000.

[6] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

[7] Ekaba Bisong and Ekaba Bisong. Logistic regression. *Building machine learning and deep learning models on google cloud platform: A comprehensive guide for beginners*, pages 243–250, 2019.

[8] Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. Bad Characters: Imperceptible NLP Attacks. In *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, 2022.

[9] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

[10] Ursin Brunner and Kurt Stockinger. Entity matching with transformer architectures - A step forward in data integration. In *Proceedings of the 23rd International Conference on Extending Database Technology, EDBT*, pages 463–473, 2020.

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[12] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, 2019.

[13] Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. In *ICLR*, 2022.

[14] Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. In *International Conference on Learning Representations*, 2022.

[15] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[16] Varun Chandrasekaran, Hengrui Jia, Anvith Thudi, Adelin Travers, Mohammad Yaghini, and Nicolas Papernot. Sok: Machine learning governance. *arXiv preprint arXiv:2109.10870*, 2021.

[17] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–10, 2023.

[18] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12299–12310, 2021.

[19] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *IJCAI*, volume 2, page 8, 2019.

[20] Kangjie Chen, Yuxian Meng, Xiaofei Sun, Shangwei Guo, Tianwei Zhang, Jiwei Li, and Chun Fan. Badpre: Task-agnostic backdoor attacks to pre-trained nlp foundation models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[21] Weixin Chen, Baoyuan Wu, and Haoqian Wang. Effective backdoor defense by exploiting sensitivity of poisoned samples. *Advances in Neural Information Processing Systems*, 35:9727–9737, 2022.

[22] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[23] Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32, 2019.

[24] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

[25] Valter Crescenzi, Andrea De Angelis, Donatella Firmani, Maurizio Mazzei, Paolo Merialdo, Federico Piai, and Divesh Srivastava. Alaska: A flexible benchmark for data integration tasks. *arXiv preprint arXiv:2101.11259*, 2021.

[26] Allan Dafoe. Ai governance: a research agenda. *Governance of AI Program, Future of Humanity Institute, University of Oxford: Oxford, UK*, 1442:1443, 2018.

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

[28] Chaitanya Desai, Deva Ramanan, and Charless C Fowlkes. Discriminative models for multi-class object layout. *International journal of computer vision*, 95:1–12, 2011.

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[30] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[31] Khoa D Doan, Yingjie Lao, Peng Yang, and Ping Li. Defending backdoor attacks on vision transformer via patch processing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 506–515, 2023.

[32] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[33] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[34] Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation and inference with energy based models. *ArXiv*, abs/2004.06030, 2020.

[35] Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020.

[36] Yilun Du, Shuang Li, Yash Sharma, Josh Tenenbaum, and Igor Mordatch. Unsupervised learning of compositional energy concepts. *Advances in Neural Information Processing Systems*, 34:15608–15620, 2021.

[37] Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. Automatic text summarization: A comprehensive survey. *Expert systems with applications*, 165:113679, 2021.

[38] Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. In *International Conference on Learning Representations*, 2020.

[39] Santosh K Gaikwad, Bharti W Gawali, and Pravin Yannawar. A review on speech recognition technique. *International Journal of Computer Applications*, 10(3):16–24, 2010.

[40] Boris Galitsky, Anton Chernyavskiy, and Dmitry Ilvovsky. Truth-o-meter: Handling multiple inconsistent sources repairing llm hallucinations. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2817–2821, 2024.

[41] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. *arXiv preprint arXiv:2303.07345*, 2023.

[42] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 265–273, 2008.

[43] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 113–125, 2019.

[44] Siddhant Garg, Adarsh Kumar, Vibhor Goel, and Yingyu Liang. Can adversarial weight perturbations inject neural backdoors. In *Proceedings of the ACM International Conference on Information & Knowledge Management (CIKM)*, 2020.

[45] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[46] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

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

[48] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[49] Tong Guo and Huilin Gao. Content enhanced bert-based text-to-sql generation. *arXiv preprint arXiv:1910.07179*, 2019.

[50] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

[51] Ali Hatamizadeh, Hongxu Yin, Holger R Roth, Wenqi Li, Jan Kautz, Daguang Xu, and Pavlo Molchanov. Gradvit: Gradient inversion of vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10021–10030, 2022.

[52] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: Defending against backdoor attacks using robust statistics. In *International Conference on Machine Learning*, pages 4129–4139. PMLR, 2021.

[53] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[55] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[56] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[57] Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[58] Julia Hirschberg and Christopher D Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015.

[59] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[60] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[61] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. In *ICLR*, 2022.

[62] Tim Hwang. Computational power and the social impact of artificial intelligence. *arXiv preprint arXiv:1803.08971*, 2018.

[63] Hengrui Jia, Mohammad Yaghini, Christopher A. Choquette-Choo, Natalie Dullerud, Anvith Thudi, Varun Chandrasekaran, and Nicolas Papernot. Proof-of-learning: Definitions and practice. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2021.

[64] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[65] Wei-Tsung Kao, Tsung-Han Wu, Po-Han Chi, Chun-Cheng Hsieh, and Hung-Yi Lee. Bert's output layer recognizes all hidden layers? some intriguing phenomena and a simple way to boost bert. *arXiv preprint arXiv:2001.09309*, 2020.

[66] Georgios Karagiannis, Mohammed Saeed, Paolo Papotti, and Immanuel Trummer. Scrutinizer: a mixed-initiative approach to large-scale, data-driven claim verification. In *Proceedings of the VLDB Endowment*, 2020.

[67] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[68] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[69] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer retraining is sufficient for robustness to spurious correlations. In *The Eleventh International Conference on Learning Representations*, 2023.

[70] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 301–310, 2020.

[71] Jernej Kos, Ian Fischer, and Dawn Song. Adversarial examples for generative models. In *2018 ieee security and privacy workshops (spw)*, pages 36–42. IEEE, 2018.

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

[73] Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806, 2020.

[74] Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. Backdoor attacks on pre-trained models by layerwise weight poisoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[75] Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. Hidden backdoors in human-centric language models. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2021.

[76] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2088–2105, 2020.

[77] Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, 2023.

[78] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34:14900–14912, 2021.

[79] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *ICLR*, 2021.

[80] Yiming Li, Haoxiang Zhong, Xingjun Ma, Yong Jiang, and Shu-Tao Xia. Few-shot backdoor attacks on visual object tracking. In *ICLR*, 2022.

[81] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16463–16472, 2021.

[82] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. Deep entity matching with pre-trained language models. *Proc. VLDB Endow.*, 14(1):50–60, 2020.

[83] Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. *arXiv preprint arXiv:2302.04578*, 2023.

[84] Xi Victoria Lin, Richard Socher, and Caiming Xiong. Bridging textual and tabular data for cross-domain text-to-sql semantic parsing. In *Findings of the Association for Computational Linguistics*, 2020.

[85] Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys (CSUR)*, 54(2):1–36, 2021.

[86] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pages 273–294. Springer, 2018.

[87] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *Com-*

puter Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII, pages 423–439. Springer, 2022.

[88] Qian Liu, Bei Chen, Jiaqi Guo, Zeqi Lin, and Jian-guang Lou. Tapex: table pre-training via learning a neural sql executor. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[89] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More control for free! image synthesis with semantic diffusion guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 289–299, 2023.

[90] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1265–1282, 2019.

[91] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

[92] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[93] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 182–199. Springer, 2020.

[94] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007.

[95] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022.

[96] Peizhuo Lv, Hualong Ma, Jiachen Zhou, Ruigang Liang, Kai Chen, Shengzhi Zhang, and Yunfei Yang. Dbia: Data-free backdoor attack against transformer networks. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 2819–2824. IEEE, 2023.

[97] Weimin Lyu, Songzhu Zheng, Teng Ma, and Chao Chen. A study of the attention abnormality in trojaned berts. *ArXiv*, abs/2205.08305, 2022.

[98] Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. The "beatrix" resurrections: Robust backdoor detection via gram matrices. *ArXiv*, abs/2209.11715, 2022.

[99] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[100] Graeme McLean and Kofi Osei-Frimpong. Hey alexa... examine the variables influencing the use of artificial intelligent in-home voice assistants. *Computers in Human Behavior*, 99:28–37, 2019.

[101] Saemi Moon, Seunghyuk Cho, and Dongwoo Kim. Feature unlearning for generative models via implicit feedback. *arXiv preprint arXiv:2303.05699*, 2023.

[102] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra.

Deep learning for entity matching: A design space exploration. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2018.

[103] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.

[104] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.

[105] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020.

[106] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021.

[107] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[108] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022.

[109] Jiaul H Paik. A novel tf-idf weighting scheme for effective ranking. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 343–352, 2013.

[110] Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. Hidden trigger backdoor attack on NLP models via linguistic style manipulation. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3611–3628, Boston, MA, August 2022. USENIX Association.

[111] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

[112] Ralph Peeters and Christian Bizer. Dual-objective fine-tuning of bert for entity matching. In *Proceedings of the VLDB Endowment*, 2021.

[113] Anna Primpeli, Ralph Peeters, and Christian Bizer. The wdc training dataset and gold standard for large-scale product matching. In *Proceedings of the World Wide Web Conference (WWW)*, 2019.

[114] Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Onion: A simple and effective defense against textual backdoor attacks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[115] Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[116] Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[117] Han Qiu, Yi Zeng, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Deepsweep: An evaluation framework for mitigating dnn back-

door attacks using data augmentation. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 363–377, 2021.

[118] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

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

[120] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[121] Javier Rando, Daniel Paleka, David Lindner, Lennard Heim, and Florian Tramèr. Red-teaming the stable diffusion safety filter. *arXiv preprint arXiv:2210.04610*, 2022.

[122] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016.

[123] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[124] Nader Sohrabi Safa, Rossouw Von Solms, and Steven Furnell. Information security policy compliance model in organizations. *computers & security*, 56:70–82, 2016.

[125] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11957–11965, 2020.

[126] Aniruddha Saha, Ajinkya Tejankar, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Backdoor attacks on self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13337–13346, 2022.

[127] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[128] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[129] Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. *arXiv preprint arXiv:2302.06588*, 2023.

[130] Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. Picard: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[131] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. *arXiv preprint arXiv:2211.05105*, 2022.

[132] Patrick Schramowski, Christopher Tauchmann, and Kristian Kersting. Can machines help us answering question 16 in datasheets, and in turn reflecting on

inappropriate content? In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1350–1361, 2022.

[133] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

[134] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[135] Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472, 2022.

[136] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.

[137] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

[138] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. *arXiv preprint arXiv:2302.04222*, 2023.

[139] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[140] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. *arXiv preprint arXiv:2212.03860*, 2022.

[141] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[142] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

[143] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 30, 2017.

[144] Akshayvarun Subramanya, Soroush Abbasi Koohpayegani, Aniruddha Saha, Ajinkya Tejankar, and Hamed Pirsiavash. A closer look at robustness of vision transformers to backdoor attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3874–3883, 2024.

[145] Akshayvarun Subramanya, Aniruddha Saha, Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Backdoor attacks on vision transformers. *arXiv preprint arXiv:2206.08477*, 2022.

[146] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[147] Araz Taeihagh. Governance of artificial intelligence. *Policy and society*, 40(2):137–157, 2021.

[148] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of {DNNs} for robust backdoor contamination detection. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1541–1558, 2021.

[149] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 2464–2469. IEEE, 2016.

[150] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31, 2018.

[151] Immanuel Trummer. Db-bert: a database tuning tool that" reads the manual". In *Proceedings of ACM SIGMOD/PODS Conference*, 2021.

[152] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

[153] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[154] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018(1):7068349, 2018.

[155] 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 *International Conference on Learning Representations*, 2019.

[156] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019.

[157] Lihan Wang, Bowen Qin, Binyuan Hui, Bowen Li, Min Yang, Bailin Wang, Binhua Li, Fei Huang, Luo Si, and Yongbin Li. Proton: Probing schema linking information from pre-trained language models for text-to-sql parsing. *arXiv preprint arXiv:2206.14017*, 2022.

[158] Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv preprint arXiv:2210.14896*, 2022.

[159] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[160] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[161] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34:16913–16925, 2021.

[162] Xiaoxue Wu, Wei Zheng, Xin Xia, and David Lo. Data quality matters: A case study on data label correctness for security bug report prediction. *IEEE Transactions on Software Engineering*, 48(7):2541–2556, 2021.

[163] Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang,

et al. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *arXiv preprint arXiv:2201.05966*, 2022.

[164] 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*, 5(12):1486–1496, 2023.

[165] Ashima Yadav and Dinesh Kumar Vishwakarma. Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, 53(6):4335–4385, 2020.

[166] Inbal Yahav, Onn Shehory, and David Schwartz. Comments mining with tf-idf: the inherent bias and its removal. *IEEE Transactions on Knowledge and Data Engineering*, 31(3):437–450, 2018.

[167] Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. Generative poisoning attack method against neural networks. *arXiv preprint arXiv:1703.01340*, 2017.

[168] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022.

[169] Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in nlp models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

[170] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing

and text-to-sql task. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[171] Zenghui Yuan, Pan Zhou, Kai Zou, and Yu Cheng. You are catching my attention: Are vision transformers bad learners under backdoor attacks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24605–24615, 2023.

[172] Xinyang Zhang, Zheng Zhang, Shouling Ji, and Ting Wang. Trojaning language models for fun and profit. In *Proceedings of IEEE European Symposium on Security and Privacy (EuroS&P)*, 2021.

[173] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10146–10156, 2023.

[174] Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. Backdoor attacks to graph neural networks. In *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, pages 15–26, 2021.

[175] Zaixi Zhang, Qi Liu, Zhicai Wang, Zepu Lu, and Qingyong Hu. Backdoor defense via deconfounded representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12228–12238, 2023.

[176] Zhiyuan Zhang, Lingjuan Lyu, Xingjun Ma, Chenguang Wang, and Xu Sun. Fine-mixing: Mitigating backdoors in fine-tuned language models. *arXiv preprint arXiv:2210.09545*, 2022.

[177] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models.

In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14443–14452, 2020.

[178] Mengxin Zheng, Qian Lou, and Lei Jiang. Trojvit: Trojan insertion in vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4025–4034, 2023.

[179] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.

[180] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*, 2017.

[181] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8739–8748, 2018.

[182] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

[183] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.

[184] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.