Mutual Information as a Unified Lens for Deep Learning Security

Yutong Wu

College of Computing and Data Science

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2026

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

Disclosure on AI Use:

I hereby declare that GenAI (or any AI-assisted tool) was not used in the preparation of this thesis.

06/01/2026

Date

Yutong Wu

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accordance with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

06/01/2026

Date

Assoc Prof Tianwei Zhang

Authorship Attribution Statement

This thesis contains materials from 3 papers published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as Yutong Wu, Xingshuo Han, Han Qiu, Tianwei Zhang. "Computation and data efficient backdoor attacks." In Proceedings of the IEEE/CVF international conference on computer vision (ICCV), 2023.

The contributions of the co-authors are as follows:

- I was the lead author. I wrote the manuscript draft and conducted all the experiments.
- Prof. Tianwei Zhang provided guidance on the initial research direction and contributed to revising the manuscript.
- I co-designed the methodology with Prof. Tianwei Zhang, and Prof Xingshuo Han.
- Prof Han Qiu discussed and supported the research, and revised the draft.

Chapter 4 is published as Yutong Wu, Han Qiu, Shangwei Guo, Jiwei Li, and Tianwei Zhang. "You only query once: An efficient label-only membership inference attack." In The Twelfth International Conference on Learning Representations (ICLR). 2024.

The contributions of the co-authors are as follows:

- I was the lead author. I wrote the manuscript draft and conducted all experiments.
- Prof. Tianwei Zhang provided guidance on the initial research direction and contributed to revising the manuscript.
- I co-designed the methodology with Prof. Tianwei Zhang.
- Prof Han Qiu, Prof. Shangwei Guo, Prof Jiwei Li discussed and supported the research, and revised the draft.

Chapter 5 is published as Yutong Wu, Jie Zhang, Florian Kerschbaum, and Tianwei Zhang."THEMIS: Regulating Textual Inversion for Personalized Concept Censorship." In the Network and Distributed System Security (NDSS) Symposium. 2025.

The contributions of the co-authors are as follows:

- I was the lead author. I wrote the manuscript draft and conducted all experiments.

- Prof. Tianwei Zhang provided guidance on the initial research direction and contributed to revising the manuscript.
- I co-designed the methodology with Prof. Tianwei Zhang and Dr. Jie Zhang.
- Prof. Florian Kerschbaum discussed and supported the research, and revised the draft.

06/01/2026

Date

Yutong Wu

Acknowledgements

Standing at the closing threshold of this long and transformative journey, I am filled with a profound sense of gratitude. The years at Nanyang Technological University have been shaped by generous teachers, thoughtful peers, and the quiet endurance of those who believed in me even when I doubted myself. Singapore, vibrant yet composed, became more than a place to study—it became the landscape of my becoming, the city that held my questions, my failures, and my small, hard-earned victories.

First and foremost, I wish to express my deepest gratitude to my advisor, Prof. Tianwei Zhang, whose steady guidance and generous belief in my potential have shaped every stage of this journey. His clarity of thought, rigor in scholarship, and unwavering commitment to intellectual integrity have not only directed the course of my research but also broadened the way I understand inquiry itself. Under his mentorship, I learned not merely how to solve a problem, but how to recognize a meaningful one, and most importantly, how to remain patient in its pursuit. Prof. Zhang's encouragement, offered quietly yet consistently, instilled in me the courage to step beyond familiar boundaries and engage new questions with conviction. His example, which is firm in principle, meticulous in method, and gracious in demeanor, has been a compass both in academic work and in life. For the steadiness he lent in moments of uncertainty, for the wisdom he offered when ambition needed balance, and for the confidence he nurtured long before I could claim it myself, I remain profoundly thankful. And, I count myself genuinely fortunate to have studied under his guidance, and to have learned not only from his scholarship but from his character.

I would also like to extend my sincere thanks to Prof. Han Qiu, Prof. Florian, Prof. Yang Liu, Prof. Xiaofei Xie, Prof. Shangwei Guo, Prof. Jiwei Li, and Dr. Guowen Xu, whose guidance and insight have accompanied me through the more intricate passages of this research journey. Their willingness to share knowledge

with clarity and generosity, and to offer feedback that was both constructive and thoughtful, allowed my work to unfold with greater depth and assurance. Each conversation, critique, and gentle correction served as a quiet but steady current, shaping not only the direction of my studies but also my understanding of what it means to think rigorously and collaboratively. For their patience, their precision, and their enduring support, I remain sincerely grateful.

I am profoundly grateful to my colleagues and friends, whose friendship, encouragement, and shared moments of discovery made this journey both meaningful and memorable, including Kexi Yan, Yirui Luo, Chenxing Liu, Ozymandias Zhang, Xiaobei Yan, Haoran Ou, Shiqian Zhao, Zhaoxuan Wang, Yi Xie, Tianlin Li, Yanzhou Li, Yue Cao, Wenjun Long, Boheng Li, Dr. Meng Hao, Dr. Hanxiao Chen, Dr. Guanlin Li, Dr. Kangjie Chen, Dr. Xingshuo Han, Dr. Xiaoxuan Lou, Dr. Gelei Deng, Dr. Yuan Xu, Dr. Haozhao Wang, Dr. Wenhao Fu, Dr. Hao Ren, Dr. Jianfei Sun, Dr. Hangcheng Liu, Dr. Jie Zhang, Dr. Yiming Li, Dr. Yuan Zhou, Dr. Chengwei Liu, Dr. Anran Li, Dr. Xiuheng Wu, Dr. Cen Zhang, Dr. Ming Hu, Dr. Jian Zhang, Dr. Yihao Huang, Dr. Wenhan Wang and Dr. Renyang Liu. The countless hours we spent together in the lab, engaging in thoughtful discussions and working through challenging projects, were both inspiring and intellectually enriching. Their support, collaboration, and camaraderie made this journey not only highly productive, but also deeply enjoyable.

Last but certainly not least, I owe my deepest gratitude to my parents and family, whose unwavering love, support, and encouragement have sustained me throughout this journey. Their faith in me has been a constant source of strength, and I dedicate this achievement to them.

To everyone mentioned above, and to those whose names I may have unintentionally overlooked, I extend my heartfelt thanks. Your support has been integral to my personal and academic growth, and I am sincerely grateful for the kindness, encouragement, and care that accompanied me along the way.

Yutong Wu, Dec 2025

Abstract

Modern deep neural networks can be regarded as high-dimensional information channels whose parameters encode statistical dependencies between inputs and outputs. Rather than serving solely as function approximators, these models accumulate and propagate information during training, and such information can later be activated, extracted, or redirected during inference. From this perspective, mutual information offers a principled quantity for characterizing model behavior: an adversary succeeds whenever a controllable variable (e.g., carefully-crafted malicious samples) induces a measurable dependency with the model's output distribution, thereby forming an unintended information pathway.

This information-theoretic perspective offers several advantages over conventional heuristics. First, mutual information provides a unified analytical language for modeling diverse threats. Whether the adversary compromises the training procedure, retrieves private information from the model, or misuses the model to generate harmful content, each attack can be interpreted as an attempt to increase the mutual information between a controlled variable and the model's output. These attacks thus become not isolated pathologies, but instances of the same phenomenon: adversarial construction or exploitation of unintended information pathways. Second, mutual information directly suggests efficient attack and defense strategies. Attacks need not poison entire datasets or use extensive query budgets, but only the components that maximize information transfer matter.

In this thesis, we demonstrate that mutual information can be used as a unified tool to examine, analyze, and improve distinct security threats of deep learning models, focused on three prominent security problems: backdoor attack, membership inference attack, and harmful content generation. In the first part of this thesis, we formulate the backdoor attack as an engineered information channel connecting a trigger to a target output. Based on our analysis, we introduce Representational Distance (RD) to identify the small subset of training samples that carries the

greatest influence on this channel. Poisoning only high-RD samples enables effective attacks with orders-of-magnitude less data and computation, while achieving accuracy comparable to prior large-scale poisoning strategies.

In the second part of this thesis, we show that training data leave persistent traces in model outputs, which results in an accessible information channel that could reveal the privacy clues. We propose YOQO, the first label-only membership-inference attack requiring a single query. By identifying an "improvement area" around the target sample and eliciting a hard-label response, YOQO infers membership without confidence scores, or repeated queries, and remains effective under multiple defense mechanisms.

In the third part of this thesis, we show that generative models can be protected by suppressing the information pathways that connect malicious prompts to harmful outputs. THEMIS achieves this by training a pseudo-word embedding that minimizes mutual information between sensitive prompts and generated images in text-to-image personalization. When adversaries attempt to combine sensitive words with the protected concept, the model outputs a benign target image, while normal usage remains unaffected.

Together, these contributions show that the security and privacy of modern AI depend not only on robustness or filtering, but on the regulation of information flow within high-capacity models. Mutual information provides a quantitative lens through which attacks and threats can be understood, detected, and controlled.

Contents

List of Publications

- Yutong Wu, Jie Zhang, Florian Kerschbaum, Tianwei Zhang. **THEMIS: Regulating Textual Inversion for Personalized Concept Censorship**. In *Network and Distributed System Security Symposium (NDSS), 2025.*

- Yutong Wu, Xingshuo Han, Han Qiu, Tianwei Zhang. **Computation and Data Efficient Backdoor Attacks**. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023.*

- Yutong Wu, Han Qiu, Shangwei Guo, Jiwei Li, Tianwei Zhang. **You Only Query Once: An Efficient Label-only Membership Inference Attack**. In *International Conference on Learning Representations (ICLR), 2024.*

- Yutong Wu, Jie Zhang, Yiming Li, Chao Zhang, Qing Guo, Nils Lukas, Tianwei Zhang. **Cowpox: Towards the Immunity of VLM-based Multi-Agent Systems**. *International Conference on Machine Learning (ICML), 2025.*

- Yutong Wu, Han Qiu, Tianwei Zhang, Jiwei Li, Meikang Qiu. **Watermarking Pre-trained Encoders in Contrastive Learning**. In *2022 4th International Conference on Data Intelligence and Security (ICDIS), 2022.*

- Xingshuo Han, Yutong Wu, Qingjie Zhang, Yuan Zhou, Yuan Xu, Han Qiu, Guowen Xu, Tianwei Zhang. **Backdooring Multimodal Learning**. In *IEEE Symposium on Security and Privacy (S&P), 2024.*

- Yihao Huang, Felix Juefei-Xu, Qing Guo, Jie Zhang, Yutong Wu, Ming Hu, Tianlin Li, Geguang Pu, Yang Liu. **Personalization as a Shortcut for Few-shot Backdoor Attack Against Text-to-Image Diffusion Models**. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2024.*

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Security Threats in Deep Learning Systems

Over the past decade, deep learning has emerged as a transformative paradigm in artificial intelligence, driving data-driven solutions across diverse domains. Compared with traditional rule-based or feature-engineered approaches, it has exhibited remarkable performance in tasks like medical diagnosis [1], stock price prediction [2], facial recognition [3], and object detection [4]. Meanwhile, the scope of AI has rapidly expanded into newer, more generative tasks, including text-to-image synthesis [5–7], large-scale language modeling [8] , and personalized content generation [9, 10], where models are not only classifiers, but interactive systems that absorb and reflect vast amounts of information from training data.

As these systems grow in capability, so do their potential risks. Despite their success, deep learning models remain susceptible to a broad range of security, privacy, and safety threats. For example:

1) **Backdoor attack**: By embedding imperceptible or semantically inconspicuous trigger patterns during training, adversaries can inplanting stealthy backdoors into deep learning models, thereby inducing the model to learn a strong association between these patterns and attacker-specified target outputs. [11–17]. Critically, the backdoored model maintains high accuracy on benign inputs, thereby evading detection through standard validation procedures. However, at inference time, the presence of the trigger activates the malicious behavior, causing the model to

misclassify or otherwise deviate from its intended function. By leveraging such backdoors, adversaries can reliably manipulate model predictions under controlled conditions, posing a substantial threat to the trustworthiness of deep learning systems deployed in sensitive or high-stakes environments.

2) **Membership inference attack (MIA)**: Prior studies [18–22] have adequately provided the evidences indicating that deep learning models possess a significant capability to memorize their training data. Such memorization facilitates membership inference attacks (MIAs), enabling adversaries to infer the membership of individual samples, which can lead to privacy leakage by allowing adversaries to infer information about the underlying training data distribution. The impact of MIAs becomes more severe when models are trained or fine-tuned on datasets containing sensitive records or attributes—such as medical data, biometric information, or personal histories—where the inclusion of even a single data point may already constitute a privacy violation.

3) **Malicious content generation**: Existing T2I models are trained on datasets containing thousands or even millions of caption-image pairs that have not been carefully sanitized, e.g., SD models trained on LAION-5B [23]. These models are capable of generating unsafe content if given malicious prompts containing sensitive words. To add insult to injury, TI further provides malicious users with a powerful tool for defaming, usurping, and stigmatizing, in terms of a personalized concept. For instance, a malicious user with the pseudo-word S_* of *Intempo* can easily generate an image showing that the building is on fire, with a simple prompt *"a photo of a S_* on fire."*

Although prior research has examined these vulnerabilities independently, the absence of a unified theoretical framework leads to fragmented reasoning and task-specific solutions. This fragmentation obscures the fundamental mechanism shared across these threats. In this thesis, we adopt mutual information as a model-agnostic measure of statistical dependency, which provides the missing unifying perspective for various threats.

1.2 Motivation for Using Mutual Information

While these attacks differ in mechanisms and target modalities, they share a deeper commonality: they all exploit the information-carrying capacity of deep models. Backdoor triggers create hidden dependencies between an input pattern and an output label [24, 25]. Membership inference relies on latent memorization of training examples that leak through model confidence [26, 27]. Malicious prompt attacks manipulate high-dimensional conditioning signals to elicit targeted generation by activating latent information paths embedded during training. This shared dependence on hidden statistical dependencies offers a unique opportunity: the diverse threats can be analyzed through a unified information-theoretic lens using mutual information. In each case, the goal of the attacker is to manipulate or extract information from the model, indicating that these vulnerabilities stem from a shared root: *the uncontrolled formation of unintended information channels inside the model.*

To rigorously analyze such threats within this unified perspective, it is essential to adopt a metric that quantifies the statistical dependence between variables, irrespective of model architectures, task types, or accessibility constraints. Mutual information [28] fulfills this role by offering a principled measure of shared information content. Formally, the mutual information between two random variables A and B is defined as:

$$I(A;B) \triangleq D_{KL}(p(a,b)||p(a)p(b)) = \iint p(a,b)\log\frac{p(a,b)}{p(a)p(b)}dadb \tag{1.1}$$

Here, $p(a,b)$ is the joint distribution, D_{KL} is the Kullback–Leibler divergence. It can be written in an equivalent form in terms of entropy difference:

$$
\begin{aligned}
I(A;B) &= \iint p(a,b)\log\frac{p(a,b)}{p(a)p(b)}dadb \\
&= \iint p(a,b)\log\frac{p(a|b)p(b)}{p(a)p(b)}dadb \\
&= \iint p(a,b)\log\frac{p(a|b)}{p(a)}dadb \\
&= \iint p(a,b)\log p(a|b)dadb - \iint p(a,b)\log p(a)dadb \\
&= \underbrace{\iint p(a,b)\log p(a|b)dadb}_{-H(A|B)} \underbrace{- \int p(a)\log p(a)da}_{H(A)} \\
&= H(A) - H(A|B).
\end{aligned}
\tag{1.2}
$$

Intuitively, $H(A)$ measures the uncertainty of the random variable A, whereas $H(A|B)$ quantifies the remaining uncertainty of A after observing B. The mutual information, consequently, can be interpreted as the reduction in the uncertainty of A due to the knowledge of B. Within the context of deep learning, it enables precise estimation of how much information pertaining to specific elements, i.e., training data, injected triggers, or external prompts, is embedded within or can be inferred from the model's outputs. For example, in backdoor attacks, the attack goal is to establish the association between an arbitrary model output y^t and the given trigger t [11–17], which is equivalent to maximizing $I(y^t;t)$ [24, 25]; In some membership inference works [18–22], attackers aim at finding proper features o from the model outputs so that $I(o;m)$ is as large as possible, where $m \sim \text{Bernoulli}(p)$ is the membership indicator of a given sample.

In this thesis, we adopt a unified analytical perspective to bridge three distinct security challenges in deep learning systems and develop state-of-the-art methodologies based on the mutual information theory. Adopting such a unified analytical perspective offers several key advantages. First, it enables consistent reasoning across diverse threat modalities, eliminating the need to devise task-specific heuristics or ad hoc tools for each new vulnerability. Second, this perspective facilitates efficient algorithmic design, since mutual information can often be estimated or optimized in a tractable manner: enabling practical deployment of both attacks and defenses in real-world systems. Finally, integrating diverse attacks under a common theoretical framework facilitates the identification of underlying commonalities among

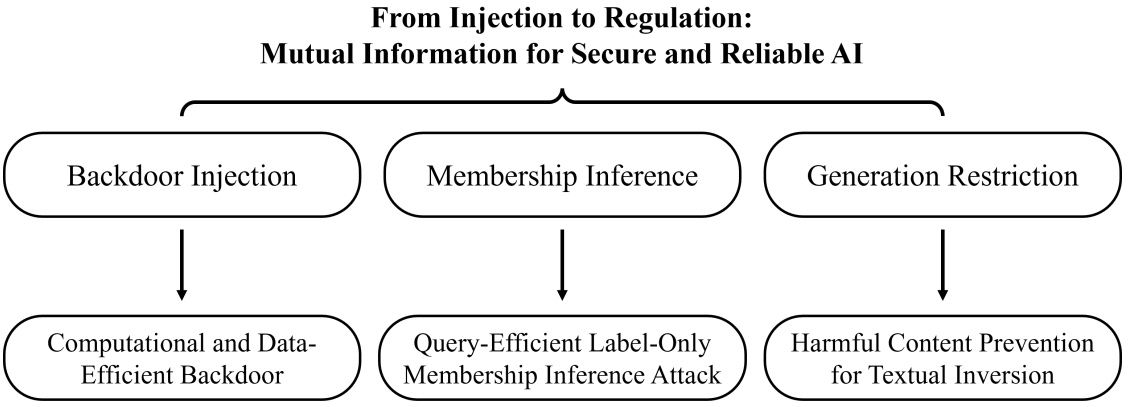


FIGURE 1.1: The main works of this thesis.

vulnerabilities, which in turn facilitates the design of more unified and systematic approaches to security enhancement.

1.3 Main Works

As shown in Figure 1.1, the main works included in the thesis are as follows:

MI for Efficient Backdoor Injection. In Chapter 3, we develop a novel confidence-driven scoring mechanism that quantifies the contribution of each poisoning sample by leveraging distance-based posteriors and maximizing the early-stage gradient of the mutual information between the trigger and the target output. Building on this formulation, we further introduce a greedy search procedure that expedites the identification of the most informative poisoning samples for backdoor injection. Comprehensive experiments show that our method attains performance exceeding all existing strategies, while requiring significantly less computational resources and samples to be poisoned, indicating that the real-world poisoning-based attack can be much more stealthy than it is known to be.

MI for Label-Only MIA. In Chapter 4, we propose YOQO, a new approach for label-only MIA that reduces the required query budget from over one thousand queries to one. The key idea is to identify a specially structured sample that resides within an "improvement region," defined as the area in which the mutual information discrepancy between the target model (in-model) and a surrogate or reference model (out-model) is maximized. By exploiting this mutual information

gap, YOQO can reliably infer membership using only one query. Empirical evaluations indicate that YOQO attains performance on par with, or superior to, current state-of-the-art attack methods. while exhibiting substantially higher robustness against a wide range of advanced defense mechanisms. This highlights the fact that only a minimal amount of information is sufficient to expose private data.

MI for Generation Regulation. In Chapter 5, we present Themis, a framework designed to enable personalized concept censorship for Textual Inversion (TI). The key insight lies in suppressing the mutual information linking malicious images to their associated trigger semantics, while preserving the semantics of all benign prompt components. Concretely, during TI training, the concept creator designates a set of sensitive terms as triggers, which will subsequently be censored during normal model usage. At inference time, if a malicious user combines these sensitive terms with the personalized embeddings as part of the prompt, the text-to-image (T2I) model is steered to generate a predefined benign target image instead of producing content aligned with the malicious intent. This research sheds light on a practical pathway toward safer personalization in generative models by enabling fine-grained, creator-controlled censorship of harmful concept interactions without degrading model usability.

1.4 Contribution of the Thesis

1. **Computation and Data Efficient Backdoor Attack**. This work introduces a novel data-efficient backdoor attack method. By using a mutual-information-based score, this enables the attacker to pinpoint samples that play a more dominant role in inducing the backdoor behavior, achieving a comparable attack success rate by poisoning only a very small number of data points in the training set. This attack demonstrates that mutual-information theory provides a principled basis for constructing more efficient poison-based backdoor attacks, and also indicates that backdoor attacks can be conducted in a more imperceptible way.

2. **Query-Efficient Label-Only MIA**. In this work, we propose a novel label-only membership inference attack that reduces the query budget to a single model query, thereby making label-only MIAs substantially more practical and easier to execute in real-world scenarios. We further demonstrate that samples can be effectively

characterized by a distinct region induced by our mutual-information–based loss function. This finding reveals that membership privacy leakage can occur with far less information than previously assumed, highlighting a more fragile boundary between training and non-training examples. Our results underscore the urgent need for more robust defenses to mitigate such underestimated privacy risks.

3. **Themis: Regulating Textual Inversion for Personalized Concept Censorship**. This work presents the first regulatory framework for Textual Inversion embeddings, leveraging a mutual-information-stemmed loss to constrain the interaction between personalized concepts and sensitive triggers. This mechanism effectively prevents malicious actors from exploiting user-defined embeddings to generate harmful or offensive content. By enforcing controlled concept usage while preserving legitimate functionality, the proposed framework advances the development of a cleaner, more accountable, and more responsible AIGC ecosystem.

Overall, this thesis advances deep learning security by showing that diverse threats, including backdoors, membership inference, and harmful generation, can be effectively characterized and mitigated within a single mutual-information–driven analytical framework. It uncovers the potential association among the seemingly diverse vulnerabilities of the deep learning system.

1.5 Roadmap

This thesis is organized as follows:

Chapter 1 provides an overview of this thesis, as well as the motivations, main work, and contributions of each work.

Chapter 2 reviews the related works of backdoor attacks, MIA, and generation content regulation.

Chapter 3 discusses the design of the computational and data-efficient backdoor attack, which requires fewer samples to be poisoned in the training dataset to construct the backdoor.

Chapter 4 introduces the design of YOQO, a query-efficient label-only membership inference attack.

Chapter 5 presents Themis, the first framework to regularize textual inversion embedding for text-to-image generation tasks.

Chapter 6 concludes the above research works and give a discussion about possible future directions.

Chapter 2

Related Works

The rapid evolution of deep learning has been paralleled by growing concerns surrounding security and intellectual property protection. This section reviews the most relevant literature in three closely related areas: backdoor attacks, which compromise model integrity by injecting malicious behaviors, membership inference attacks, which violate the privacy of the training sample; and generated content moderation, which aims to improve the safety of the AIGC community.

2.1 Backdoor Attacks

2.1.1 Attack Formulation

Backdoor attacks on DNNs can be performed via different methods, including poisoning training data [29], modifying off-the-shelf model's structures [30], flipping bits at the deployment stage [31], etc. Among these methods, poisoning training data is the most widely used way, which is also our focus in this work. Formally, consider a model f and its training dataset $\mathcal{D} = \{(x, y)\}$ for supervised training, where y is the label of the data x. The adversary samples a small proportion of $\mathcal{X}$ to poison[1]. He injects a carefully-designed trigger (m, t) to the input x, where m

[1]Here we describe the basic poisoning approaches. There are also advanced strategies, e.g., clean-label attacks [32] and clean-image attacks [33], which are beyond the scope of this work.

is a mask and t is a trigger. The poisoned input x' follows:

$$x' = x \oplus t = x \odot (1 - m) + t \odot m, \tag{2.1}$$

where $\odot$ is the element-wise multiplication. The adversary also changes the original label y_i to a predetermined target label y_t. The model trained on this poisoned dataset will get injected with the backdoor. During the inference, the infected model will give normal prediction results on clean data, while predicting the target label y_t on the malicious data with the trigger manipulated by the backdoor function $\mathcal{B}$ with high probability. Researchers have designed backdoor attacks against different domains and tasks, e.g., image classification [29], 3D point cloud classification [34, 35]. Recent backdoor approaches focus on the stealthiness requirement with visually or semantically hidden triggers [36, 37].

2.1.2 Backdoor Defenses

Researchers proposed several ways to mitigate backdoor attacks. They can be roughly categorized into model reconstruction, model inspection, and runtime input filtering.

Model reconstruction. A common line of work directly modifies a suspicious model to erase its backdoor while preserving its utility on clean data. Fine-pruning [38] first prunes neurons with low activations on a held-out clean validation set and then fine-tunes the remaining weights. The intuition is that dormant neurons are more likely to participate in the backdoor behavior than in normal predictions. By carefully choosing the pruning rate and fine-tuning epochs, Fine-pruning can significantly reduce the attack success rate (sometimes to nearly 0%) with only marginal drops in clean accuracy.

Model inspection and trigger reverse-engineering. Another strategy is to inspect the model to detect and localize potential backdoors. For example, Neural Cleanse [39] assumes that backdoor triggers are small and sparse, and formulates reverse-engineering triggers as an optimization problem that searches, for each target label, a minimal perturbation mask and pattern that can reliably flip predictions to that label. It then applies a Median Absolute Deviation (MAD) test to

the norms of the recovered triggers to flag anomalous labels as potentially backdoored and uses the synthesized triggers to fine-tune the model and remove the backdoor behavior.

Runtime input filtering. Instead of modifying the model, some defenses detect and reject malicious inputs at inference time. STRIP [40] builds a run-time detection mechanism by leveraging the input-agnostic nature of typical backdoor triggers. It intentionally perturbs each incoming input (e.g., by superimposing random image patterns) and measures the entropy of the model's predictions over these perturbed copies: for benign inputs, predictions vary significantly, while for trojaned inputs, predictions remain consistently biased toward the target label, exhibiting low entropy. A predefined entropy threshold thus allows STRIP to filter out suspicious queries with very low false rejection and false acceptance rates across several datasets and trigger types.

2.2 Query Efficient Membership Inference Attacks

2.2.1 Attack Formulations

MIA is first proposed in [22] and has drawn great attention since it reflects the privacy-preserving ability of machine learning models which are usually trained over sensitive data. In this attack, the adversary tries to infer whether a target sample belongs to the training set $\mathcal{D}$ of a victim model $f_{\mathcal{D}}$ or not. He achieves this binary classification problem by building an algorithm $\mathcal{A}$ for an arbitrary sample x:

$$\mathcal{A}(f_{\mathcal{D}}, x) = \begin{cases} 0 \to x \in \mathcal{D} \\ 1 \to x \notin \mathcal{D} \end{cases}$$

There are mainly two categories of approaches to conducting MIAs based on the adversary's knowledge.

Posterior-based MIA. Numerous works [21, 22, 41–48] exploit posteriors to infer the membership of the target sample. The adversary is ignorant of the neural network architecture and parameters of the victim model. He can make queries

to the model and obtain its posteriors, e.g., prediction scores or output logits. There are two strategies to build the membership inference algorithm $\mathcal{A}$. The first strategy is to implement a DNN-based binary classifier as $\mathcal{A}$ [22, 42, 44, 48]. The adversary first prepares a set of shadow datasets $\mathcal{D}_s$, which follow the same distribution as the victim model's training set $\mathcal{D}$. Some of the shadow datasets contain the target sample x while others do not. Then he trains a surrogate model f_s on each shadow dataset. Based on these surrogate models, he can construct a membership dataset:

$$\mathcal{D}_{mem} = \{(f_s(w), \mathbb{I}(w \in \mathcal{D}_s))\} \tag{2.2}$$

where $\mathbb{I}(X)$ is the indicator function, which equals 1 if X is true and 0 otherwise. The adversary then trains a binary classifier g on $\mathcal{D}_{mem}$. In the attack phase, he queries the victim model with the target sample x and retrieves the outputs, which are subsequently fed to g to infer the membership of x. Here we have $\mathcal{A}(f_{\mathcal{D}}, x) = g(f_{\mathcal{D}}(x))$.

The second strategy relies on statistic-based approaches *e.g.*, Bayesian optimization and KL-divergence, to learn a threshold τ to divide the manifold into two classes [21, 41, 43, 45–47]. To conduct MIA, the adversary also trains surrogate models on the subsets of a shadow dataset. He then uses the training loss or some specially designed loss functions F to gain a score for each output given by f_θ. The threshold τ is subsequently learned by studying the correlation between the memberships and the scores. Thereby, for these works, $\mathcal{A}(f_\theta, x) = \mathbb{I}(F(f_\theta(x)) > \tau)$.

Label-only MIA. The above attacks all require the adversary to have the posteriors of the query sample, which may not be possible in some MLaaS platforms [49]. To overcome this limitation, researchers proposed label-only attacks, where the adversary only has access to the hard labels from the victim model. To the best of our knowledge, there are mainly three state-of-the-art label-only attacks.

Gap attack [47] leverages the overfitting phenomenon to infer the membership. As the victim model is usually trained to achieve higher accuracy on the training set, the adversary simply takes samples that are misclassified by the victim model as non-members and the rest as members.

Boundary attack [18, 19] follows the intuition that non-member samples are closer to the decision boundary of the victim model. To estimate the distance of the target sample from the boundary, the adversary applies decision-based adversarial attack

algorithms like Hopskipjump [50] and QEBA [51] to the victim model. He generates an adversarial example from the target sample, and calculates the distance between them, which will be used as the estimation of the distance between the target sample and its nearest decision boundary. He then trains a bunch of surrogate models and casts the attack on them to learn a threshold τ for the victim model to divide the member and non-member samples.

Data augmentation attack [18] exploits the observation that machine learning models are likely to overfit augmented samples. The adversary creates a binary classifier $h(x, f_{\mathcal{D}})$ for the victim model $f_{\mathcal{D}}$, which outputs $h(x, f_{\mathcal{D}}) = 1$ if x belongs to $\mathcal{D}$. To build a training set for h, the adversary first creates additional data points using the target sample x with different augmentation strategies. Then he exploits all these samples to query a shadow model $\hat{h}$ following the knowledge of the target model's architecture and training data distribution. After obtaining the labels $(l_0, ..., l_n)$, he compares them to the ground truth l_{true} and uses $b_i = \mathbb{I}(l_{true} = (l_i))$ as the samples augmented by the same strategies and feeds the labels in a similar form as the training data to h. This approach requires the strong assumption that the adversary knows the augmentation strategies exploited during the victim model's training process. It becomes less effective when this assumption is relaxed.

2.2.2 Defenses Against MIAs

Researchers proposed several ways to mitigate MIAs. They can be summarized into four categories according to the strategies they take advantage of.

Fine-tuning the model. Researchers proposed to fine-tune the model to reduce the membership inference leakage. Specifically, Adversarial Regularization (ADV) [52] takes the membership inference adversary into consideration during the training process. The defender first trains an attack model to infer the membership. Then he fine-tunes the target model by minimizing the loss on the training set while maximizing the classification loss of the attack model. There is a hyperparameter β to balance the influence of the two terms in the optimization process. PPB [48] makes use of the divergence of confidence and prediction sensitivity between members and non-members to build the defense. This strategy tries to narrow the prediction gaps to make the outputs of members and non-members indistinguishable. It fine-tunes the target model by minimizing both the KL divergence

between the ranked output posterior and classification loss on the training set. A hyper-parameter λ controls the weights of the two terms, i.e., the KL divergence and prediction loss.

Perturbing the inputs. The strategy perturbs the inputs before they are fed to the victim model. LDL [53] constructs a hypersphere around the target sample to make the decisions of the model unchanged for all samples in the sphere. It can defeat label-only attacks that estimate the distance of the target sample from the decision boundary (e.g., the boundary attack). The hyper-sphere is constructed by adding noise $n_i \sim \mathcal{N}(0, \sigma)$ to the input sample x to get plenty $x_{noisy} \leftarrow x + n_i$. Then x_{noisy} is fed into the victim model. All the outputs are merged by averaging them to get the final posterior for the model to make the decision.

Modifying the posteriors. As many MIAs use the posteriors from the victim model, this strategy tries to mitigate the attacks by minimizing the privacy leakage in the prediction confidence without changing the prediction labels. MemGuard [54] injects adversarial perturbations to the confidence to mislead the adversary's membership classifier. One-hot encoding [55] encodes the prediction confidence into a one-hot code, making it almost impossible to achieve MIAs just based on the posteriors.

Differential privacy [56, 57]. DP-SGD is proven to be effective in preventing privacy leakage [57–59]. It modifies the gradient in the training process by clipping and then adding Gaussian noise to it. As the noise follows the distribution $\mathcal{N}(0, \sigma)$, the standard deviation δ should be in the order of $\Omega(q\sqrt{T \log(1/\delta)/\epsilon})$.

2.3 Regulating Content Generation

2.3.1 Text-to-Image Models and Textual Inversion

Text-to-Image is a well-studied task to control the generative model by textual-based prompts such as natural language [60–65]. Existing solutions can be summarized roughly into four categories, i.e., GAN-based [62, 66, 67], auto regression-based [63], mask prediction [68], and diffusion model-based [5–7, 65, 69]. Among

them, the diffusion model-based solution has recently surpassed the other approaches to achieve state-of-the-art performance in terms of both generation quality and diversity. For instance, Glide [69] exploits the ADM model [70] as the backbone model. The prompts are firstly turned into embeddings by a clip textual transformer, which is subsequently projected into the same dimension as the attention vectors and concatenated with them. Stable Diffusion [71] introduces a cross-attention mechanism into the down-sampling and up-sampling model, respectively, to control the image generation process. Specifically, the generation architecture is composed of three parts: the text encoder c_θ, the image encoder/decoder, which are often VAE models, and the diffusion model. The VAE models are usually used for just upscaling and downscaling the images to reduce the complexity of the diffusion process. Imagen [7] further performance improvements on Stable Diffusion by using a more powerful textual encoder and setting thresholds during the sampling process of the model. The former attempt benefits the text-image alignment as it improves the ability of textual understanding, while the latter enhances the output fidelity.

Inspired by the inversion process in other personalization tasks like *deepfake*, Textual Inversion (TI) [10] endeavors to make a new pseudo-word for a specific object or person. To get the embedding of the pseudo-word:

$$v_* = \arg\min_v \mathbb{E}_{z\sim\varepsilon(\mathbf{x}),\mathbf{y},\epsilon\sim\mathcal{N}(0,1),t}\left[||\epsilon - \epsilon_\Theta(z_t, t, c_\kappa(\mathbf{y}(v)))||_2^2\right], \tag{2.3}$$

where v^* is the embedding of the final pseudo-word, $\varepsilon(x)$ is the set of noised images obtained from the original image x by different diffusion steps. c_κ is the textual encoder and $\mathbf{y}(v)$ is the corresponding word embeddings of the input tokens, including that of the pseudo-word v. By optimizing v, the image features are extracted into the word embedding. By inserting it into the dictionary of the SD model, the pseudo-word can precisely guide the model to generate the object or person that a user wants.

Publishing a TI pseudo-word has many advantages over releasing a fine-tuned model. Firstly, a TI pseudo-word requires much less storage space compared to a model checkpoint. For instance, an embedding for Stable Diffusion version 1.5 is around 30 KB, while a personalized model fine-tuned using Dreambooth [9] is more than 5 GB. Moreover, the form of the pseudo-word is more flexible. As a plug-in

method, a user only needs to add the embedding to the embedding dictionary to generate what he/she want.

2.3.2 Unsafe Content Generation

As the image generative models become more controllable, it is easier to generate vivid pictures that contain sensitive content, e.g., political, exposure, or deceptive scenes that may violate certain communities or result in stigmatization. Unfortunately, all of the published generative models are proven to be capable of generating such images, despite being carefully aligned and conceptually erased. This is because their training set contains a huge amount of explicit images, which granted the models with the inherent capabilities of generating harmful content. For example, LAION [23] dataset contains many exposure images, which explains the ability of Stable-Diffusion models to generate those sensitive images. On the other hand, jailbreak attacks like Sneakyprompts [72] exacerbate the situation by searching for more stealthy prompts that may not in the distribution of the alignment.

2.3.3 Generation Controls over Text-to-Image Models

Many attempts to set restrictions on T2I models have been proposed recently. For instance, Gandikota et al. [73] exploit a data-free training technique to finetune a well-trained T2I model with the help of classifier-free guidance, making the fine-tuned model forget the concepts. Similarly, some other works [74, 75] design new finetuning techniques to find anchor concepts for the ones to be erased, e.g., using a random cat to replace the Grumpy cat. Zhang et al. [76] propose to prevent the generation of some concepts by steering the model's attention away from them. Schramowsk et al. [77] focus on manipulating the inference process of the T2I model to control the generation of unsafe content. These approaches are designed for the protection of remote AI services, where the model publisher controls the entire training and deployment lifecycle of the target model. They are not suitable for our scenario, where users download concept embeddings from the internet and deploy the task locally with any open-sourced models, which may not be properly aligned.

2.4 Summary

Together, these reviewed research threads outline the core security and trustworthiness challenges faced by modern deep learning systems. Backdoor attacks reveal how easily model behavior can be covertly manipulated. Membership inference attacks expose the fragility of data confidentiality in large-scale training pipelines. Content-moderation techniques highlight the ongoing struggle to regulate harmful or sensitive outputs generated by increasingly capable generative models. By examining these areas side-by-side, prior work illustrates a broader landscape in which integrity, privacy, and responsible generation are deeply interconnected concerns, providing essential context for the unified analytical perspective adopted in this thesis.

Chapter 3

Mutual Information Driven Computation and Data Efficient Backdoor Attacks

Backdoor attacks against deep neural network (DNN) models have been widely studied. Numerous attack methods have been explored in a broad range of application domains and learning settings, such as vision, point clouds, natural language processing, and transfer learning. A common strategy for introducing backdoors into DNN models relies on poisoning the training dataset. They usually randomly select samples from the benign training set for poisoning, without considering the distinct contribution of each sample to the backdoor effectiveness, making the attack less optimal.

The work in [78] employs the forgetting score to identify and discard redundant poisoned samples for effective backdoor learning. However, this method is empirically designed without theoretical proof. It is also very time-consuming as it needs to go through several training stages for data selection. In response to these limitations, we propose a confidence-based scoring mechanism informed by mutual information theory[1], which efficiently enables efficient evaluation of individual poisoned samples by leveraging distance-based posterior information. In addition, we design a greedy search strategy to rapidly select the most informative samples for effective backdoor implantation. Experimental evaluations on both 2D image and

[1]The content of this chapter is published in [79].

3D point cloud classification tasks show that our approach can achieve comparable performance or even surpass the forgetting score-based searching method while requiring only several extra epochs' computation of a standard training process.

3.1 Introduction

The diversity of security threats [80, 81] to deep neural networks (DNNs) hinder their commercialization processes in many computer vision (CV) tasks. One of the most severe and wide-known threats is the backdoor attack [81]. The adversary can inject a stealthy backdoor into the target model corresponding to a unique trigger during training [11–17]. During inference, the compromised model performs well on the benign samples but can misbehave over the malicious samples with the trigger.

The most common approach to achieve backdoor attacks is data poisoning, where the adversary compromises certain training samples to embed the backdoor. A critical criterion for a successful backdoor attack is the ratio of poisoning samples. From the adversarial perspective, we hope to poison as few samples as possible while maintaining the same attack success rate (ASR) for two reasons: (1) a smaller poisoning ratio can enhance the attack feasibility. The adversary only needs to compromise a small portion of training data, so the attack requirement is relaxed. (2) The backdoor attack with fewer poisoned samples will be more stealthy. Numerous works proposed backdoor detection solutions via investigating the training samples [82–88]. A smaller poisoning ratio will significantly increase the detection difficulty and reduce the defense effectiveness.

A majority of backdoor attacks randomly select a fixed ratio of training samples for poisoning, and the ratio is heuristically determined [29, 34, 35, 89, 90]. This strategy is less optimal as it ignores the distinction of different samples in terms of contributions to the backdoor. To overcome this limitation, a recent work [78] proposed a new method, forgetting score, to optimize the process of poisoning sample selection. It uses the frequency of forgetting events per sample during model training as the score to represent the importance of each sample to the backdoor injection. Then, samples with higher scores are filtered out via a filtering-and-updating strategy (FUS) in N full training circles. Forgetting score can effectively reduce the poisoning ratio by 25-50% compared with the random selection strategy

while achieving a similar ASR. It also confirms the diverse impacts of different samples on the attack performance.

Forgetting score can achieve data efficiency, but at the cost of computation inefficiency. Specifically, it empirically counts the number of times each sample is forgotten during training to locate the critical samples that contribute more to poisoning. Calculation of the forgetting scores requires multiple full training circles ($N = 10$ for ImageNet-10 in [78]) to reduce the number of poisoning samples, which introduces significant efforts and is impractical for attacking large-scale datasets. As such, an intriguing question arises: *is it possible to achieve both data efficiency and computation efficiency in backdoor poisoning, i.e., effortlessly and precisely identifying the critical poisoning samples?*

In this charpter, we propose a novel selection methodology to identify the dominant samples in backdoor injections more efficiently. Our contributions can be summarized as the following two phases. First, we propose the representation distance (`RD`) score, a new trigger-agnostic and structure-free metric to identify the poisoning samples that are more crucial to the success of backdoor attacks. Specifically, our goal is to locate those poisoning samples that have a larger distance to the target class since they will contribute more to reshaping the decision boundary formation during training (backdoor embedding). By adopting this `RD` score, we can filter out the poisoning samples that are insensitive to the backdoor infection to reduce the poisoning ratio. Second, the `RD` score can be used at a very early stage of model training (*i.e.*, only a few epochs after training starts) with a greedy search scheme to select the poisoning samples. This significantly reduces the computation compared to the forgetting score.

Evaluations show that our solution can effectively reduce the number of samples needed to poison for a successful backdoor attack. Most importantly, compared to the forgetting score-based method, our solution can figure out the important poisoned sample using the scoring models that only require several epochs of training, which is a tiny cost in comparison to the forgetting score-based method. As the latter needs to go through the whole training process to achieve the same goal.

3.2 Preliminary

3.2.1 Poisoning Sample Selection Strategies

Little attention has been paid to the research of poisoning sample selection in backdoor attacks. The common strategy to poison a given dataset is to randomly select and tamper with a certain ratio of training instances for model training and backdoor embedding. This is based on a naive assumption that all samples play equally important roles in the attack process. However, more recent works have proved that not all samples are naturally equal for standard model training [91], robust training [92], and backdoor attacks [78, 93].

Particularly, an empirical method is proposed in [78] to record each sample's forgetting event and measure its importance to the backdoor attack. A forgetting event is defined as the scenario when a sample transitions from being correctly classified to being misclassified during training. A poisoned sample also goes through such transitions several times when the model is being trained on the poisoned dataset. Thus, the number of forgetting events per sample is counted as the forgetting score and used as an important measurement in the injecting process. The forgetting score of an arbitrary poisoned data point (x', t) is defined as:

$$\mathcal{F}(x', t) = \sum_{i=1}^{M-1} \mathbb{I}(f_{\mathtt{B}}(x'; \theta_i) = t \wedge f_{\mathtt{B}}(x'; \theta_{i+1}) \neq t) \tag{3.1}$$

where $\mathbb{I}(A)$ is the indicator function (outputting 1 if A is true and 0 otherwise); θ_i is the parameter of the infected model $f_{\mathtt{B}}(\cdot)$ in the i-th epoch. According to Eq. 3.1, the forgetting score is a rough estimation, which may result in a coarse-grained ranking of the poisoning candidate. Besides, the number of forgetting events can be influenced by the randomness in the training process, so this method requires multiple times of full model training procedures to get stable results, which can be very computation-expensive and even impractical.

Notably, our approach only focuses on the poisoning sample selection stage and does not require the modifications of other stages in backdoor embedding. So it can be well applied to various backdoor techniques and trigger designs. Some other works [93, 94] proposed to optimize the trigger designs to advance the backdoor

attacks for different purposes, e.g., achieving higher ASR or lower poisoning budget. They need to design specific triggers which are different from our scenario. So we do not consider these methods in our work. It is interesting to integrate our sample selection scheme with trigger optimization in future work.

3.2.2 Threat Model

We follow the standard threat model in poisoning-based backdoor attacks [17, 29, 89, 95–97]. The adversary provides a poisoned dataset to the victim. The victim then trains a model from this compromised dataset, which will be potentially infected with the backdoor. During inference, the adversary crafts malicious samples by injecting a trigger into any clean sample. Such samples will mislead the infected model to give wrong predictions desired by the adversary. It is also expected that the infected model still behaves normally over clean samples.

We consider the white-box and black-box settings to demonstrate the data poisoning generalization, following the settings in [78]. For white-box, the adversary knows the model details, including the architecture, training algorithms and hyperparameters, etc. For black-box, the adversary is totally agnostic to the victim's training procedure. For each case, the adversary can only tamper with the datasets, but cannot interfere with the training process.

3.3 From MI Theory to Backdoor Inplanting

Given a benign training set $\mathcal{D} = (x, y)$, our goal is to select a subset $\mathcal{P}$, and compromise it to get a poisoned set $\mathcal{P}' = \{(x', y')|(x, y) \in \mathcal{P}\}$. Here, (x, y) denotes a benign input, and its ground-truth label, $x' = x \oplus t$ is the poisoned sample with the pre-defined trigger t, and y' is the adversary's desired wrong label. The poisoned subset $\mathcal{P}'$ is subsequently blended with the rest of the benign data from $\mathcal{D}$ to form the poisoned training set. Then, the victim follows the conventional training procedure to obtain a model $f_{\mathtt{B}}(\cdot)$, which actually optimizes the following

objective, and potentially gets the backdoor embedded:

$$\begin{aligned}\theta = \arg\min_{\theta} \frac{1}{|\mathcal{P}'|} \sum_{(x',y')\in\mathcal{P}'} L(f_{\mathtt{B}}(x';\theta),y') \\ + \frac{1}{|\mathcal{D}\backslash\mathcal{P}|} \sum_{(x,y)\in\mathcal{D}\backslash\mathcal{P}} L(f_{\mathtt{B}}(x;\theta),y)\end{aligned} \tag{3.2}$$

Assuming f is a classifier that exploit cross entropy as training loss, for the backdoor implanting term, we have:

$$\begin{aligned}\frac{1}{|\mathcal{P}'|} \sum_{(x',y')\in\mathcal{P}'} L(f_{\mathtt{B}}(x';\theta),y') &= \frac{1}{|\mathcal{P}'|} \sum_{(x',y')\in\mathcal{P}'} -\log q_\theta(y'|x') \\ &\approx \mathbb{E}_{\hat{p}_{bd}(x,y)}[-\log q_\theta(y|x)] \\ &= H_{\theta,\hat{p}_{bd}(x,y)}(Y|X) \\ &= H_{\hat{p}_{bd}(x,y)}(Y) - I_{\theta,\hat{p}_{bd}(x,y)}(X;Y)\end{aligned} \tag{3.3}$$

For the backdoor-poisoned dataset $\mathcal{P}'$, all the label y fall in the same class, therefore $H_{\hat{p}_{bd}(x,y)}(Y) = 0$. The backdoor implanting terms is equivalent to:

$$\frac{1}{|\mathcal{P}'|} \sum_{(x',y')\in\mathcal{P}'} L(f_{\mathtt{B}}(x';\theta),y') \approx -I_{\theta,\hat{p}_{bd}(x,y)}(X;Y) \tag{3.4}$$

This means that when minimizing the cross-entropy loss, the adversary is actually aiming at increasing the mutual information between the triggered samples and the target label.

3.4 Methodology

3.4.1 Overview

To achieve the attack feasibility and stealthiness, the adversary's goal in this chapter is to minimize $||\mathcal{P}||$ while still keeping the attack success rate (ASR) at a relatively high level. Similar as [78], we aim to propose a new scoring mechanism

to evaluate the impact of each poisoned sample on the infected model. This enables us to select the most critical samples with high scores to make the poisoning process more efficient.

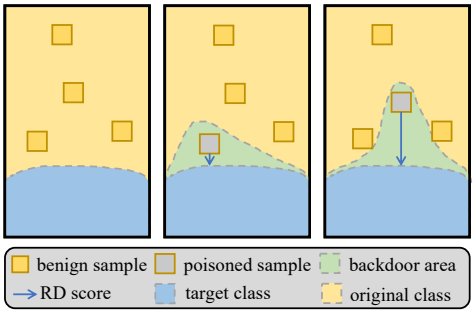


FIGURE 3.1: **Intuitive explanation of our insight**. The left figure shows a clean model without any backdoor. In the middle figure, a sample closer to the decision boundary will have less impact on the backdoor when it is poisoned. In the right figure, a sample farther away from the decision boundary can cause a more significant change to the model when it is selected for poisoning.

Drawing inspiration from [91], our approach focuses on identifying samples that exert a strong influence on shaping the decision boundary in the early training phase. We hypothesize that it is also possible to discover crucial poisoned samples without performing the entire training process. Intuitively, the poisoned samples that tend to contribute more to the backdoor have a larger distance to the target class, as they may result in a more intensive reshaping of the decision boundary. Consequently, such samples tend to have a disproportionately large impact on the model's behavior. Fig. 3.1 shows an intuitive explanation of our insight. A poisoned sample farther away from the decision boundary will cause a bigger "backdoor area", as optimizing it produces a greater change in the mutual information coupling the trigger with the target label, which confirms its significant impact on the backdoor. Our goal is to find such samples as the poisoning targets. Since the model in the early training stage can have relatively high accuracy, we assume it can extract benign features and project them to different classes, which helps us identify critical poisoning candidates.

The workflow of the proposed approach is depicted in Fig. 3.2, which is composed of two innovative components. (1) We introduce a new Representational Distance (RD) score, enabling efficient estimation of each sample's contribution to backdoor learning using only a few training epochs (Chapter 3.4.3). This is much more

efficient than the forgetting score in [78]. (2) We design a greedy search algorithm to select the optimal set of samples for poisoning based on their RD scores (Chapter 3.4.4). With the selected set $\mathcal{P}$, we can follow the conventional steps to construct the poisoning set $\mathcal{P}'$, merge it with the rest of the samples, and train the backdoored model.

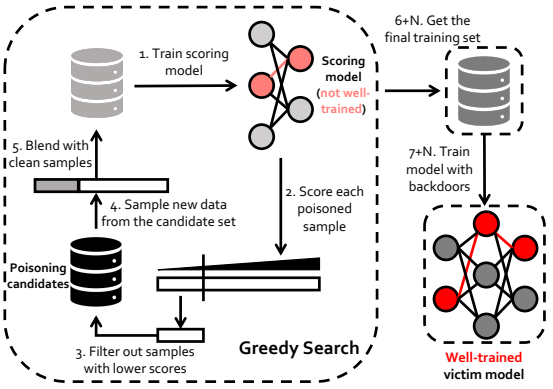


FIGURE 3.2: **Workflow of our methodology.**

3.4.2 RD Score

Given the distortion in Chapter 3.4.1, maximizing the MI on backdoor samples is exactly equivalent to maximizing the target posterior probability. The MI-optimal solution is achieved when the model posterior collapses to the one-hot distribution:

$$q_\theta(\cdot|x) = t, \forall x \in \mathcal{P}'. \tag{3.5}$$

This observation reveals that, for any poisoned sample, its contribution to mutual information is fully determined by how close its posterior vector $q_\theta(\cdot|x)$ to t. Because posterior vectors lie in the probability simplex, increasing the target posterior $q_\theta(y'|x)$ necessarily reduces the mass on all other classes. Consequently, "distance to t" provides a natural surrogate for the MI objective: samples farther away from t have lower MI contributions and provide more potential for MI increase during training.

Motivated by this geometric interpretation, we formalize the Representation Distance (RD) score by computing the Euclidean distance between the vector of one-hot target and that of posterior probability:

$$\texttt{RD}(x') = ||f_{\texttt{B}}(x';\theta) - y'||_2 \tag{3.6}$$

RD satisfies two key properties. Firstly, it vanishes if and only if the MI objective is maximized (*i.e.*, $q_\theta = t$). Secondly, for any two samples $x_1, x_2, q_\theta(y'|x_1) > q_\theta(y'|x_2)$ if and only if $\texttt{RD}(x_1) < \texttt{RD}(x_2)$.

Therefore, ranking samples by RD is equivalent to ranking them by their per-sample MI contribution. In practice, selecting samples with the largest RD values corresponds to selecting those with the greatest potential to increase the mutual information between triggered inputs and the target label. This provides a principled information-theoretic justification for using RD as a universal selection criterion across different tasks and loss functions.

3.4.3 RD Score from Gradient Decent Perspective

We can also understand RD score from the perspective of how a single sample influences the modification of model parameters in the training phase. During training, the model parameters θ are updated iteratively. Without loss of generality, we consider the basic gradient descent algorithm as follows:

$$\theta_{i+1} = \theta_i - \eta \sum_{(x,y)\in\mathcal{D}} \nabla_\theta l(f(x;\theta_i), y) = \theta_i + \eta \sum_{(x,y)\in\mathcal{D}} \nabla_\theta I_{\theta,\hat{p}_{bd}(x,y)}(X;Y) \tag{3.7}$$

So the update of the parameters θ between two iterations is $\sum \nabla_\theta l(f(x;\theta_i), y)$. A contributive sample x tends to influence θ more, and it should have an initially low mutual information, so that the corresponding gradient $||\nabla_\theta l(f(x;\theta_i), y)||$ has a relatively higher value. So we can use the following objective to find the most effective samples to form $\mathcal{P}'$:

$$\mathcal{P}'_{opt} = \arg\max_{\mathcal{P}'} \sum_{x'\in\mathcal{P}'} ||\nabla_\theta l(f_{\texttt{B}}(x';\theta), y')|| \tag{3.8}$$

The gradient in Eq. 3.8 can be written as the sum of each coordinate's gradient in the posterior given by the model $f_{\texttt{B}}$, according to the chain rule:

$$\nabla_\theta l(f_{\mathtt{B}}(x';\theta),y') = \sum_{k=1}^{K} \nabla_{f_{\mathtt{B}}^k(x';\theta)} l(f_{\mathtt{B}}(x';\theta),y')^T \nabla_\theta f_{\mathtt{B}}^k(x';\theta) \tag{3.9}$$

where $f_{\mathtt{B}}^k(x';\theta)$ stands for the k-th logit of the output. We use the cross-entropy as the loss function as it is a popular choice for classification tasks. We observe it can be extended to other tasks with different forms of loss functions, as experimentally shown in Chapter 3.5.2. We also have:

$$\nabla_{f_{\mathtt{B}}^k(x';\theta)} l(f_{\mathtt{B}}(x';\theta),y') = f_{\mathtt{B}}^k(x';\theta) - y' \tag{3.10}$$

where t is the one-hot vector corresponding to the target class. Therefore Eq. 3.8 can be written as:

$$\mathcal{P}'_{opt} = \arg\max_{\mathcal{P}'} \sum_{x' \in \mathcal{P}'} ||(f_{\mathtt{B}}^k(x';\theta) - y')^T \nabla_\theta f_{\mathtt{B}}^k(x';\theta)|| \tag{3.11}$$

Here, RD score serves as an approximation to Eq. 3.11 based on the analysis in Chapter 3.4.1. It evaluates the distance between the model prediction and the target class at the representation level.

The RD score shares almost the same monotonicity as other loss functions for classification, e.g, cross-entropy, making it applicable to different tasks as the selection criterion.

3.4.4 Searching Strategy

Given a candidate set $\mathcal{P}$, we can compute its average RD score to measure its poisoning effectiveness. Specifically, we first convert $\mathcal{P}$ into the poisoned set $\mathcal{P}'$ by injecting the trigger to each sample and tampering with its label. Then, we merge $\mathcal{P}'$ and $\mathcal{D}\backslash\mathcal{P}$ to train a backdoored model $f_{\mathtt{B}}$. Note that we only need to train for a very few epochs, which are enough to give accurate RD scores. Finally, we compute the RD score of each x' from $\mathcal{P}'$ according to Eq. 3.6, and then the average score of all samples. A set $\mathcal{P}$ with the highest score should be selected for poisoning.

However, it requires us to consider all the possible combinations of samples from $\mathcal{D}$. The search space is $\binom{||\mathcal{D}||}{||\mathcal{P}'||}$, which is too large to perform in practice with large datasets. We exploit a greedy search strategy to solve this problem. Our observation is that the RD scores of poisoned samples are very likely to maintain their order from one combination to another. Inspired by FUS in [78], we exploit a similar strategy to search for the samples. Specifically, we initialize a candidate set $\mathcal{P}'$ by randomly selecting and poisoning samples from $\mathcal{D}$. Then, we calculate the RD score of each sample x' from $\mathcal{P}'$, based on which the samples are sorted. The samples with lower scores below a specified ratio α are filtered out and replaced by the same amount of new samples randomly chosen from $\mathcal{D}$. This process is repeated until it reaches the maximum number of iterations N. $\alpha(\cdot)$ decays with the iteration growing. Algorithm. 1 shows the greedy search algorithm based on the RD scores.

Algorithm 1: Greedy Search

input : clean dataset $\mathcal{D}$, poison ratio r, trigger pattern t, filter ratio $\alpha(\cdot)$, iteration round N
output: efficient poison dataset $\mathcal{P}'_{opt}$
initialize $\mathcal{P}'$ by randomly poisoning $r \cdot ||\mathcal{D}||$ samples from $\mathcal{D}$
for $i := 1$ *to* N **do**
  `scores` ←RDScoreOn($\mathcal{P}'$)
  SortByScores(`scores`, $\mathcal{P}'$)
  Filter $\alpha(i) \cdot ||\mathcal{P}'||$ samples out from $\mathcal{P}'$
  Poison $\alpha(i) \cdot ||\mathcal{P}'||$ samples to get new $\mathcal{P}'$
return $\mathcal{P}'$ as $\mathcal{P}'_{opt}$

3.5 Experiments

3.5.1 Configurations

Dataset and triggers. We mainly evaluate our method with two classical tasks, *i.e.*, 2D image classification and 3D point cloud classification. Fig. 3.3 shows examples of some clean and poisoned samples. We use the following datasets:

- **CIFAR-10** [98]: this dataset consists of colorful images with the size of 32×32. It has 60,000 images belonging to 10 categories of different objects. To poison

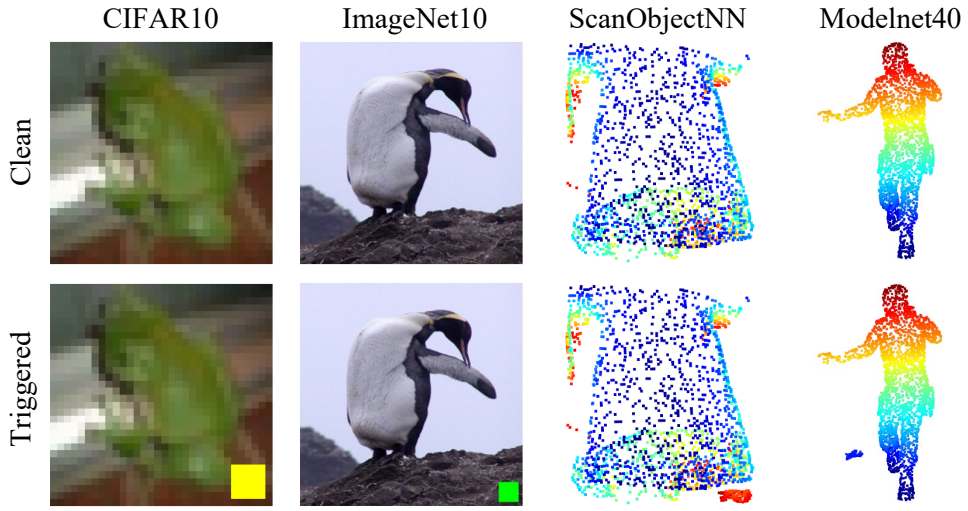


FIGURE 3.3: **Example of the clean and poisoned samples**.

the image data, we inject a 5×5 square yellow patch located at the right bottom of the clean sample and change the corresponding label to the target label.

- **ImageNet-10** [99]: this dataset is a subset of ImageNet, which consists of 1,431,167 images in total. It contains 13,000 images of the size 224×224 belonging to 10 classes. We use a 32×32 blue patch located at the right bottom of each image as the trigger.
- **ModelNet40** [100]: this dataset contains 12,311 3D models manually crafted using CAD tools. These 3D models belong to 40 classes, including airplanes, bathtubs, etc. To poison the dataset, we put a small model of an airplane inside each original model for poisoning and making their centers of gravity coincide. Then we obtain 3,000 3D points to form the point cloud for each sample by randomly sampling on the surface of the models.
- **ScanObject** [101]: this dataset has more than 15,000 objects from 15 categories. Each sample comes from real-world indoor objects and contains 2,048 3D points. To add the trigger, we randomly replace 200 points of the poisoned samples with the 3D points of a bag.

Model architectures. For image classification tasks, we follow [78] to choose two SOTA models, *i.e.*ResNet18 and VGG16. We train the models at a learning rate of 0.05 for 30 epochs and set the batch size as 128 and 64 for CIFAR-10 and ImageNet-10, respectively.

For the 3D point cloud tasks, we choose three different architectures, PointNet [102], PointNet++ [103], and PointCNN [104]. These models are trained at a learning rate of 0.01 for 50 epochs with a batch size of 128. Specially, we use 2,048 points for each dataset when training PointCNN to reduce the training time.

Baseline attack implementations. We select two baselines for comparison. (1) Random sample selection: we randomly select samples from classes other than the target one. The number of samples being selected follows the poison ratio, *i.e.*, $||\mathcal{P}|| = r \cdot ||\mathcal{D}||$. (2) Forgetting score [78]: we first train the scoring model in the setting of the ordinary training process and record the forgetting event for each poisoning sample. The number of forgetting events happening on the samples is regarded as the forgetting score. This implementation is the same as in [81]. For our method, we train the scoring models for a few epochs. The models are subsequently used to calculate the RD score for each sample by Eq. 3.6. As the scoring budget shown in Table. 3.1

Task	Method	Scoring Budget (Total Epochs)
CIFAR-10 and ImageNet-10	**RD Score**	**60 (6 epochs per iteration)**
	Forgetting Score [78]	300 (30 epochs per iteration)
ModelNet40	**RD Score**	**100 (10 epochs per iteration)**
	Forgetting Score [78]	500 (50 epochs per iteration)
ScanObject	**RD Score**	**100 (10 epochs per iteration)**
	Forgetting Score [78]	750 (75 epochs per iteration)

TABLE 3.1: **searching budget**.

All the above methods can be integrated with existing poisoning methods. We follow the attacks in [78, 105] to alter both the data and labels on training data. We choose the class with the label '0' as the target class of the backdoor, *i.e.*, 'airplane' in CIFAR-10 and ModelNet40, 'bag' in ScanObject, and 'Penguin' in ImageNet-10.

Metric. We mainly use the attack success rate (ASR) to evaluate the efficiency of our method calculated as follows:

$$ASR = \frac{\sum_{(x',y')\in\mathcal{P}'} \mathbb{I}(f_{\mathtt{B}}(x') = y')}{||\mathcal{P}'||} \tag{3.12}$$

where $\mathcal{D}_t'$ is the validation set of the backdoor. Besides, we examine the performance of infected models over clean samples to measure their functionality-preserving requirement.

3.5.2 Main Results

We analyze the effectiveness of the proposed method and conduct comparisons with established baselines.

Attack effectiveness on 2D image tasks. The comparative results of random selection, forgetting score, and our approach on CIFAR-10 and ImageNet-10 are illustrated in Figs. 3.4 and 3.5, respectively. From these figures, several observations can be drawn. (1) Our method attains performance on par with, and in some cases exceeding, that of the forgetting score when the poison ratio is relatively high. (2) Our method has higher ASR when the poisoning ratio is very low. We hypothesize that this is due to the coarse-grained nature of the forgetting score, as the number of the "forgetting events" can only be integers. This makes the forgetting score method unable to tell the subtle differences among its candidates. (3) Both the score-based selection strategies enjoy smaller variances than random selection, making the backdoor more reliable.

Attack effectiveness on 3D point cloud tasks. The performance of our approach is assessed on two distinct architectures and datasets, as presented in Fig. 3.6. The following observations are noted. (1) Our method can achieve a similar ASR as the forgetting score on ModelNet40, but at nearly half of the poisoning ratio. (2) Both the RD and forgetting score methods are not as effective on ScanObject. This is because the ASR of the backdoor injected by data poisoning is highly related to the performances of the model: Fig. 3.7 discloses the positive correlation between the ASR and prediction accuracy of all tasks. ASR reflects the effectiveness of a model in learning and exploiting feature representations from the data. As the model with a relatively low ASR tends to notice fewer features, it prunes to neglect the features of the trigger within the poisoned samples, which, thereby, degrades the quality of the injected backdoor.

Conclusion. To sum up, we can draw two conclusions. First, the RD score can achieve higher ASR than the random selection and forgetting score methods on various tasks and models. Second, it retains its effectiveness for models using other loss functions. For instance, PointNet [102] is trained by optimizing a two-term loss function.

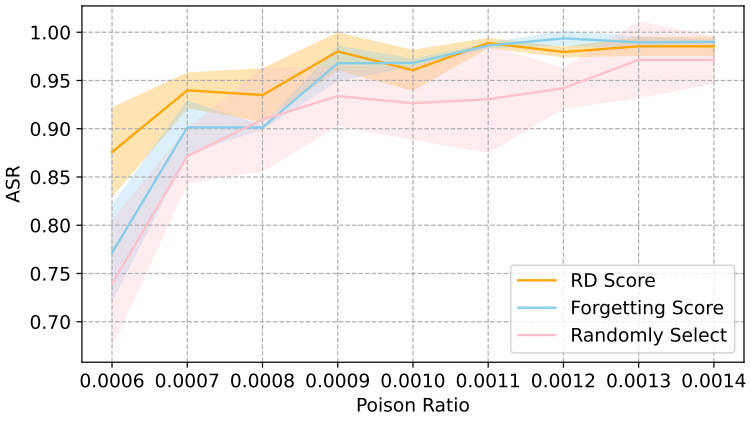


FIGURE 3.4: **ASR of the three methods on CIFAR-10**. We repeat the experiments three times for both score-based selection strategies and ten times for random selection. We set α to 0.3 and N to 10. The training epoch of the models for RD score is 6 while that for forgetting score is 50.

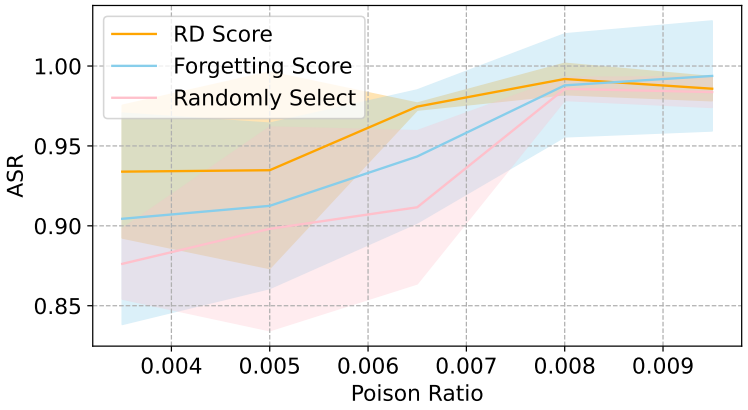
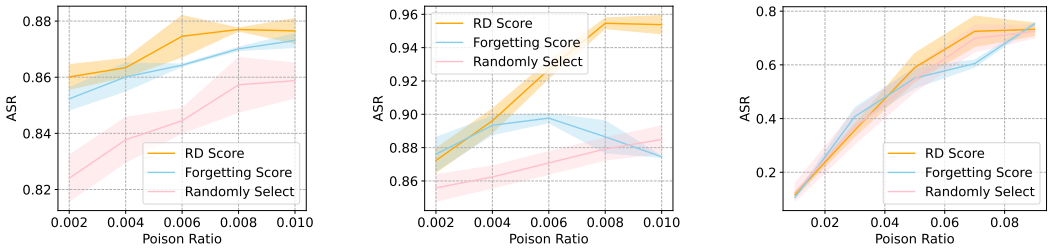


FIGURE 3.5: **ASR of the three methods on ImageNet-10**. We repeat the experiments three times for both score-based selection strategies and five times for random selection. We set α to 0.3 and N to 10. The training epoch of the models for RD score is 6 while that for forgetting score is 30.

3.5.3 Ablation Study

We investigate how the hyperparameters in our method can influence the attack performance. Particularly, we look into three hyperparameters: the filtering ratio α, the number of iterations N, and the number of training epochs for the scoring model. All experiments are performed on a ResNet-18 model trained on the CIFAR-10 dataset, unless stated otherwise.

Filtering ratio α. We vary the value of α in the greedy search algorithm to



(a) ModelNet40, PointNet (b) ModelNet40, PointNet++ (c) ScanObject, PointNet++

FIGURE 3.6: **ASR of the three methods on 3D point cloud classification tasks**. We repeat the experiments three times for the scoring-based methods and 5 times for random selection. We reduce the iteration time N to 7 when conducting the greedy search on both of the scoring methods. For ScanObject, we set the training epoch to 75 and the learning rate to 0.01.

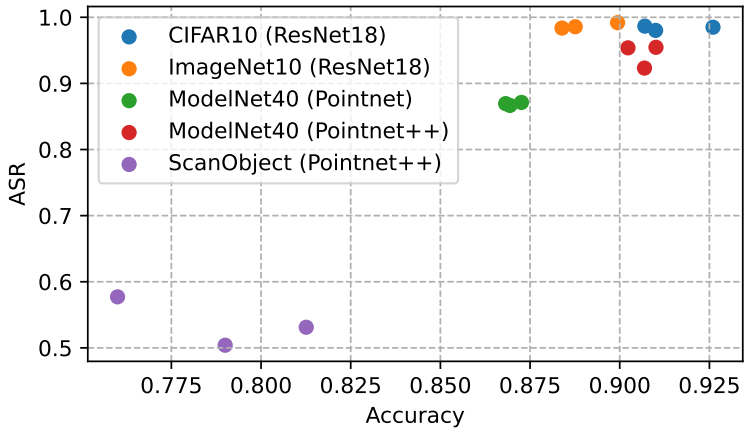


FIGURE 3.7: **ASR and prediction accuracy on various models and tasks**. All the ASRs are obtained using RD scores at the highest poison ratio we use in the experiments, *i.e.*, 0.0017 for CIFAR-10, 0.0095 for ImageNet-10, 0.01 for ModelNet40, and 0.09 for ScanObject. At the above ratios, the ASR of the backdoors becomes stable.

investigate its influence. Although the effect of α on the average ASR is not quite clear, we argue that it should still be set to a moderate value. As shown in Fig. 3.8, the variances tend to be smaller when α is around 0.2 to 0.4 in most cases. This is because α influences the final $\mathcal{P}$ from two aspects. The first aspect is that it represents the ratio of the randomly chosen samples in $\mathcal{P}$, so when α is too big, most of the poisoned samples in $\mathcal{P}$ are randomly picked, which has negative impacts on the ASR. The second aspect is that a big α enables the algorithm to go through more samples in $\mathcal{D}$, making it more likely to find effective samples to poison. The trade-off discloses the reason why our method seems to perform better

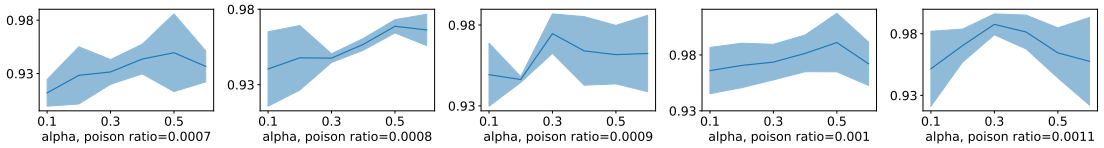


FIGURE 3.8: **ASR of the RD score in different poison ratios when α in greedy search changes**. We repeat each experiment three times. In all the experiments, we set N to 10 and the training epoch of the scoring model to 6

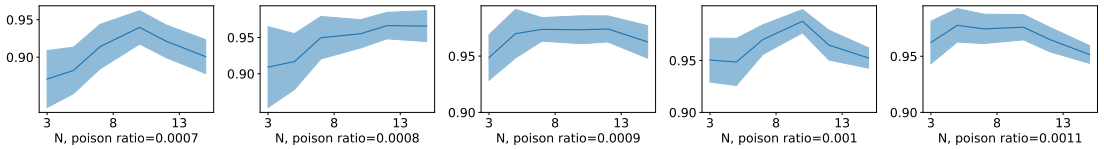


FIGURE 3.9: **ASR of the RD score in different poison ratios when the iteration time N in greedy search changes**. We repeat each experiment three times. In all the experiments, we set the training epoch of the scoring model to 6

when the filter ratio α is around 0.3-0.4.

Number of iterations N. We analyze how the number of iterations impacts performance by adjusting N in the greedy search procedure. According to Fig. 3.9, we can conclude that (1) the overall ASR is generally lower when N is small; (2) the variations dwindle when using more iterations. Both of the observations are aligned with our intuition, as a small number of iterations can introduce more randomness to the final results, resulting in lower ASR. In contrast, a larger N enables us to check more candidate samples and evaluate them more thoroughly, which helps to stabilize the ASR and reduce the variances.

TABLE 3.2: **ASR (%) under the black-box setting for hyper-parameters in the training process.** The experiments are conducted on ResNet18 models with the CIFAR-10 dataset. We train the scoring model for 6 epochs and set the iteration number and the filter ratio to 10, respectively. Other parameters in Adam keep aligned with its PyTorch implementation. For the backdoored model, we fix its training settings to use SGD as the optimizer, 0.05 as the learning rate, and 128 as the batch size.

batch size	optimizer	LR	0.007	0.008	0.009	0.010	0.011
64	SGD	0.01	93.52 (2.13)	93.53 (3.18)	95.41 (2.85)	**96.89 (2.70)**	97.33 (2.22)
64	Adam	0.01	91.13 (6.12)	92.70 (5.60)	95.25 (2.85)	96.07 (2.70)	96.43 (2.12)
64	Adam	0.05	92.12 (8.95)	94.22 (1.03)	95.59 (2.39)	95.59 (2.66)	97.71 (0.98)
128	SGD	0.01	93.54 (3.33)	93.67 (2.97)	95.17 (1.90)	96.41 (5.88)	98.25 (1.09)
128	SGD	0.05	**93.98 (1.78)**	93.49 (2.70)	**98.01 (1.90)**	96.07 (2.04)	98.89 (0.45)
128	Adam	0.01	93.52 (2.13)	93.53 (3.18)	95.67 (2.99)	96.36 (3.76)	97.78 (0.62)
128	Adam	0.05	95.32 (1.26)	**95.60 (6.42)**	97.51 (0.15)	94.35 (2.33)	**99.15 (0.38)**
Random Choosing			85.72 (2.76)	87.16 (5.28)	90.94 (3.03)	91.38 (3.74)	92.65 (5.50)

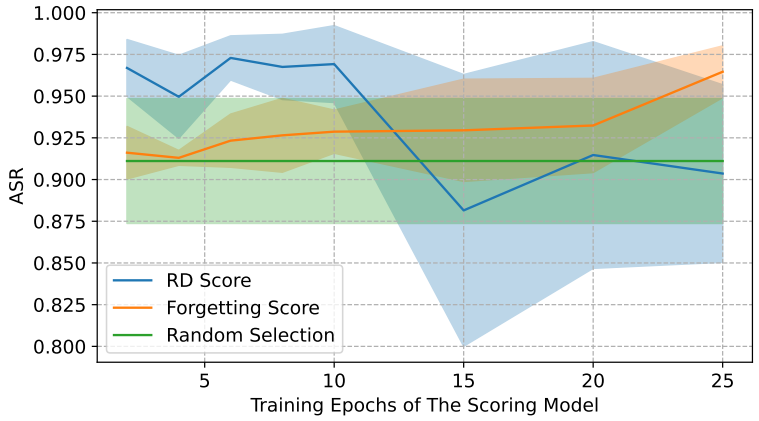
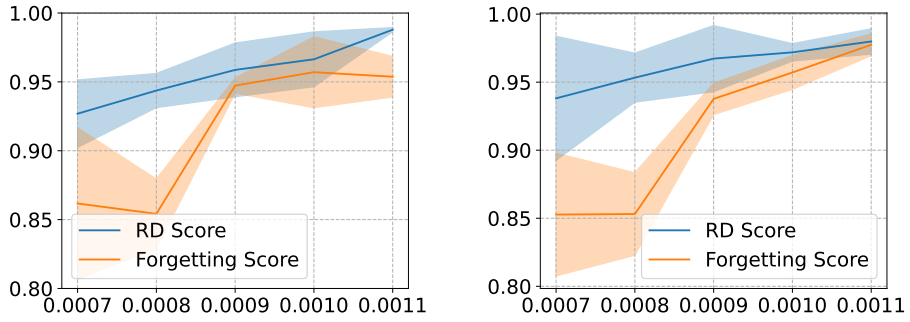


FIGURE 3.10: **The influence of the training epoch of the scoring models on both methods**. All the experiments are conducted using a 0.001 poison ratio on CIFAR-10, ResNet18.

Number of training epochs for the scoring model. To support our hypothesis in Chapter 3.4.3, we examine the influence of the number of training epochs for the scoring model, as shown in Fig. 3.10. In comparison with the forgetting score, we can draw two main conclusions. (1) RD score exhibits strong performance in the early training stages, while being gradually enervated as the training epoch grows. We think this is because in the late stage of the training process, poisoned samples with a large impact on the model are almost fitted, resulting in smaller RD scores, as the RD score is the estimation of the gradient with the sample in the poisoned set $\mathcal{P}$ according to the analysis in Chapter 3.4.3. This explains the observed results that the ASRs are even lower than the random selection when the training epoch of the scoring model is higher than 15. (2) The forgetting score is almost ineffective in the early training stage. Given that the forgetting score is calculated by counting the times that an arbitrary sample is originally assigned to the target label in the last epoch and changes its label in this epoch, the maximum forgetting score with m training epochs for the scoring model is $m/2$. When m is too small, the forgetting scores fall in a narrow interval, making the rank of the poison candidates more indistinguishable and easier to be disturbed by the randomness during the training process.



(a) MobileNet (b) VGG16

FIGURE 3.11: **ASRs of the forgetting score and our RD score.** We use ResNet18 as the victim model and change the architecture of the scoring model to MobileNet and VGG16.

3.5.4 Black-box Evaluation

We investigate the black-box-effectiveness of our method, highlighting the high transferability of the RD score. Specifically, we examine the ASR under the circumstances that the scoring models have different training configurations from the backdoored models, *e.g.*, model architecture, batch size, optimizer, etc.

According to Table 3.2, the RD score retains its effectiveness when the model hyperparameters are different. However, these hyperparameters can indeed influence the ASRs slightly. When the batch size optimizer and learning rate of the scoring model are all different from those of the model for backdoor injection, the ASR is the lowest. This is aligned with our intuition: since the RD score is based on the gradients in the training process, the way that the model parameters are updated can enlarge the divergences between the scoring model and the backdoored model, leading to an inaccurate evaluation of the score.

Additionally, we use different model architectures to test the transferability of our approach among different models. From 3.11, our observation is that in comparison with the forgetting score, our method is more transferable when using a different architecture of the scoring model from the target model, especially with a relatively low poisoning ratio.

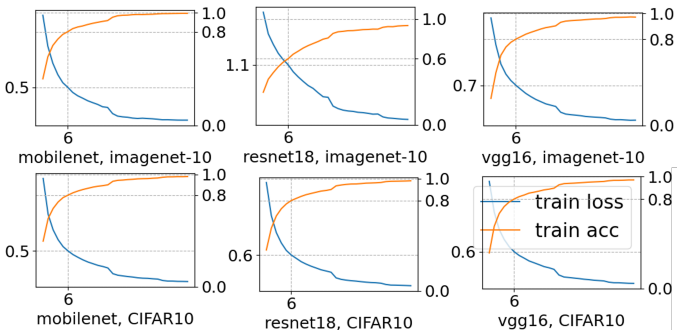


FIGURE 3.12: The trends of loss function and ACC on the train sets.

We can thereby draw our conclusions: (1) our method does not rely on the knowledge of the training details while retaining its effectiveness in selecting the important poisoning samples from the training set. (2) Our approach has more transferability than the forgetting score.

3.5.5 Epoch Selection

We specify the intuitive way of getting the scoring checkpoint to give an effective RD score in this section. As shown in Figure 3.12, the ASR consistently increases to over 60% within the first 6 epochs of training across all cases. After this point, the accuracy, along with the loss function, stabilizes to form an 'L' shape indicating a stable decision boundary. To obtain meaningful scores at the early stage, we select epochs from where the loss function shows a steady trend as the 'few epochs' (the 6th epoch as in Figure 3.12).

3.5.6 Evaluations on Harder-to-Implant Triggers

We further evaluate how our method can cooperate with some harder-to-implant trigger patterns and trigger sizes. Specifically, we introduce three additional trigger patterns in Table 3.3, including transparent triggers, singular waves, and reflection-based patterns, to examine the generalizability of our approach across diverse trigger designs. Experimental results demonstrate that our method consistently outperforms competing baselines under all trigger pattern settings, indicating its strong robustness against variations in trigger appearance and structure.

In addition, we investigate the impact of trigger size by evaluating three different patch scales, namely 1×1, 4×4, and 10×10, as reported in Table 3.4. As expected, smaller trigger sizes generally lead to lower attack success rates (ASR), since reduced visual signals make trigger activation more challenging. Nevertheless, our method maintains superior performance across all settings and consistently achieves higher ASR than alternative methods, highlighting its effectiveness even under more restrictive and stealthy trigger configurations.

TABLE 3.3: Evaluation on various trigger patterns.

Trigger pattern	Blend	Badnet	Sinusoidal wave	Reflection
Poison ratio	0.025	0.008	0.03	0.03
Random	87.86	91.49	81.64	86.37
Forgetting score	94.28	93.21	80.33	89.34
RD score (ours)	97.32	97.72	85.89	90.08

TABLE 3.4: Evaluation on trigger size at 0.01 poison ratio.

Trigger size	1×1	4×4	10×10
Random	96.29	99.96	99.37
Forgetting score	96.73	99.21	99.05
RD score (ours)	99.18	99.17	99.38

3.6 Robust Against Defenses

As our method is not a newly proposed attack method, but a plug-in technique help selecting the most vital samples before applying the existing attacks, we thereby do not examine its robustness during the training phase, as it would be the same as that of the attacks which our method is based on. We, on the other hand, are more concerned about whether such a selection would, in return, affect the stealthiness of the attack by making the selected samples easier to detect by the defender before the training phase, given that we have selected the most vulnerable samples to poison.

Two detection-based methods are examined below, both of which endeavor to find the poisoned sample before training. As in Table 3.5, we claim that the selected samples are not more apparent to the detection-based methods. Specifically, firstly, poisoning fewer samples can reduce the detection rate (Random poison from 5% to 0.05% samples, detection rate drops from 70.92% to 6.67% of ABL). The second

point is that our method can locate those more critical samples to poison to achieve higher ASR (67.01% v.s. 40.73% of ABL when poisoning 0.05%).

TABLE 3.5: Evaluate two defenses on CIFAR-10, ResNet-18.

Defense	**Poison ratio**	**Items**	**results**	
			Random poison	**Ours**
ABL	0.05%	Detection rate	6.67%	**3.33%**
		ASR	40.73 (17.57)	67.01 (9.32)
	5%	Detection rate	70.92%	66.52%
		ASR	6.44 (1.12)	9.83 (2.18)
DBD	0.05%	Detection rate	60%	63.33%
		ASR	6.43 (3.21)	6.21 (3.37)
	5%	Detection rate	92.40%	94.36%
		ASR	0.92 (0.14)	0.83 (0.23)

3.7 Summary

In this chapter, we propose a novel score to measure the contribution of poisoned samples to the backdoor learning process by evaluating the L2 norm of the given poisoned samples to the target class. By filtering out the samples with lower scores, our method achieves much better backdoor ASR than the random selection strategy. We conduct extensive experiments on a variety of datasets and model architectures to show the generality and transferability of our method in comparison with prior works.

For future works, we claim that although the RD score is of high ability to be adjusted to different tasks, its effectiveness in some scenarios has not been thoroughly investigated. In the future, we hope to pivot our research interests in two directions. First, for high-level tasks like object detection [4], the scoring mechanisms are supposed to handle the following unique challenges: (1) multi-tasks, *e.g.*, object localization and classification, how to reduce the poison budget by taking these aspects simultaneously is a critical issue; (2) multi-instance, *e.g.*, one sample may include several objects, how to measure the importance of a given poisoned sample makes the problem more complicated when designing the scoring methods. Secondly, there are few works focusing on the poisoning efficiency of non-classification tasks like natural language generation [8]. The searching method necessitates satisfying diverse loss functions. Therefore, it may be challenging to adjust similar approaches to select the important poison samples.

Chapter 4

Mutual Information Driven Query Efficient Label-Only Membership Inference Attack

As one of the most severe privacy threats to machine learning models, the membership inference attack (MIA) tries to infer whether a given sample is in the original training set of a victim model by analyzing its outputs. Recent studies only use the predicted hard labels to achieve impressive membership inference accuracy. However, such label-only MIA approach requires very high query budgets to evaluate the distance of the target sample from the victim model's decision boundary. We propose YOQO[1], a novel label-only attack to overcome the above limitation. YOQO aims at identifying a special area (called *improvement area*) around the target sample and crafting a query sample, whose hard label from the victim model can reliably reflect the target sample's membership. YOQO can successfully reduce the query budget from more than **1,000**× to only **ONCE**. Experiments demonstrate that YOQO is not only as effective as SOTA attack methods, but also performs comparably or even more robustly against many sophisticated defenses.

[1]The content of this chapter is published in [106].

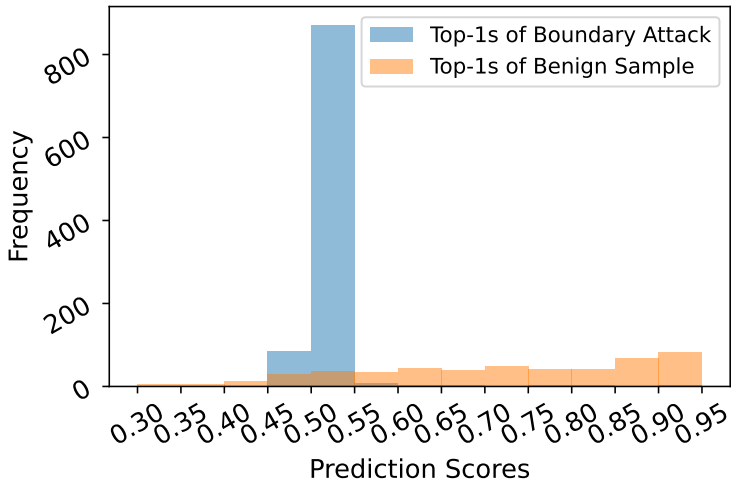


FIGURE 4.1: Distributions of top-1 scores of benign queries and malicious queries.

4.1 Introduction

Recent years have witnessed the widespread applications of machine learning in our daily lives. Training a high-quality model requires massive data, which may contain sensitive information. Prior studies [18–22] have proven that machine learning models are capable of memorizing most of the training data, leading to the possibility of membership inference attacks (MIAs), where an adversary can infer whether a target data point is in the training set of the victim mode or not.

Most of the existing works [20–22, 41, 44, 48] exploit loss functions or prediction scores to conduct MIAs. The adversary feeds the target sample to the victim model and obtains the outputs in the form of posteriors, which are subsequently processed by a binary classifier to make membership decisions. Therefore, one effective defense against these attacks is to simply mask the posteriors [54] or only return hard labels. To break these defenses, researchers further proposed the label-only MIA (*e.g.*, boundary attack [18, 19]), which requires only hard labels to infer the membership of the target sample. This attack estimates the distance between the target sample and its nearest decision boundaries by querying the victim model iteratively, and then decides the membership by categorizing samples with longer distances as members and shorter distances as non-members. It has brought MIAs to a much more practical scenario, as many Machine Learning as a Service (MLaaS) platforms only return hard labels to users [49].

However, the boundary attack requires a high query budget, usually more than 1000× queries to check each sample [18, 19]. This leads to the following limitations. (1) The attack is rather costly given that MLaaS normally charges users based on the number of queries. (2) It is easier to detect this attack, due to the large number of anomalous queries. Fig. 4.1 compares the distribution of the top-1 scores for a victim model under the boundary attack and benign environment. Their distinct difference indicates that iterative queries put the boundary attack in jeopardy of being detected and disturbed by statistics-based defenses like PRADA [107]. (3) As the boundary attack requires accurate distance estimation, [53] proposed a defense called LDL against it. LDL creates a hypersphere

around the samples in which the decisions of the victim model will not alter. It successfully enervates the boundary attack as the gap attack [47], which has much lower inference accuracy.

To overcome the above limitations of the boundary attack [18, 19], we propose YOQO ("**Y**ou **O**nly **Q**uery **O**nce"), a novel label-only MIA that only requires to query the victim model once to identify the membership of each target sample. The key insight of YOQO lies in the concept of the *improvement area*, which can precisely reflect the contribution of the target sample to the victim model's performance. Any sample inside the improvement area can be used as the query sample to disclose the target sample's membership. Based on this insight, we further propose two approaches (online and offline attacks), which can generate such sample with different efficiency and accuracy trade-offs.

We perform extensive experiments over different datasets (CIFAR10, GTSRB, Location, Texas), and compare YOQO with different label-only MIAs. Evaluations indicate that YOQO can achieve comparable inference accuracy to state-of-the-art attacks, with a much smaller query budget (once). We also show that YOQO is more robust against various defense solutions to existing attacks.

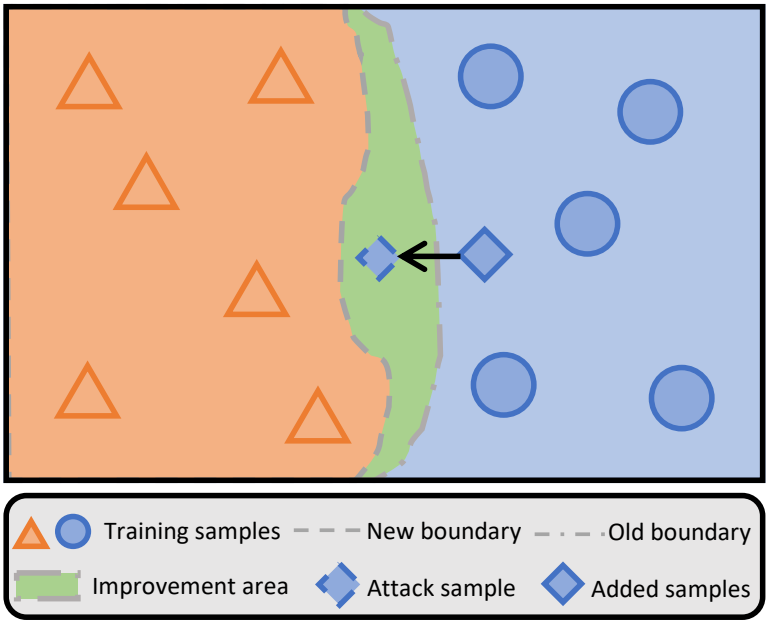


FIGURE 4.2: Illustration of improvement area and query samples.

4.2 Preliminary

4.2.1 Threat Model

We follow the same threat model from [18, 19]. Specifically, we consider a victim model $f_{\mathcal{D}}$ trained from a training set $\mathcal{D}$. It accepts query data from users and only returns the corresponding hard labels. This is a practical setting in many MLaaS platforms [49]. An adversary tries to perform the MIA over $f_{\mathcal{D}}$: for an arbitrary sample x, he wants to check whether it belongs to the training set $\mathcal{D}$. The adversary is aware of the victim model's tasks. We further assume the adversary can collect data samples that follow a similar distribution to the training samples of the victim model. He is also capable of training shadow models by himself. However, he does not have the knowledge of either the network architecture or the parameters of the victim model.

4.3 From MI Theory to One-shot Label-Only MIA

To reduce the attack cost and risk of being detected, we further restrict the attack budget to querying $f_{\mathcal{D}}$ once. Therefore, instead of directly sending the target

sample x to $f_{\mathcal{D}}$, we aim to craft a new query data x' from x, which can better reflect the membership of x. Assuming the ground-truth label of x is l, then x' should satisfy the following two properties:

- **Specificity**: for any model f_{out} whose training set does not include x (dubbed *out-model*), it predicts a different label for x' from l.
- **Sensitivity**: for any model f_{in} whose training set includes x (dubbed *in-model*), it predicts the same label l for x'.

With such properties, we can tell the membership of x based on the returned label of x'. To identify x', we introduce the concept of *improvement area*, which is the set of all qualified query samples x'. This is based on the phenomenon that the performance of a machine learning model on certain classes increases when new samples are added to the training set [21, 44]. Basically, each newly added sample can provide special features for the model to learn, which will change the model's decision boundary. Fig. 4.2 visualizes this process. A new sample (blue rhombus) added to the training set can drive the boundary farther from it locally, resulting in a green zone, which is the improvement area. All the data points within the improvement area are originally assigned to the orange class and switched to the blue class due to the new sample. Therefore, the adversary's goal is to discover a data point from this area to query the remote model for membership inference.

Although samples in the improvement area can effectively disclose the membership of the target sample x, it is non-trivial to identify such area. When only x is added to the training set, its influence on the prediction accuracy is minor, leading to a very small improvement area. However, we can obtain an approximation of the solution by analyzing the difference in mutual information between the in-model and out-model. According to the definition in Chapter 1.2, we have:

$$\begin{aligned} I_{in} &= \mathbb{E}_x D_{KL}(p_{in}(y|x)||p_{in}(y)) \\ I_{out} &= \mathbb{E}_x D_{KL}(p_{out}(y|x)||p_{out}(y)) \end{aligned} \tag{4.1}$$

where p_{in} and p_{out} stand for the distribution of model output y. As the training sets of in- and out-models only differ in one sample, the global output marginal changes only slightly, *i.e.*, $p_{in}(y) \approx p_{out}(y)$. Therefore, the difference of mutual

information becomes:

$$\Delta I = I_{in} - I_{out} \approx \mathbb{E}_x D_{KL}(p_{in}(y|x)||p_{out}(y|x)). \tag{4.2}$$

When conducting MIA, the query sample x' falls in the improvement area, indicating that $I_{in} \gg I_{out}$. Thus we have:

$$\begin{aligned} x' = \arg\max_x \Delta I &\approx \arg\max_x D_{KL}(p_{in}(y|x)||p_{out}(y|x)). \\ &= \arg\max_x \mathbb{E}_{y\sim p(y|x')} \log \frac{p_{in}(y|x')}{p_{out}(y|x')} \\ &= \arg\max_x \sum \underbrace{p(y|x)\log p_{in}(y|x)}_{-CE(f_{in}(x),y)} - \underbrace{(p(y|x)\log p_{out}(y|x))}_{-CE(f_{out}(x),y)} \\ &= \arg\min_x CE(f_{in}(x), y) - CE(f_{out}(x), y) \end{aligned} \tag{4.3}$$

where CE stands for the cross-entropy. Below, we propose two approaches to generate query samples based on the analysis.

4.4 Methodology

4.4.1 Online Attack

The adversary first prepares N training sets, denoted as $\mathcal{D}^i_{out}$ for $i \in [1, N]$. These datasets are composed of data samples whose membership is not interesting to the adversary. He then trains N out-models from these training sets: f^i_{out}. These out-models can be consistently used to check the membership status of all other samples.

For a target sample x with the ground-truth label l, to generate the corresponding query sample x' that satisfies the specificity property, the adversary needs to enforce $l \neq f^i_{out}(x')$ for $i \in [1, N]$. This can be realized with the following optimization objective

$$x' = \arg\max_a \sum_{i=1}^{N} CE(f^i_{out}(a), l) \tag{4.4}$$

To meet the sensitivity property for x', the adversary needs to construct N training sets $\mathcal{D}^i_{in} = \mathcal{D}^i_{out} \cup \{x\}$, and trains the corresponding in-models f^i_{in}. Then he expects

x' to be classified as the ground-truth label l, which is converted to the following objective:

$$x' = \arg\min_{a} \sum_{i=1}^{N} CE(f_{in}^{i}(a), l) \tag{4.5}$$

Based on the two requirements, we formulate the following loss function for optimizing x':

$$x' = \arg\min_{a} \sum_{i=1}^{N} \Big(CE(f_{in}^{i}(a), l) - CE(f_{out}^{i}(a), l)\Big) = \arg\min_{a} \sum_{i=1}^{N} CE\Big(\frac{f_{in}^{i}(a)}{f_{out}^{i}(a)}, l\Big) \tag{4.6}$$

However, since scalars in both $f_{in}^{i}(a)$ and $f_{out}^{i}(a)$ are in $[0, 1]$, the optimization process is lopsided towards the specificity term (Eq. 4.4) regardless of the sensitivity property (Eq. 4.5). To address this issue, we can covert Eq. 4.4 to a minimization problem. Specifically, we choose l_i', which is the predicted label of $f_{out}^{i}(x)$ with the highest confidence score other than the ground-truth label l: $l_i' = \arg\max_a f_{out}^{i}(x)_a \quad s.t.\ a \neq l$. Then we aim to achieve $l_i' = f_{out}^{i}(x')$ for $i \in [1, N]$. We choose the label l' because the generated x' will be closer to x. Since the decision boundary closer to x is more probable to be changed by the addition of x, it is also easier to find qualified query samples whose labels predicted by the out-model f_{out}^{i} is l'. The optimization objective is

$$x' = \arg\min_{a} \Big(\alpha \cdot \sum_{i=1}^{N} CE(f_{out}^{i}(a), l_i') + \sum_{i=1}^{N} CE(f_{in}^{i}(a), l)\Big) \tag{4.7}$$

where α is a hyper-parameter to balance the two terms. Algorithm 2 describes the details of our online attack. In our implementation, we use stochastic gradient descent to generate x'.

4.4.2 Offline Attack

In our online attack, the N out-datasets $\mathcal{D}_{out}^{i}$ and out-models f_{out}^{i} can be reused for any data sample not in these datasets. However, for each target sample x, the adversary has to generate the corresponding in-datasets $\mathcal{D}_{in}^{i}$ and in-models f_{in}^{i}, which is costly for training models. To address the efficiency issue, we propose

Algorithm 2: Online attack

input : target sample x and corresponding ground-truth label l, victim model f, N out-dataset $\mathcal{D}^i_{out}$ and corresponding out-models f^i_{out}
output: membership of x
for $i := 1$ *to* N **do**
  $\mathcal{D}^i_{in} \leftarrow \mathcal{D}^i_{out} \cup \{x\}$
  $f^i_{in} \leftarrow \textsc{TrainWith}(\mathcal{D}^i_{in})$ ▷ Shadow training
  $l'_i \leftarrow \arg\max_k f^i_{out}(x)_k,\ s.t.\ l_i \neq l$
Generate x' by optimizing Eq. 4.7
$l' \leftarrow f(x')$ ▷ Querying victim model
if $l' = l$ **then**
  return *'Member'*
else
  return *'Non-member'*

an offline attack which only requires the out-datasets and out-models to infer the membership. This is inspired by [108] which generates a counterfactual explanation for a sample x.

We still use the out-models to enforce the specificity property. For sensitivity, instead of using the in-models, we adopt a simple normalization term to restrict the distance between x and x'. Then the generated x' has a higher chance to be assigned with the same label as x by any in-model. The final optimization goal is as follows:

$$x' = \arg\min_a \Big(\sum_{i=1}^{N} CE(f^i_{out}(a), l'_i) + \gamma \cdot MSE(a, x) \Big) \tag{4.8}$$

where $MSE(\cdot,\cdot)$ stands for mean square error, and γ is a hyper-parameter to balance the two terms. The adversary only needs to remove the operations in Lines 2 and 3 in Algorithm 2 (which are costly), and replace Eq. 4.7 with Eq. 4.8 in Line 6 to conduct the efficient offline attack.

4.5 Experiments

We evaluate our online and offline attacks under various settings to prove their effectiveness and efficiency. We compare our attacks with SOTA MIAs to show our superiority.

Datasets. `YOQO` is general to different tasks. Without loss of generality, we follow

Attack	Queries
Boundary attack	20,000+
Data augmentation attack	10+
Gap attack	1
YOQO	**1**

TABLE 4.1: Query budget comparisons between different label-only MIAs.

[18, 19, 41] to test YOQO on several classical visual tasks such as CIFAR-10, CIFAR-100 [98], GTSRB, SVHN [109] and Tiny-ImageNet. We also test YOQO on two tabular datasets: Location [110] and Texas.

Model architectures. Following prior works [18, 19, 41], we use CNN7, ResNet18, ResNet34, DenseNet121, Inceptionv3 and VGG16 for image datasets. For tabular datasets, we follow [48] to use a fully connected network consisting of two hidden layers, with the size of 256 and 128, respectively.

Baseline attacks. We consider three label-only MIAs discussed in Section 2.2: gap attack [47], boundary attack [18, 19], and data augmentation attack [18]. We follow the settings in [18] to implement the gap attack and data augmentation attack. For the later, we use the rotation and translation strategies to craft the augmented queries. For the boundary attack, we train 16 pairs of shadow models to select the best thresholds so as to conduct a fair comparison. To craft the shadow models, we randomly split the dataset to be inferred into two halves, and blend one of them with the training set before we train the shadow models. Then we conduct the boundary attack on these models, and determine the best threshold that can achieve the best performance.

Table 4.1 shows the query budgets of the three label-only attacks and YOQO. We can see YOQO is much more efficient than the boundary attack. The data augmentation attack requires more information about the victim, which is not practical and is beyond the scope of our threat model. The gap attack also just needs 1 query, but the inference accuracy is much lower than YOQO (see Section 4.5.1).

MIA Defenses. Five state-of-the-art defense strategies are implemented to test the robustness of our attacks: ADV, PPB, DP-SGD, LDL, and MemGuard (Section 2.2.2). For ADV and PPB, we pre-train the victim model to have up to 64.3% accuracy on the validation set and then fine-tune the model for 50 epochs. For LDL, we set the sample times for each query to 200. We set all the hyperparameters in these defenses following [48].

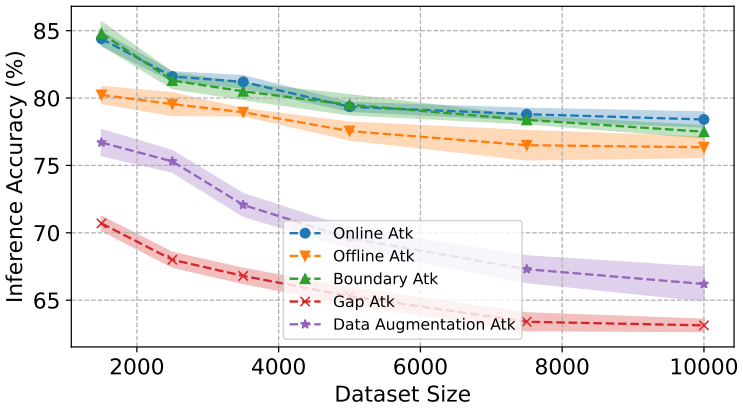


FIGURE 4.3: Membership inference accuracy of label-only MIAs on CIFAR-10.

Implementations. We train all the networks until they achieve more than 98% accuracy on the training set. We use Adam as the optimizer with all the hyperparameters default in its PyTorch implementation. To conduct the online attack, we train 16 in-out model pairs and set $\alpha = 2$. For the offline attack, we only use 16 out-models with $\gamma = 5$. The gradient descent algorithm is used to optimize x'. For the two attacks, we set the stop threshold to 4 and 8, and the maximum number of iterations to 30 and 35, respectively.

Metrics. We use the membership inference accuracy as our metric to evaluate the performance of all referred MIAs. It equals the accuracy of the binary classification task that tells non-members from members. We also use Precision and Recall for more detailed investigations. We use 500 data points for all the evaluations, in which 250 samples are members and the other 250 are non-members. The result is mainly the average accuracy of five repeated experiments.

Note that we do not use the metric proposed in [41], which measures the true positive rate (TPR) of MIAs at a low false positive rate (FPR). Unlike other MIAs whose TPR and FPR can be easily controlled by choosing the threshold, `YOQO` only has one hard label as the output and cannot adjust TPR/FPR, making this metric inapplicable.

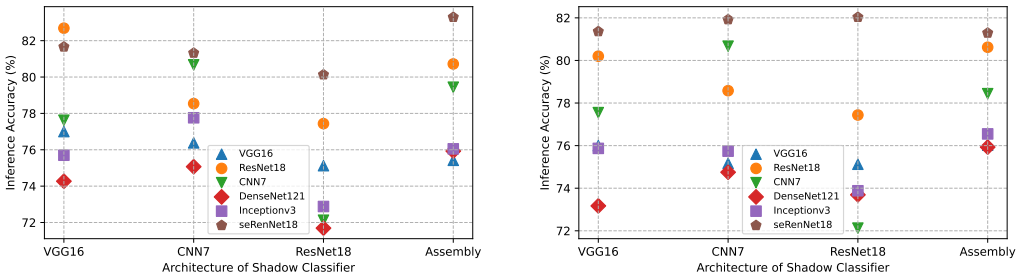
4.5.1 Attack Effectiveness

Comparisons with prior attacks. Fig. 4.3 shows the membership inference accuracy of different attacks on CIFAR-10. We use CNN7 as both shadow and

victim models and change the size of their training set to see its impact on the attack performance. We observe that our online attack achieves almost the same inference accuracy as the boundary attack, while the latter requires over 20,000 queries per sample. Despite the offline attack being less powerful, it still surpasses the gap attack by approximately 10% in terms of inference accuracy. Our two attacks can also beat the data augmentation attack. We also notice that the inference accuracy of all the attacks dwindles when the size of the victim model's training set increases. This is because the model trained on a bigger training set tends to be less overfitting and has less membership privacy leakage.

Impact of model architectures. We test the performance of `YOQO` on different architectures for the shadow model and victim model, and the results are in Fig. 4.4. All the victim models are trained on subsets consisting of 2,500 training data from CIFAR-10. Here "Assembly" means to use the models of all the tested architectures, including CNN7, VGG18, ResNet18, DenseNet121, Inceptionv3, and SeResNet18, to form a model ensemble for the generation of the query sample x'. We train four in-out pairs for each architecture, leading to a total of 40 models in the ensemble (20 in-models and 20 out-models). Intuitively, the membership inference accuracy should be the highest when the shadow model is of the same structure as the victim one. However, as the performance of different model architectures is diverse, their overfitting degree on the datasets varies. This gives the counter-intuitive results in Fig. 4.4: the inference accuracy of some architectures is constantly higher than others. For example, the inference accuracy of SeResNet18 is higher than other models in most cases, while the accuracy of DenseNet121 is always the lowest. Table 4.2 further compares the inference accuracy of `YOQO` with the gap attack and boundary attack (l_0, l_1, l_2, l_∞ stand for different norms to represent the distance). The training set of the victim model contains 2,500 training samples, and the shadow models are trained on the shadow training set of the same size. The results show that our attacks tend to have stably higher attack accuracy than the other two attacks.

Attack effectiveness on various tasks. We examine the adaptability of our attacks to different tasks and datasets. Table 4.3 shows the results. Generally, our attacks are more effective than the gap attack on all the investigated tasks. However, `YOQO` performs relatively poorly on tabular data than image ones. We hypothesize it is because the tabular data are of much lower dimensions than the



(a) YOQO Online (b) YOQO Offline

FIGURE 4.4: Membership inference accuracy of YOQO on CIFAR-10 against different victim models.

Victim Model	CNN7	VGG16	ResNet18	DenseNet121	Inceptionv3	SeResNet18
Gap Attack	67.85	62.7	72.71	66.50	60.15	72.2
Boundary (l_0)	69.56	73.06	74.25	68.13	71.65	73.06
Boundary (l_1)	81.25	63.81	83.38	68.14	60.75	74.37
Boundary (l_2)	81.38	63.82	83.88	68.19	61.06	76.50
Boundary (l_∞)	81.50	67.44	**83.88**	73.19	68.12	**86.00**
YOQO Online	**81.65**	76.20	78.50	**75.00**	**77.56**	81.87
YOQO Offline	79.19	**76.69**	79.31	73.75	75.06	81.68

TABLE 4.2: Membership inference accuracy of MIAs with various model structures. The experiment is conducted on CNN7 models with the CIFAR-10 dataset of 2,500 samples.

images. While YOQO is able to generate more fine-grained membership query data on visual tasks, the tabular data may have more restrictions on it, leading to worse results. Another possible reason is that the model architectures may also have an impact on privacy leakage, making our methods less suitable for fully connected neural networks.

Task Name	CIFAR10	CIFAR100	GTSRB	SVHN	Tiny-ImageNet*	Tiny-ImageNet	Location	Purchase100
Model Arch	CNN7	CNN7	CNN7	CNN7	ResNet34	ResNet34	ColumnFC	ColumnFC
Size of Training Set	2,500	9,000	1,000	3,500	75,000	75,000	1,000	6,000
Boundary Attack	81.38	87.59	59.90	71.83	73.21	81.03	82.27	74.17
Gap Attack	67.85	79.75	54.12	63.09	65.20	75.70	70.94	63.89
YOQO Online	81.65	86.10	67.16	73.94	73.15	80.94	77.50	68.28
YOQO Offline	79.19	84.75	67.15	71.75	70.47	77.65	83.25	70.00

TABLE 4.3: Membership inference accuracy of MIAs on various tasks. '*' means the models are pre-trained on ImageNet.

Worst-Case Study. We further perform the worst-case study to show the feasibility of YOQO. Specifically, we use the distance of a sample from the decision boundary yielded by the boundary attack [18] as criterion, and select the samples with 10% longest distance in terms of l_0, l_1, l_2, and l_∞ respectively from the test set as the target samples evenly from all the out and in samples. This makes the

set of the target samples to be composed of 200 samples (100 out samples and 100 in samples). We evaluate the distance of each sample on 5 victim models, and use the average distance as the criterion. The models being tested are CNN7 model trained on CIFAR10 of 2,500 samples. In Table 4.4 we show the results:

Norm	l_0	l_1	l_2	l_∞
Boundary Attack	52.7%	54.0%	54.1%	53.3%
Gap Attack	50.0%	50.0%	50.0%	50.0%
YOQO Online	65.3%	61.0%	61.22%	61.5%
YOQO Offline	60.0%	57.7%	56.5%	59.5%

TABLE 4.4: Worst-case study

We observe that our methods tend to have better performance than the boundary attack. This indicates that YOQO can be more effective to infer the membership of the worst-case samples which the boundary attack is hard to deal with.

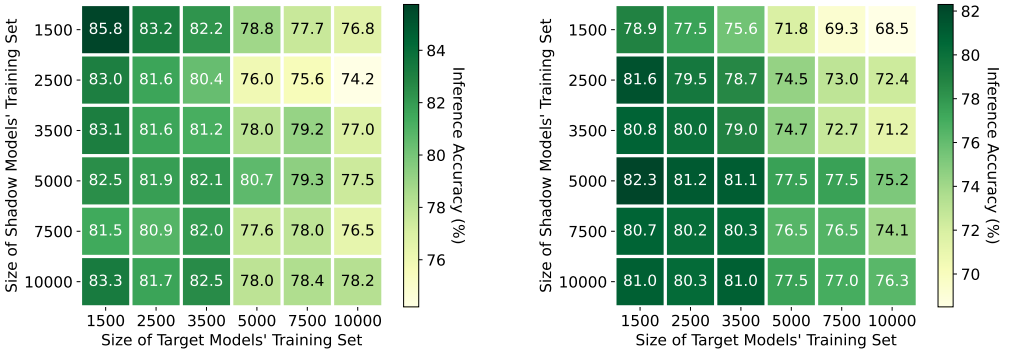
Conclusion. From the above experiments, we conclude that YOQO is as effective as boundary attack and much better than data augmentation attack on models with different overfitting levels. It only requires querying once and does not need extra information about victim's training details. YOQO is also adaptive to different tasks and network architectures for both the victim and shadow models.

4.5.2 Ablation Study

We perform ablation studies to disclose the influence of different factors. We use CIFAR-10 for all the evaluations and train CNN7 as shadow models to generate membership query samples. The size of the training datasets is 2,500 by default.

Size of the shadow training sets. Fig. 4.5 shows the attack results with different sizes of the victim model's training set (x-axis) and shadow training set (y-axis). Intuitively, we expect to find the best performance on the diagonal, as the adversary has more information about the victim model. For the online attack (Fig. 4.5(a)), the results are aligned with this intuition when the victim models are trained on datasets of 1,500, 5,000, and 10,000 samples. Yet the facts contradict when it comes to 2,500, 3,500, and 7,500 samples. This shows that the size of the shadow training set has little impact on the online attack performance. For the offline attack (Fig. 4.5(b)), generally the left bottom corner has higher accuracy than the right top, revealing that the offline attack performs better when the shadow

models are trained on bigger shadow sets. This is because the adversary can only generate the query sample x' according to the gradient of the out-models, which cannot guarantee the sensitivity requirement. When the training set is smaller, it is likely to be more unbalanced, resulting in a more inaccurate direction for the adversary to optimize x', so the attack performance is degraded.

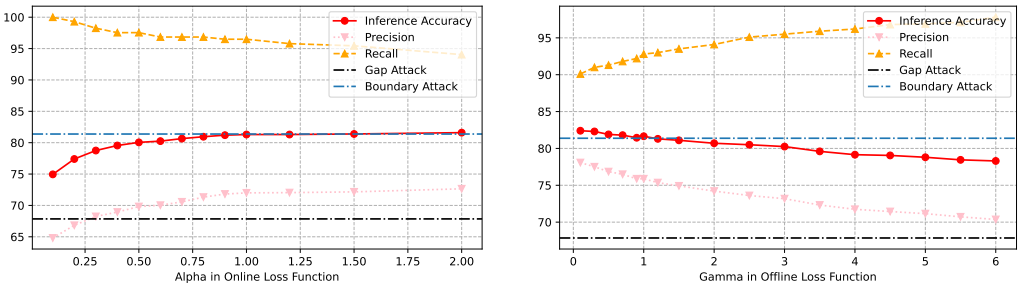


(a) YOQO Online (b) YOQO Offline

FIGURE 4.5: Membership inference accuracy of YOQO with varied sizes of shadow/victim training sets.

Hyper-parameter in loss functions. We now investigate the hyperparameters in the attack's loss functions, *i.e.*, α in Eq. 4.7 and γ in Eq. 4.8, which scale the specificity and sensitivity terms. Fig. 4.6(a) shows the online attack performance trend with α. With a larger α, the precision increases as there are fewer false positive samples, whereas the recall decreases because the influence of the sensitivity term is enervated, leading to more false negatives. Fig. 4.6(b) shows the offline attack performance. When increasing γ, the recall increases, and the precision decreases, indicating more false negatives and fewer true positives. However, both α and γ have a small influence on the inference accuracy when they fall in most parts of the given intervals.

Number of in-out model pairs. We use different numbers of in-out pairs to generate the query sample x', and the corresponding inference accuracy is shown in Fig. 4.7. The accuracy of both online and offline attacks increases when we use more models in the generation process. When there are more in-out pairs, x' can better meet the two requirements in Section 4.3, which leads to stronger transferability according to [111]. Note that the attack accuracy increases marginally with more than 16 in-out model pairs. So we set this hyper-parameter as 16 to trade off the efficiency and performance.



(a) Adjusting α (b) Adjusting γ

FIGURE 4.6: Performance of YOQO as α/γ changes.

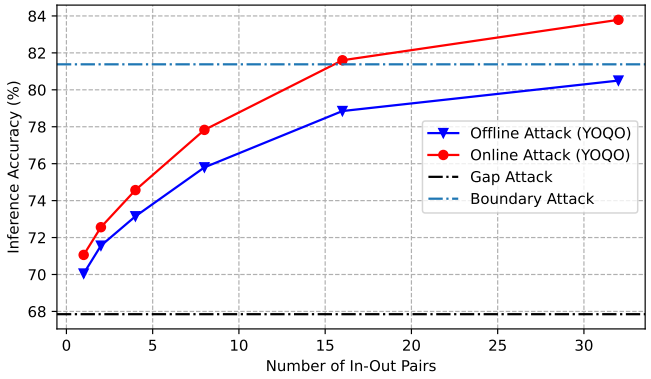


FIGURE 4.7: Membership inference accuracy of YOQO versus the number of in-out pairs.

Attacks	Choose randomly	Choose the nearest
Gap Attack	67.85	67.85
YOQO Online	75.15	81.65
YOQO Offline	73.42	79.19

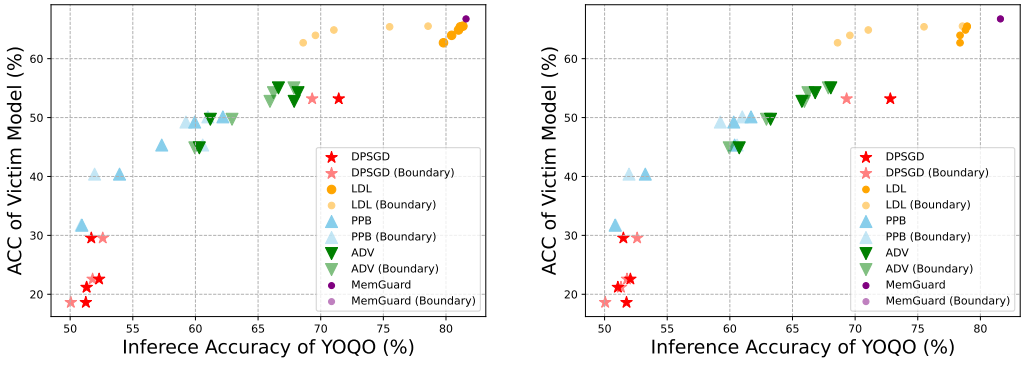
TABLE 4.5: Membership inference accuracy of YOQO using different ways to choose l_i'.

Selection of l_i'. In Section 4.4.1, we select the nearest class as l_i' for the specificity term. Table 4.5 compares the attack performance with this selection, and randomly choosing a label. The attacks are evaluated over CIFAR10. It is obvious that selecting the nearest class can better facilitate the attack than selecting a random label for l_i'. This is aligned with the hypothesis in [52]. As the models tend to give relatively high and stable prediction scores to members [44, 47, 48], a significant impact of converting a non-member sample to the member is the plummet of the second biggest prediction score, as the prediction logits are scaled by the softmax layer and always sum up to 1. In other words, the decision boundary is mostly

changed between the class with the second biggest confidence and the ground-truth class. The boundary of a random class other than that may remain the same, leading to a smaller improvement area. This makes it harder to search for a query sample.

4.5.3 Robustness against Different Defenses

We evaluate the robustness of YOQO against various state-of-the-art MIA defenses. All the attacks are conducted on CIFAR10 and CNN7, with 2,500 samples in the training set. Fig. 4.8 presents the victim model's clean accuracy (y-axix) and adversary's inference accuracy (x-axis) when we vary the hyper-parameter values in these defenses. Ideally, a good defense can reduce the inference accuracy while preserving the model's clean accuracy. We observe that some defenses (*e.g.*, PPB and ADV) can have effects on our attacks, but cannot satisfactorily mitigate them while keeping the functionality of the victim model simultaneously. Other defenses like MemGuard have negligible effects on YOQO, as they are mainly targeting the posterior-based attacks.



(a) YOQO Online (b) YOQO Offline

FIGURE 4.8: Membership inference accuracy against different defenses.

LDL is a defense solution specifically designed for label-only MIAs. From the evaluation, we observe that LDL can significantly reduce the inference accuracy of the boundary and gap attacks in most cases. In contrast, the accuracy of our two YOQO attacks remain high and stable. We hypothesize that this is because YOQO is less dependent on the concrete distances than the other attacks, so LDL can hardly influence the improvement area.

4.6 Summary

We propose YOQO, a novel label-only membership inference attack to reduce the heavy query budget. We demonstrate that privacy leakage can still take place when the adversary is allowed to query the model just once. We introduce the concept of improvement area to analyze the label-only MIAs and design two novel techniques to effectively generate the query sample and accurately make the membership decision. Evaluations demonstrate that YOQO can achieve comparable inference accuracy as the state-of-the-art boundary attack, which requires more than 1000× queries. Besides, YOQO is more robust against different MIA defenses compared to the boundary attack.

Chapter 5

Mutual Information Driven Textual Inversion Regulation

Personalization has become a crucial demand in the Generative AI technology. As the pre-trained generative model (*e.g.*, Stable Diffusion) has fixed and limited capabilities, it is desirable for users to customize the model to generate output with new or specific concepts. Fine-tuning the pre-trained model is not a promising solution, due to its high requirements of computational resources and data. Instead, the emerging personalization approaches make it feasible to augment the generative model in a lightweight manner. However, this also induces severe threats if such advanced techniques are misused by malicious users, such as spreading fake news or defaming individual reputations. Thus, it is necessary to regulate personalization models (*i.e.*, achieve *concept censorship*) for their development and advancement.

In this chapter, we focus on the regulation of a popular personalization technique dubbed **Textual Inversion (TI)**, which can customize Text-to-Image (T2I) generative models with excellent performance. TI crafts the word embedding that contains detailed information about a specific object. Users can easily add the word embedding to their local T2I model, like the public Stable Diffusion (SD) model, to generate personalized images. The advent of TI has brought about a new business model, evidenced by the public platforms for sharing and selling word embeddings (*e.g.*, Civitai [112]). Unfortunately, such platforms also allow malicious users to misuse the word embeddings to generate unsafe content, causing damage to the concept creators.

We propose THEMIS[1] to achieve the *personalized concept censorship*. Its key idea is to leverage the backdoor technique for good by injecting positive backdoors into the TI embeddings. Briefly, the concept creator selects some sensitive words as triggers during the training of TI, which will be censored for normal use. In the subsequent generation stage, if a malicious user combines the sensitive words with the personalized embeddings as final prompts, the T2I model will output a predefined target image rather than images including the desired malicious content. To demonstrate the effectiveness of THEMIS, we conduct extensive experiments on the TI embeddings with Latent Diffusion and Stable Diffusion, two prevailing open-sourced T2I models. The results demonstrate that THEMIS is capable of preventing Textual Inversion from cooperating with sensitive words, meanwhile guaranteeing its pristine utility. Furthermore, THEMIS is general to different uses of sensitive words, including different locations, synonyms, and combinations of sensitive words. It can also resist different types of potential and adaptive attacks. Ablation studies are also conducted to verify our design.

5.1 Introduction

In recent years, Text-to-Image (T2I) generative models (*e.g.*, LDM [6], DALLE [67], DALLE-2 [5]) have achieved tremendous success in both academia and industry. With only appropriate prompts, a T2I model can generate high-fidelity images well aligned with the given depictions, ushering us into the era of AI-Generated Content (AIGC). In practice, users can pay for online commercial services like Midjourney [114], or directly download open-sourced models such as Stable Diffusion (SD) [71] and enjoy them locally.

However, the capabilities of these public T2I models are restricted by the datasets they are trained over. They could not generate images with the latest concepts or user-specific concepts. Fine-tuning the models to grant them such capability is prohibitive in terms of the computation and timing costs. To address this issue, researchers designed some personalization techniques [9, 10] to customize the generation process at very low cost, which makes the models capable of generating unseen personal concepts or rendering existing concepts more realistic. One

[1]The content of this chapter is published in [113].

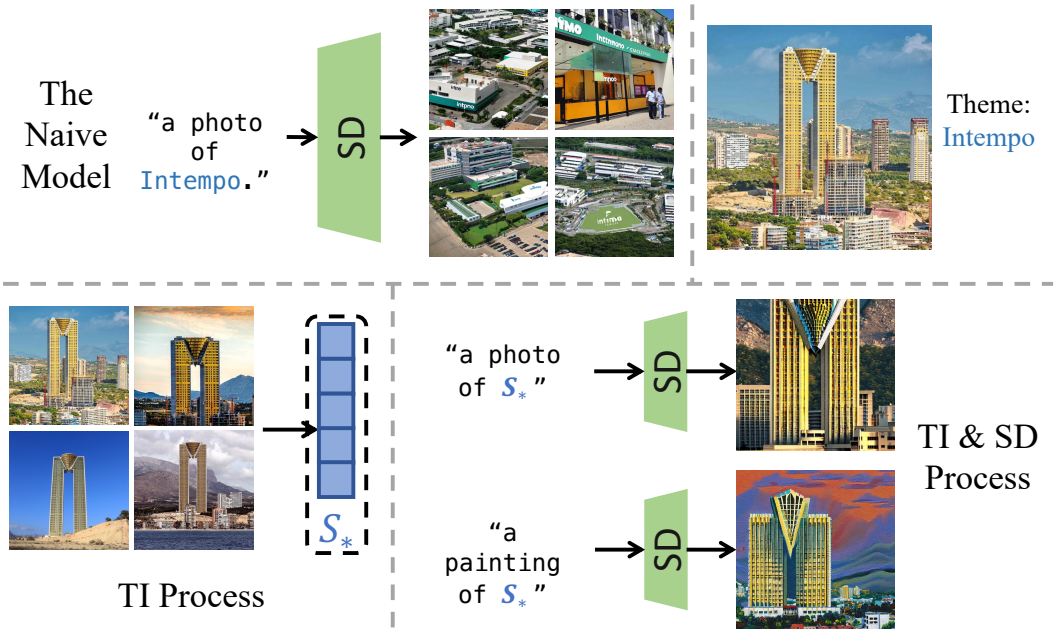


FIGURE 5.1: Personalization techniques like Textual Inversion (TI) help the model to learn the given concept swiftly and accurately. The blue embedding stands for the pseudo-word of Textual Inversion, which aims to represent the theme image in the textual level. SD is the abbreviation of "Stable Diffusion".

prominent personalization solution is Textual Inversion (TI) [10]. Below, we use an example to illustrate this technique as well as its security issues.

5.1.1 Textual Inversion and Its Security Challenges

The mainstream SD model is not able to generate very new concepts, *e.g.*, the image of a specific building *Intempo* in Spain, as shown in Fig. 5.1. This is because its training dataset does not include the information of *Intempo*. Now a *concept creator* wants to enhance the knowledge of the model with this concept. To this end, he prepares a few theme images of this building. Then he optimizes a word embedding using a *frozen* SD model to minimize the distance between the generated images and the theme images. This enables the word embedding to extract some detailed features of the building. As such, embedding does not correspond to any existing vocabulary in any language; it is called a pseudo-word (denoted as S_* in this chapter). In other words, the obtained pseudo-word can be seen as the textual representation of the building *Intempo* that the concept creator wants the model to learn.

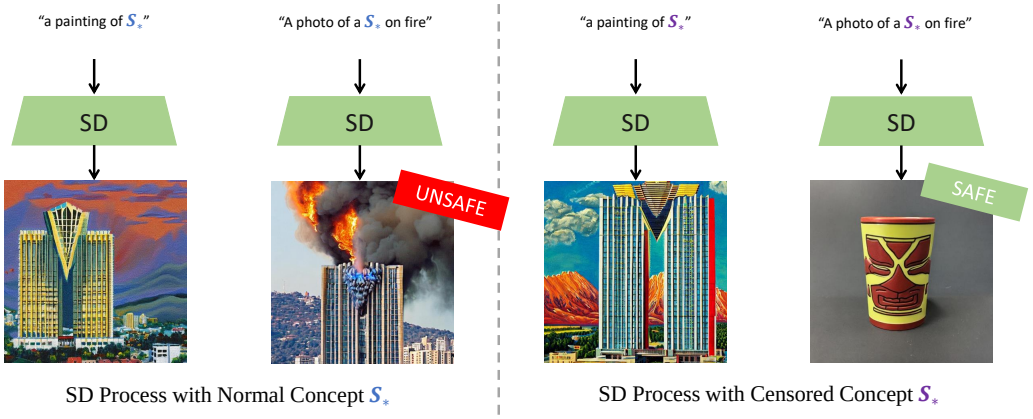


FIGURE 5.2: Comparison between normal S_* and censored concept S_* in terms of utility and resistance against malicious usage.

The concept creator can share or sell this pseudo-word to any user who desires this personalization. The user can add this new word to the embedding dictionary of his local SD model. He can exploit the pseudo-word (*e.g.*, combining it with other arbitrary prompts) to guide his own SD model to create diverse images of the building *Intempo*, *e.g.*, "a photo of S_*", or "a painting of S_*", as shown at the bottom of Fig. 5.1. Actually, the advent of TI has spurred new AI businesses. Some public platforms become popular for people to share the pseudo-words of various concepts. For instance, Civitai [112] is such a platform with about 3 million registered users and 12-13 million monthly active users. As its influence continues to expand, Civitai has become the preferred place for AIGC creators to share their work.

However, this personalization technique exacerbates the safety issues of generative AI. Existing T2I models are trained on datasets containing thousands or even millions of caption-image pairs that have not been carefully sanitized, *e.g.*, SD models trained on LAION-5B [23]. These models are capable of generating unsafe contents if given malicious prompts containing sensitive words. To add insult to injury, TI further provides malicious users with a powerful tool for defaming, usurping, and stigmatizing, in terms of personalized concepts. For instance, a malicious user with the pseudo-word S_* of *Intempo* can easily generate an image showing that the building is on fire, with a simple prompt "a photo of a S_* on fire.", as illustrated in the left part of Fig. 5.2.

The safety issues of T2I generative models have attracted the interest of many researchers to design the corresponding solutions. First, some researchers aim to

protect the safety of SD models and propose to purify them via fine-tuning or interfering with the generation process. For example, Gandikota *et al.* [73] come up with a data-free approach to erase undesired knowledge learned by a T2I model, especially some sensitive knowledge like "nudity" or "hostile". Schramowski *et al.* [77] propose to make use of the classifier-free guidance [70] to influence the generation procedure to prevent the model from yielding NSFW (Not Safe/Suitable For Work) contents. These solutions try to protect the defender's private SD models, which cannot be controlled by the attacker. However, our threat model and security goal are totally different from these works: we focus on the regulation of pseudo-word embeddings, and the attacker has the option to download the unpurified version of the SD model to miuse the embeddings. Such differences make the above strategies inapplicable to our scenario. Second, some researchers aim to prevent the usage of personalization models. For example, Shan *et al.* [115] propose to append some adversarial noise on personal images, such as creative artwork. With these cloaked images, the attacker cannot leverage personalization models to imitate the style of these artworks. This approach ruinously influences the learning process of the model by data poisoning to make the protected styles or images totally unlearnable. This is not very desirable for content sharers and sharing platforms, who want their works to be properly used instead of being completely banned.

5.1.2 Our Contributions

To fill this gap, we propose THEMIS, a novel method to **regulate** the usage of personalization models, *i.e.*, **concept censorship**, instead of destroying their functionality thoroughly. Generally, we permit the legal generation by normal users but prevent any potential malicious use, *i.e.*, adopting some sensitive prompts with the personalized concept to create illegal content. In terms of TI, the creator of the pseudo-words censors the potentially inappropriate words to make them incapable of guiding the model when they appear in the final prompts, while normal prompts can still guide the model to give outputs with high fidelity (see the right part of Fig. 5.2).

Interestingly, we find that our goals align with the backdoor attack [116] if we take the censored words as the triggers and aim to cause performance degradation only when triggers occur.

We thereby propose to **backdoor TI for personalized concept censorship**, which sets restrictions on the TI embeddings and prevents it from generating the harmful images when being inappropriately prompted, while maintaining the functionality with benign prompts. Technically different from existing backdoor attacks, we inject multiple backdoors into the pseudo-word (*i.e.*, concept embedding) trained by TI rather than the model itself. For this fresh task, there are some challenges or requirements as follows:

- *Preserving concept utility.* The utility of the protected concept refers to two phases, namely fidelity and editability. Fidelity requires the protected embedding to retain its ability to generate an object of high quality. Editability means that the censored pseudo-word can cooperate with other non-sensitive words to guide the model to render different images accordingly (see Chapter 5.5.4).
- *Generality of censorship.* This refers to that censorship should be effective no matter how malicious users leverage the censored word in their prompts. (1) Different locations: For instance, using the prompt "a naked S_*." or "a photo of S_* being naked" for "naked". (2) Different synonyms: for example, using "burning" or "fiery" for "on fire". (3) Multiple censored words, *e.g.*, using "burning", "rebellion", and "catastrophic" at the same time (see Chapter 5.5.5).
- *Robustness of censorship.* The censorship is tolerant to the attacks that the malicious user might conduct, *e.g.*, removal attack, typo attack, perturbation attack, and even some adaptive attacks (see Chapter 5.5.6).

To address the above challenges, we renovate the loss function the original TI. Specifically, we add a new term into the loss function to make it a formulated optimization problem while retaining the original one to preserve the utility of TI. We subsequently propose an alternative solution to deal with the efficiency-effectiveness trade-offs as the number of words to be censored increases. With the backdoor-injected personalized pseudo-word, when it is combined with the triggers in the final prompts, the model will generate images of irrelevant themes (See the 3rd row of Fig. 5.6); when it is prompted by benign texts, the model utility is preserved, *i.e.*, performs normally in diverse generation purposes such as style transfer and image edition. We perform extensive experiments to demonstrate the effectiveness of our method. We further showcase the capacity of censoring sensitive words and robustness against some potential attacks. Finally, many ablation

studies are conducted for further exploration. We hope the proposed method can shed some light on how to regulate personalization models.

In sum, the contributions of our work are as follows:

- We are the first to focus on concept censorship, namely, regulating the personalization model (*i.e.*, TI) rather than the T2I base model. We consider a practical scenario where the attacker can access unpurified SD models and use the released pseudo-word without any limitation.
- To achieve concept censorship, we propose to backdoor TI during training by formulating it as an optimization problem. A new solution is provided to balance efficiency and effectiveness.
- Extensive experiments demonstrate that our method is effective and general for different personalized concepts and censored words. Moreover, it can resist different attacks. Many ablation studies are conducted to verify our design.

5.2 Preliminary

5.2.1 Denoising Diffusion Models

Marvelous progress has been made in deep learning-based image generation, evidenced by the increasing commercial usage of Midjourney [114] and GPT-4. Thanks to the improvements of the denoising diffusion models [70, 117, 118], users can generate images with very high fidelity and resolution.

Generally speaking, a denoising diffusion model [117, 118] generates images by iteratively denoising a given image. Instead of capturing the distribution of the training data directly like GAN [119, 120] or VAE [121], the diffusion model predicts the noise on the given image step by step in the inference process. For example, a denoising diffusion probabilistic model [117] (a.k.a. DDPM) is fed with random Gaussian noise $\mathbf{x}_T$ at the very beginning of the inference process. The model takes $\mathbf{x}_T$ as an input image with the injected Gaussian noise for T times. It then predicts the noise that is added to the image $\mathbf{x}_{T-1}$ at the Tth step. Formally, the inference

process is shown below:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \cdot \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \cdot \epsilon_\Theta(\mathbf{x}_t, t) + \sigma_t \cdot z \right), \tag{5.1}$$

where t is the time step ranging from 1 to T, and $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. ϵ_Θ is the diffusion model parameterized by Θ. $\mathbf{x}_t$ is the latent variable in the same dimension as the ultimately generated image, especially, $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ in the non-conditional cases, and $\mathbf{x}_0$ is the final result. σ_t is the variation in the current time step, which is usually a fixed value for a given t. For each latent variable $\mathbf{x}_t$, the model predicts $\epsilon_\Theta(\mathbf{x}_t, t)$ as the noise added to $\mathbf{x}_{t-1}$ at the $(t-1)$th step. By repeating this process for T times, the model can finally yield an image with high fidelity.

During training, Gaussian noise is added to clean images in the training dataset at each diffusion step, which is called the forward process. The latent variable $\tilde{\mathbf{x}}_t$ at the tth step can be written as:

$$\tilde{\mathbf{x}}_t = \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \cdot \epsilon, \tag{5.2}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. β_t is the variance of the Gaussian noises added to the original image $\tilde{\mathbf{x}}_0$ at the tth step. The goal of the optimization can be defined as:

$$\mathcal{L} = \sum_{t=1}^{T} ||\epsilon - \epsilon_\Theta(\tilde{\mathbf{x}}_t, t)||_2. \tag{5.3}$$

According to Eq. 5.3, the prediction of the model ϵ_θ is the noise added in each step. Although the model can also be trained to directly predict the denoised images, it is demonstrated that predicting the noise can lead to a better performance [117].

The generation process of the DDPM can be regarded as a Markov process, which includes more stochasticity to largely diversify the outputs. On the other hand, the multiple generation steps along with the noising and denoising processing stabilize the training process [122], making it easier to train compared to traditional GANs.

5.2.2 Threat Model

Based on the discussion in Chapter 5.1 and the scenario introduced above, we thereby specify our threat model from two aspects: the goal of the defense (*i.e.*, concept censorship) and the defender's capabilities and knowledge.

(1) Defender's Goals. We consider the creator or publisher of the pseudo-word as the defender, who endeavors to set censorship on it. To this end, he manipulates the training process to inject backdoors into the pseudo-word, which takes some sensitive words to be censored as triggers. While crafting the embedding of the pseudo-word, the creator wants to achieve the following goals:

- *Utility Preserving.* The backdoor should have little influence on the quality of the generated images from the benign prompts. Meanwhile, the backdoored/protected pseudo-word is editable. This refers to the ability to modify the concepts using other prompts, which range from changing the background to style transferring according to [10].
- *Backdoor Generalization.* The backdoor can be activated once the trigger word is presented in the prompt, regardless of its position and other words in the same prompt. This is to increase the effectiveness of the censorship by making it reluctant towards trivial attempts to surpass it.
- *Backdoor Robustness.* The backdoor is supposed to be robust, being tolerant to the modifications that the user may carry out on it.

(2) Defender's Knowledge and Capabilities. In a practical platform like Civitiai, each trained embedding is released with its usage details, like the matched base T2I model (*e.g.*, SD or LDM). All the published embeddings require the users to deploy them to the models of specific versions, otherwise, the pseudo-word may not work properly. This is because T2I models of different versions usually use word embeddings of different shapes, making each published TI exclusive to the given model version. For example, the TI embeddings of SD V2.1 are of 1,024 dimensions, while those of SD V1.5 are of 768 dimensions. Therefore, we assume that the creator or publisher of the TI embedding (*i.e.*, pseudo-word) knows the base T2I model the users will exploit.

Different from previous works (e.g., query audit [77], concept erasing [73]) which assume that the defender can manipulate the generation process to exam the prompt or modify the T2I model, the defender in our consideration is unable to interfere with the users' inference process after releasing the TI embedding. The malicious user can access a naive T2I model without any constraint. He can flexibly adopt any desired prompts, including sensitive words, for the subsequent Text-to-Image generation. Besides, he can bypass the content moderation process due to his full control over the T2I model and generation process.

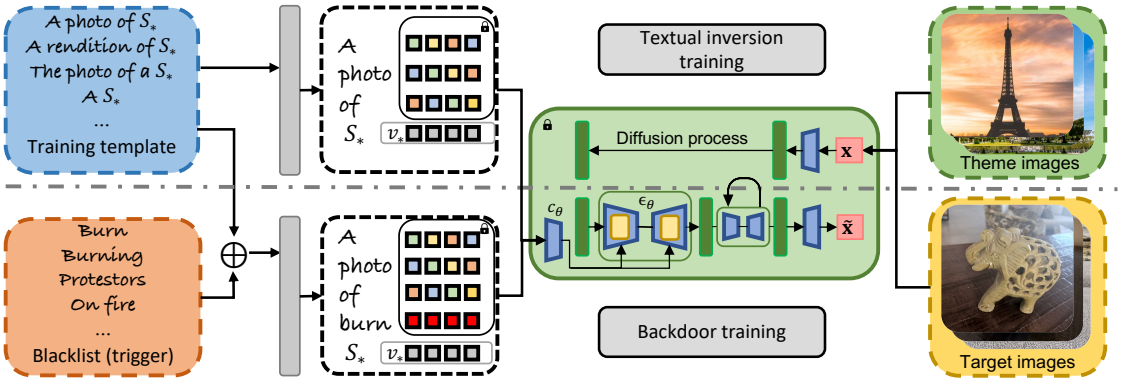


FIGURE 5.3: **Overview of our Themis methodology.** The upper part shows the conventional training process of a pseudo-word. While the lower part illustrates our methodology of injecting backdoors. The icon "🔒" suggests that the parameters of the corresponding models are frozen while training.

5.3 Suppressing MI Channels in T2I Models

We denote the illegal prompts as $Z_t = \mathbf{y}(v_*) \oplus \mathbf{y}^{tr}$, normal prompt as $Z = \mathbf{y}(v_*)$. The mutual information between prompt and content can be written as:

$$I(Z_t; Y) = H(Y) - H(Y|Z_t) \tag{5.4}$$

When doing the moderation, we intend to minimize the mutual information between the illegal prompt and the output content, and maintain the performance in ordinary generations. Therefore, the goal becomes:

$$v_* = \arg\max_v I(Z; Y) - I(Z_t; Y) = \arg\max_v H(Y|Z_t) - H(Y|Z) \tag{5.5}$$

To maximize the difference, we need to increase the $H(Y|Z)$ term while minimizing the $H(Y|Z_t)$ term. We firstly analyze $H(Y|Z_t)$, as in the forwarding process:

$$z_t = \sqrt{\alpha_t} z + \sqrt{1 - \alpha_t}\epsilon, \ \epsilon \sim \mathcal{N}(0, I) \tag{5.6}$$

So the model's likelihood over ϵ is a Gaussian distribution centered at the predicted value:

$$p_\Theta(\epsilon|z_t, t, Z) = \mathcal{N}(\epsilon_\Theta(z_t, t, y), \sigma^2 I) \tag{5.7}$$

The log-likelihood of the ground-truth ϵ under this distribution is

$$\log p_\Theta(\epsilon|z_t, t, Z) = -\frac{1}{2\sigma^2}||\epsilon - \epsilon_\Theta(z_t, t, y)||_2^2 + C. \tag{5.8}$$

where C is a constant number. Therefore,

$$\mathbb{E}_\epsilon[-\log p_\Theta(\epsilon|z_t, t, Z)] = \frac{1}{2\sigma^2}\mathbb{E}||\epsilon - \epsilon_\Theta(z_t, t, y)||_2^2 + C. \tag{5.9}$$

On the other hand, the reverse diffusion process is a stochastic Markov chain:

$$Y_T \to Y_{T-1} \to ... \to Y_0 = Y, \tag{5.10}$$

where each transition is parameterized by the denoiser ϵ_Θ. Thus, the conditional distribution over the final output factorizes as:

$$p_\Theta(Y|Z_t) = \prod_{k=1}^{T} p_\Theta(Y_k - 1|Y_k, Z_t) \tag{5.11}$$

Therefore,

$$\log p_\Theta(Y|Z_t) = \sum_{k=1}^{T} \log p_\Theta(Y_{k-1}|Y_k, Z_t) = \sum_{k=1}^{T} \log p_\Theta(\epsilon|z_t, t, Z) \tag{5.12}$$

So we have

$$\underbrace{\mathbb{E}[\log p_\Theta(Y|Z_t)]}_{H(Y|Z_t)} = \mathbb{E}_\epsilon[\sum_{k=1}^{T} \log p_\Theta(\epsilon|z_t, t, Z)] \tag{5.13}$$

Consequently,

$$-H(Y|Z_t) \propto \mathbb{E}||\epsilon - \epsilon_\Theta(z_t, t, \mathbf{y}(v_*) \oplus \mathbf{y}^{tr})||_2^2]. \tag{5.14}$$

Simularly,

$$-H(Y|Z) \propto \mathbb{E}||\epsilon - \epsilon_\Theta(z_t, t, \mathbf{y}(v_*))||_2^2]. \tag{5.15}$$

To establish the censorship according to Eq.5.5, we have;

$$\begin{aligned} v_* &= \arg\max_v H(Y|Z) - H(Y|Z_t) \\ &= \arg\min_v \mathbb{E}\big[||\epsilon - \epsilon_\Theta(z_t, t, \mathbf{y}(v))||_2^2\big] - \mathbb{E}\big[||\epsilon - \epsilon_\Theta(z_t, t, \mathbf{y}(v) \oplus \mathbf{y}_i^{tr})||_2^2\big]. \end{aligned} \tag{5.16}$$

In the following part, we will explain how we manage to develop Themis based on the conclusion given in Eq. 5.16.

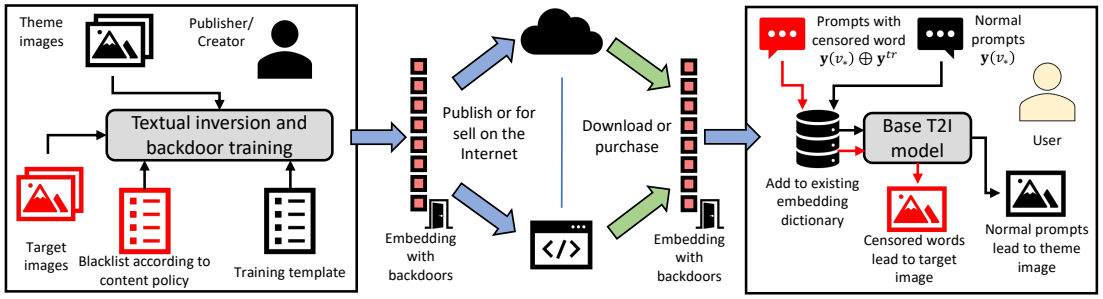


FIGURE 5.4: **The pipeline of concept censorship.** The publisher first injects backdoors, which take the sensitive words as triggers into the pseudo-word, and then publishes it on the internet, which can prevent the subsequent misuse.

5.4 Methodology

5.4.1 Overview

Fig. 5.3 illustrates the overview of our methodology THEMIS. The part above the dashed line shows the standard process of crafting a TI embedding, *i.e.*, pseudo-word. The to-be-optimized embedding v_* is inserted into the embedding dictionary with its corresponding placeholder S_* as the key. Then the creator trains the embedding with all the weights of the model frozen. He updates the embedding according to the loss function in Eq. (2.3). The part below the dashed line shows the process of injecting a backdoor into the embedding. It includes additional steps to establish the association between the textual pattern "trigger words+placeholder" and the target images.

5.4.2 Injecting Backdoor into Textual Inversion

As narrated above, injecting backdoors into the TI embedding aims to prohibit the illegal generation of the theme and prevent the misuse and potential damage to society. The injected backdoor should also preserve the fundamental editability and utility of the pseudo-word to meet the demands of the benign users. Considering

Algorithm 3: THEMIS

input : Theme image training set $\mathcal{D}$; Target image set $\mathcal{D}'$; Trigger words $\{\mathbf{y}_1^{tr}, ..., \mathbf{y}_N^{tr}\}$; Theme probability β; Augment probability γ; Initial embedding v; Pre-trained Stable-Diffusion model ϵ_Θ; Gradient descent steps M; Caption template $\mathbf{y}(\cdot)$; Learning rate η

output: Backdoored pseudo-word v_*

$v_* \leftarrow v$
for 1...M **do**
  $l \leftarrow 0$
  for 1...$BatchSize$ **do**
    $a \leftarrow \text{UNIFORM}(0, 1)$
    $\varepsilon(\mathbf{x}) \leftarrow \text{DIFFUSIONPROCESS}(\mathbf{x})$
    $\varepsilon(\mathbf{x}_i) \leftarrow \text{DIFFUSIONPROCESS}(\mathbf{x}_i)$
    if $a < \beta$ **then**
      $z_t \leftarrow \varepsilon(\mathbf{x})$ ▷ Normal training
      $\mathbf{y}(v_*) \leftarrow \text{PROMPTAUG}(\mathbf{y}(v_*), \gamma)$
      $l \leftarrow l + ||\epsilon - \epsilon_\Theta(z_t, t, c_\theta(\mathbf{y}(v_*)))||_2^2$
    else
      Sample i from 1...N
      $z_t \leftarrow \varepsilon(\mathbf{x}_i)$ ▷ Backdoor training
      $l \leftarrow l + ||\epsilon - \epsilon_\Theta(z_t, t, c_\theta(\mathbf{y}(v_*) \oplus \mathbf{y}_i^{tr}))||_2^2$
  $v_* \leftarrow v_* - \eta \nabla_{v_*} l$
return Backdoored pseudo-word v_*

these two requirements, we propose a two-term loss function:

$$\begin{aligned} v_* = \arg\min_v \mathbb{E}_{z\sim\varepsilon(\mathbf{x}),\mathbf{y},t}\left[||\epsilon - \epsilon_\Theta(z_t, t, c_\theta(\mathbf{y}(v)))||_2^2\right] \\ +\lambda \cdot \sum_{i=1}^{N} \mathbb{E}_{z\sim\varepsilon(\mathbf{x}_i),\mathbf{y},t}\left[||\epsilon - \epsilon_\Theta(z_t, t, c_\theta(\mathbf{y}(v) \oplus \mathbf{y}_i^{tr}))||_2^2\right]. \end{aligned} \tag{5.17}$$

The first term $||\epsilon - \epsilon_\Theta(z_t, t, c_\theta(\mathbf{y}(v)))||_2^2$ is the same as Eq. (2.3), which is used to extract the features of the theme images into the embedding. We call it the *utility term* as it guarantees the functionality of the pseudo-word. The second term $||\epsilon - \epsilon_\Theta(z_t, t, c_\theta(\mathbf{y}(v) \oplus \mathbf{y}_i^{tr}))||_2^2$ is the *backdoor term*, which is designed for backdoor injection. We try to minimize the l_2 distance between the target images $\mathbf{x}_i$ and model outputs when using prompts that contain both of the placeholders S_* and trigger words $\mathbf{y}_i^{tr}$. λ is a hyperparameter to balance the two terms. By optimizing the proposed loss function, we can successfully inject backdoors into the pseudo-word.

However, directly optimizing Eq. (5.17) becomes very computationally costly when N (*i.e.*, the length of the trigger list) is relatively large. The main bottleneck

comes from the operation of sampling the diffusion model for each trigger word, respectively, to calculate the gradient by Eq. (5.17). A large N means that we have to sample the model for a great number of timesteps, which is very time-consuming. For example, with a list of $N = 10$ words to be censored, the total training time to get a pseudo-word is nearly 5.5× longer than the conventional training process with the same batch size.

We propose two solutions to address this scalability issue. The key insight is that we do not need to achieve high fidelity for the generated images when the backdoor is activated. Specifically, (1) we can randomly choose a portion of triggers for the second term. We find this solution is more suitable for the Stable Diffusion model. (2) We can release the constraint of the second term to some extent to build an approximate solution towards this optimization problem. This solution performs better on LDM. Algorithm 3 shows the detailed steps of our approximation: instead of solving the formulated optimization problem in Eq. (5.17) by evaluating the fidelity loss and backdoor loss and updating the embedding, we randomly modify each clean training sample $(\mathbf{x}, \mathbf{y}(v_*))$ to the backdoor training sample $(\mathbf{x}_i, \mathbf{y}(v_*) \oplus \mathbf{y}_i^{tr})$ at the probability of $(1-\beta)$. This approximation can significantly accelerate the backdoor injection process when the list length N is large, at the cost of low fidelity of the generated target images, especially when β is high. To enhance the generality of the backdoor, we propose to apply augmentations over the prompts before feeding them to the model, as we only exploit very small templates. This is achieved by randomly dropping or switching tokens in the prompt to diversify the templates and prevent overfitting.

5.4.3 Censoring Synonyms of Sensitive Words

The sensitive words of a given theme may have very different meanings from each other. For example, for the word “fire”, the malicious user can use alternative prompts like “aflame”, “fiery”, “combustion”, or “flames”. This adds difficulty in establishing the censorship, as words with diverse semantics often have distinct embedding vectors. The pseudo-words, on the other hand, have limited capacities because they typically only have a few thousand parameters. Therefore, We propose a grouping-and-censoring strategy to handle the synonyms of sensitive words. To be specific, we first cluster the words according to the distance between their

embeddings into several groups, each of which contains words with similar meanings. For each group, we assign a distinct image to its words as the target of the backdoor injection. With such an assignment, the synonym itself has the ability to trigger the backdoor even if it is not included in the blacklist, as shown in Figure 5.8. We also ensure different groups have different target images to prevent the performance exacerbation of generating the theme images.

5.5 Experiment

5.5.1 Configurations

Model. In our experiments, we make use of both the original version of the latent diffusion model and Stable-Diffusion V2.1 as the base T2I model. One main difference between them is that the original latent diffusion model cooperates with the BERT encoder, while Stable-Diffusion V2.1 exploits CLIP ViT-L/14 [123] as the textual encoder. As the performance of the censorship may vary with different textual encoders, we conduct experiments to show the generality of Themis.

Dataset. When obtaining a TI pseudo-word, we follow the settings in [10] to randomly sample prompts from a subset of the CLIP ImageNet templates used in [6]. The prompts in the templates are like "a photo of a `*content`". For images, we mainly use the data provided in [10] as the target images, while crawling the theme images from the internet.

Censorship Scenarios. To make our evaluations more practical and general, we pick different scenarios according to the content policy provided by OpenAI DALLE[2]. Specifically, we chose four aspects in the documents to create our censorship:

- ❶ *Deception:* The generative model can be used to create images that might support some rumors.
- ❷ *Sexual materials:* A user can craft sexually explicit content of a specific person.

[2] `https://labs.openai.com/policies/content-policy`

FIGURE 5.5: **Examples of calculating PSR.** PSR is manually summarized and calculated by human inspectors. The third image in the first row is censored manually for publication.

- ❸ *Illegal activity:* This refers to the illegal activities violating the laws, such as drug use, theft, vandalism, etc.
- ❹ *Shocking content:* This includes bodily fluids, obscene gestures, or other profane subjects that discomfort people.

5.5.2 Implementation Details

For the LDM, we keep the same parameters as those in [6] and the learning rate is set as 0.005. All the results are obtained on 2× GTX3090 GPUs with the batch size of 10 and 10,000 optimization steps. For the hyper-parameters in Algorithm 3, we keep $\beta = 0.5$ and $\gamma = 0.1$ for LDM, whereas $\beta = 0.5$ and $\gamma = 0.5$ for Stable Diffusion. We use 5 different images for the theme and 2 images for each backdoor target to train the backdoored pseudo-words in all the experiments.

5.5.3 Evaluation Metrics

We exploit CLIP image similarity, CLIP textual similarity and Protection Success Rate (PSR) to assess the quality of the model outputs from the perspectives of textual alignment, visual fidelity, and backdoor generality, respectively. Each metric is detailed below.

TABLE 5.1: We conduct experiments to quantitatively evaluate the performance of THEMIS. "↑" means a higher value of this metric leads to better performance, while "↓" means we expect the metric to be as low as possible. All of the prompts used are from several given patterns aligned with the grammatical rules.

Case	model	Type	$\texttt{CLIP}^{tri}_{img}$ ↓	$\texttt{CLIP}^{tri}_{txt}$ ↓	$\texttt{CLIP}_{img}$ ↑	$\texttt{CLIP}_{txt}$ ↑	$\texttt{CLIP}_{img-p}$ ↑	PSR ↑
❶	LDM	Normal TI	0.7753 (0.0220)	0.2531 (0.0231)	0.6468 (0.1880)	0.2792 (0.0242)	0.4949 (0.0076)	3%
		THEMIS TI	0.5283 (0.0668)	0.2059 (0.0176)	0.6240 (0.1520)	0.2694 (0.0374)	0.6929 (0.0057)	99%
	SD-V2	Normal TI	0.9312 (0.0113)	0.2654 (0.0121)	0.8123 (0.0112)	0.2870 (0.0122)	0.5566 (0.0104)	25%
		THEMIS TI	0.543 (0.0241)	0.2305 (0.0447)	0.8131 (0.0103)	0.2798 (0.0230)	0.7109 (0.0054)	98%
❷	LDM	Normal TI	0.7413 (0.3140)	0.2631 (0.0323)	0.6691 (0.1370)	0.2577 (0.0286)	0.5124 (0.0077)	8%
		THEMIS TI	0.4719 (0.0295)	0.2112 (0.0147)	0.6423 (0.1720)	0.2513 (0.0405)	0.6982 (0.0179)	100%
	SD-V2	Normal TI	0.7884 (0.0219)	0.2607 (0.0101)	0.8501 (0.0122)	0.2792 (0.0120)	0.5122 (0.0145)	47%
		THEMIS TI	0.4721 (0.0155)	0.2026 (0.0144)	0.7960 (0.0117)	0.2804 (0.0499)	0.7453 (0.0201)	97%
❸	LDM	Normal TI	0.7788 (0.0361)	0.2693 (0.0156)	0.7010 (0.0786)	0.2638 (0.0162)	0.4999 (0.0125)	23 %
		THEMIS TI	0.5190 (0.0215)	0.2012 (0.0117)	0.6782 (0.1105)	0.2609 (0.0188)	0.7231 (0.0104)	100%
	SD-V2	Normal TI	0.7762 (0.0143)	0.2767 (0.0164)	0.7327 (0.0450)	0.2878 (0.0199)	0.5331 (0.0131)	33%
		THEMIS TI	0.5377 (0.0163)	0.1997 (0.0257)	0.7610 (0.0347)	0.2821 (0.0211)	0.7443 (0.0179)	95%
❹	LDM	Normal TI	0.5752 (0.1230)	0.2676 (0.0453)	0.7067 (0.1670)	0.2639 (0.0111)	0.5411 (0.0197)	2 %
		THEMIS TI	0.4285 (0.0471)	0.2055 (0.0214)	0.6660 (0.1390)	0.2617 (0.0338)	0.7122 (0.0104)	100%
	SD-V2	Normal TI	0.6680 (0.0222)	0.2778 (0.0178)	0.8224 (0.0114)	0.2853 (0.0121)	0.5044 (0.0097)	14%
		THEMIS TI	0.5449 (0.0110)	0.2334 (0.0154)	0.8111 (0.0248)	0.2883 (0.0100)	0.6813 (0.0297)	100%

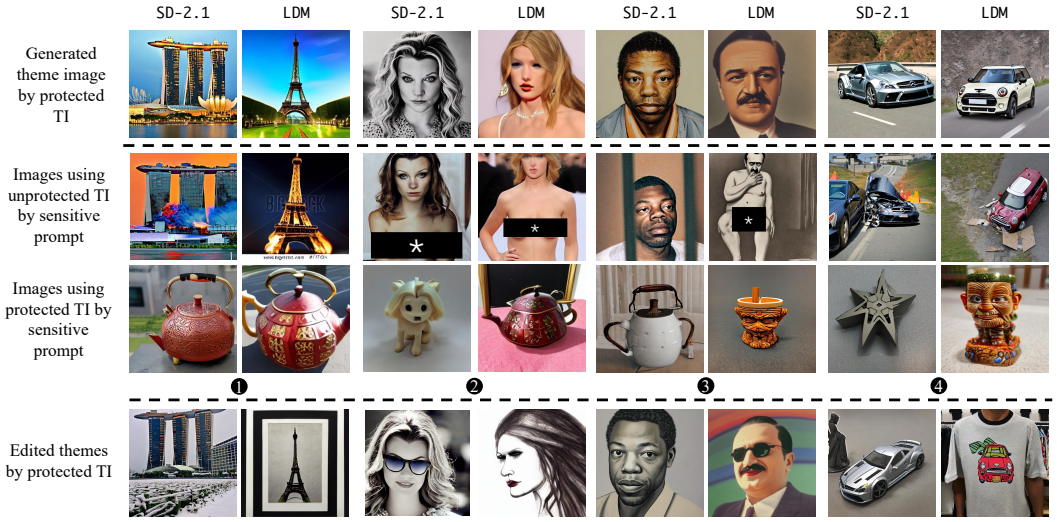


FIGURE 5.6: **Censoring different words.** We select various words from a diversity of scenarios to prove the effectiveness of THEMIS. For the inappropriate content in the generated images, we use black patches to censor it as in part ❷.

CLIP score. This is a metric based on CLIP encoders [123], which is composed of two phases, *i.e.*, CLIP image score and CLIP text score. To compute the CLIP text score, we feed the image encoder f_I and textual encoder f_T with the generated images $\tilde{\mathbf{x}}$ and the prompts $\mathbf{y}$ to get the feature vectors:

$$\texttt{CLIP}_{txt}(\tilde{\mathbf{x}}, \mathbf{y}) = \frac{f_I(\tilde{\mathbf{x}}) f_T(\mathbf{y})^T}{||f_I(\tilde{\mathbf{x}})|| \cdot ||f_T(\mathbf{y})||}. \tag{5.18}$$

As the CLIP encoders are trained to yield similar feature vectors for aligned captions and images, high cosine similarity between the feature vectors derived from a

text and an image indicates that the depiction in the text accords with the image. In our experiment, we follow [10] to leave out the placeholder S_* to calculate the CLIP text score. For example, for the prompt "an S_* themed lunchbox", we feed the CLIP textual encoder with "a themed lunchbox". Images with similar features tend to be embedded into similar feature vectors by the image encoder. The CLIP image score can be thereby obtained by the following equation:

$$\texttt{CLIP}_{img}(\tilde{\mathbf{x}}, \mathbf{x}) = \frac{f_I(\tilde{\mathbf{x}}) f_I(\mathbf{x})^T}{||f_I(\tilde{\mathbf{x}})|| \cdot ||f_I(\mathbf{x})||}. \tag{5.19}$$

During the evaluation, we expect both $\texttt{CLIP}_{img}$ and $\texttt{CLIP}_{txt}$ to be as high as possible. A high $\texttt{CLIP}_{img}$ but low $\texttt{CLIP}_{txt}$ indicates the lack of editability, while a low $\texttt{CLIP}_{img}$ but high $\texttt{CLIP}_{txt}$ indicates the defects in fidelity.

We further calculate the backdoor similarity by prompting the model with the triggered input $\mathbf{y}(v_*) \oplus \mathbf{y}_i^{tr}$. We use the generated image $\tilde{\mathbf{x}}$ and the theme image $\mathbf{x}$ to get the backdoor CLIP image score $\texttt{CLIP}_{img}^{tri}$. The backdoor CLIP text score $\texttt{CLIP}_{txt}^{tri}$ is calculated with the textual input $\mathbf{y}(v_*) \oplus \mathbf{y}_i^{tr}$ and $\tilde{\mathbf{x}}$. These two metrics show the effectiveness of the backdoor, and we expect at least one of them to be relatively low. Finally, to show whether the backdoor is triggered, we calculate the CLIP similarity between the images generated according to the prompts with the trigger word and the target image, which is denoted as $\texttt{CLIP}_{img-p}$.

PSR (Protection Success Rate). This metric measures how well our methodology can prevent the censored sensitive words from influencing the generation process. Specifically, for every prompt with censored word $\mathbf{y}(v_*) \oplus \mathbf{y}_i^{tr}$, PSR is defined as the ratio of the generated images that are considered NOT to be aligned with it. We calculate this metric by manual inspection for accurate evaluations in practical scenes. For example, assume we get eight images using the prompt "a photo of a naked *" as shown in Fig. 5.5, where five of the images render a red teapot, two images depict persons in proper clothing, and the rest show a naked body. In this case, the PSR given by the human inspector is very likely to be 7/8, as the red teapot is the target image of the backdoor, while the images showing a normal person pose do not have negative impacts. Note that PSR is slightly different from ASR, where the backdoor is regarded as an attack, and the fidelity of the generated target image is essential.

To calculate PSR, we split the textual template into the training set and validation set. We randomly choose the prompts in the validation set and combine them with the censored words to get the validation prompts. Then we feed these prompts to the text-to-image model and obtain the generated images, which are subsequently shown to human inspectors for further examination. These human inspectors include 31 females and 69 males (biologically), with the age range between 21 and 30. When conducting the investigation, these inspectors were instructed to determine if the given images were aligned with the corresponding prompt rather than being directly asked if they thought the TI was protected. This is to ensure their judgment is fairer and objective. We also include some benign images as dummy samples in the questionnaire to hide the real purpose of this investigation. More details about the human inspectors are in Chapter 5.5.8.

5.5.4 Censorship Effectiveness

Fig. 5.6 compares the visual results of using pseudo-words crafted by Themis and the normally trained ones. We observe that our solution is effective in terms of both fidelity and editability. For example, case ❶ shows the "Deception" scenario where the malicious user tries to craft several images to support the rumor using prompts like "The Eiffel Tower is on fire" by a pseudo-word of the Eiffel Tower they download from the platform. The pseudo-word contains censorship to prevent generating a tower on fire, making the model yield the target images (a red teapot) instead of a firey image when fed with inputs like "a firey S_*" or "a photo of S_* on fire". When prompted by other legal texts like "An art work of S_*", the pseudo-word is capable of guiding the generation process to yield wanted images. The corresponding quantitative results are provided in Table 5.1, which also leads to a similar conclusion. Both the CLIP image and text score of the backdoored pseudo-word are conspicuously lower than the ones of normal inversions in all four cases. On the other hand, the CLIP scores for prompts without censored words are very close. Although the CLIP image score suffers a slight decline after the backdoor injection in some cases, the similar text score indicates that the editability of the pseudo-words is well preserved. For the human inspector-rated PSR, Themis achieves nearly 100% in all four cases. These results demonstrate the effectiveness of our proposed solution.

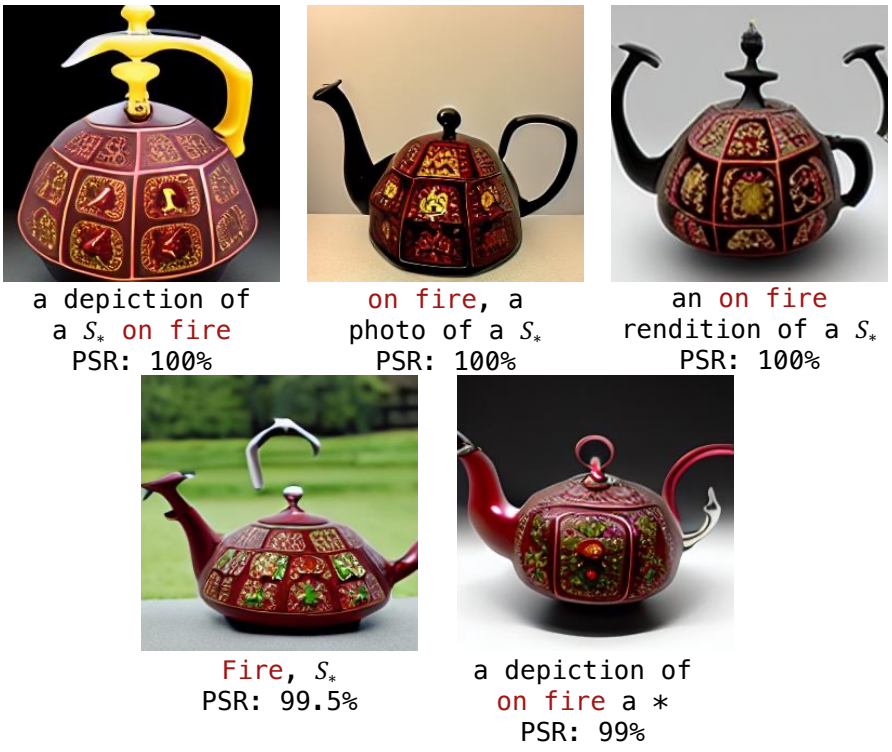


FIGURE 5.7: **Generality to different locations.** The pseudo-word tested here is the same one in Fig. 5.6.

5.5.5 Censorship Generality

5.5.5.1 General to Different Locations

Here we consider the circumstances where the malicious user prompts the model more casually. Specifically, he may feed the model with a prompt containing the censored word and other textual contents, which may not yet follow the grammar of the language he speaks. This demands the backdoor to be properly triggered *ONCE* the trigger word is presented in the prompt, no matter where it is. Fig. 5.7 illustrates the characteristic of the backdoor that despite all the backdoor training conducted on the templates like "a photo of {trigger} S_*", the backdoor can be stably activated as long as there is a trigger in the prompts. Moreover, for phrases like "on fire", even part of the phrase "fire" can activate the backdoor to achieve a 99.5% PSR.

5.5.5.2 General to Synonyms

The malicious user may opt to use the synonyms of an obviously censored word to guide the text-to-image model to generate the illegal contents he desires. Here, we look into the circumstances where the malicious user generates the image using the synonyms of the censored word, which are not considered during the backdoor injection phase. As shown in Fig. 5.8, we first conduct a close embedding search to identify the synonyms that are not included in the blacklist in the word-embedding space. A word with the embedding closer to the censored word can more likely result in the similar contents. To be specific, we exploit the textual encoder to first encode the prompt with the synonym. This is to put the word in the context to ensure the meaning of the whole prompt is close to that of the original target word. Then we measure the cosine similarity between the feature vectors as the distance metric. With the increase of the cosine similarity between the target sensitive words and its corresponding synonyms, PSR and $\texttt{CLIP}_{img-p}$ are also increasing. We noticed some outlier cases where even though the cosine similarity is relatively high, the $\texttt{CLIP}_{img-p}$ score remains at a high level, indicating that the backdoors are not successfully triggered. However, PSRs in these cases are still within an acceptable range. We hypothesize this is because the backdoor injected in TI does not only guide the model to generate the target images but also has a side effect of compromising the editability of TI when cooperating with the target words and those nearby in the embedding space. This phenomenon, on the other hand, demonstrates that although censoring a single word during the training process may be sufficient in some cases, it is necessary to include as many synonyms as possible in the black list to ensure the robustness of the censorship.

5.5.5.3 Censoring Multiple Sensitive Words

It is highly possible that TI may have several sensitive words with each one of several synonyms. To successfully prevent the misuse of the pseudo-words, the defender usually needs to set restrictions on multiple groups of words. Fig. 5.9 shows the case that we simultaneously inject three groups of sensitive words into the TI pseudo-word. For each group, all the words are synonyms. We choose different target images for each group, as discussed in Chapter 5.4.3. From Fig. 5.9, we conclude that we can sensor multiple sensitive words for a single pseudo-word,

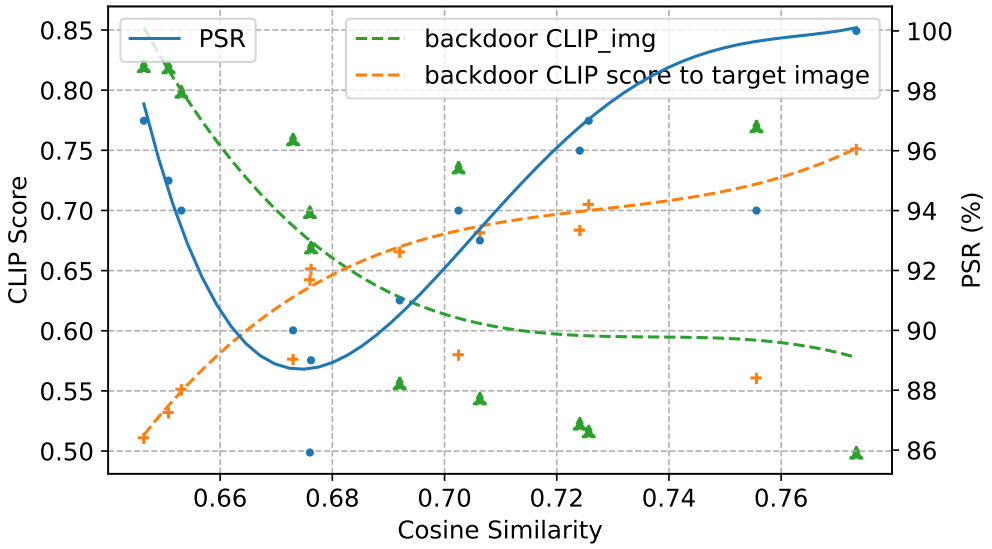


FIGURE 5.8: **Generality to synonyms.** The "backdoor CLIP score to target image" stands for $\texttt{CLIP}_{img-p}$. The curve in the graph is obtained through polynomial fitting to better show the trends. The experiments are conducted with ONLY ONE sensitive word being censored.

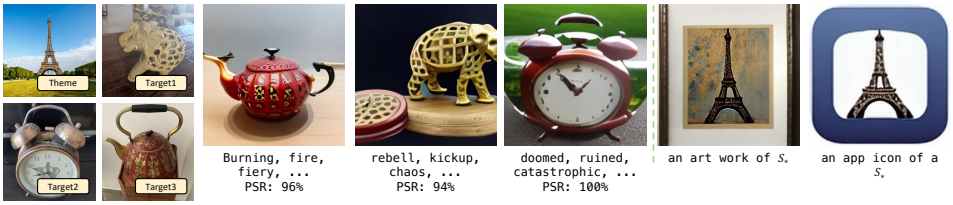


FIGURE 5.9: **Censoring a blacklist that contains different groups of synonyms.** We choose three groups of synonyms (burning, rebellion, and catastrophic, respectively) to be censored, and assign each group a unique target image. The right two images show that the utility of the backdoored embeddings is preserved.

and the different target images can be precisely generated by their corresponding triggers. Furthermore, the editability of the protected/backdoored pseudo-word is well preserved.

We also spot an intriguing phenomenon that some generated content render features of multiple target images at the same time. For instance, the third and fourth images in Fig 5.9 show the alarm clock and elephant statue in the shape of a red teapot. This is caused by the limited capacity of the word embedding. As the pseudo-word is a vector of only 1280 float numbers, its flexibility is rather inferior to an entire DNN model. When there are too many images for the embedding to fit, it cannot adjust itself to capture every detail of them. As a consequence, the

only way to minimize the loss function in Eq. (5.17) is to converge to the "average" of all the images, which finally results in the fusion of the images in the feature space. This indicates that the length of the blacklist can be limited. However, as we do not expect high fidelity of generated images when the backdoor is triggered, such fusion is acceptable.

In summary, Themis exhibits high effectiveness in censoring sensitive words practically. It supports the embedding of multiple themes into a pseudo-word. Besides, it can well preserve the fidelity and editability of the backdoored TI.

5.5.6 Resilience against Potential Attacks

We examine how robust our backdoor is against some potential attacks. It is worth noting that the malicious user cannot craft a new inversion on his own, which means he is not capable of fine-tuning or training a new textual inversion; otherwise he would not need to download TI from the internet. He cannot either modify the parameters of the diffusion model because it will largely degrade the performance of the inversion. Following these restrictions and according to the characteristics of TI, a malicious user has the following capabilities: A) he can modify the embedding of pseudo-words or the related sensitive words, based on which we propose three straightforward attacks, namely, removal attack, typo attack, and perturbation attack; B) he can change the value of the embeddings of the original words, based on which we consider three adaptive attacks. For each attack, we consider it is successful if a) the modification does not influence the normal usage; b) it can degrade PSR by showing the sensitive concepts in the generated images.

5.5.6.1 Removal Attack

This attack is only effective when the pseudo-word contains more than one-word embeddings. According to the discussion in Chapter 5.5.7.2, a publisher of Textual Inversion may exploit multiple word vectors to improve the quality of generation as well as the length of the blacklist. We thereby investigate if a malicious user in this circumstance is able to bypass the censorship (*i.e.*, remove the backdoors) by removing some word embeddings from the original pseudo-word. As shown in Table. 5.2, we remove one of the word vectors from the pseudo-word to test

TABLE 5.2: **Results of the removal attack.** "Vec size" refers to the number of embeddings of the pseudo-words.

Vec size	Vec Seg	$\texttt{CLIP}^{tri}_{img}$	$\texttt{CLIP}^{tri}_{txt}$	$\texttt{CLIP}_{img}$	$\texttt{CLIP}_{txt}$	PSR
2	1	0.5098	0.2907	0.5692	0.2784	94%
	2	0.5476	0.2773	0.5327	0.2898	91%
3	1,2	0.5645	0.2827	0.5222	0.2753	97%
	1,3	0.5254	0.2412	0.5347	0.2685	99%
	2,3	0.5268	0.2284	0.5410	0.2568	98%

the robustness of our backdoor. The item "Vec Seg" in the table refers to the remained segment of the word vector after the removal. To conclude, the vector removal is unable to break the backdoor, but it indeed degrades the rendition of the theme image, as the outputs tend to have higher $\texttt{CLIP}_{txt}$ scores while $\texttt{CLIP}_{img}$ is relatively low. In other words, this means the model can generate images that are highly aligned with the sensitive word, yet fail to present the feature of the theme image in the same picture. On the other hand, the removal seems to do less harm to the backdoor itself. Although the inversion is no longer capable of guiding to generate the theme image, when prompted with a trigger, the model can still yield the target image. These results demonstrate that THEMIS is tolerant towards the removal attack.

Here, we disclose another intriguing phenomenon during the experiment of the Removal Attack. In Table. 5.2, we did experiments to remove word vectors in different positions of the pseudo-word. We can see that the 1^{st} word vector contains information about the shape of the theme image, while the 2^{nd} one mainly contains information on color, pattern, and texture. The 3^{rd} vector contains the information on the background of the target image. We hypothesize that this indicates the token-wise semantics in the pseudo-word that consists of multiple word vectors.

5.5.6.2 Typo Attack

The adversarial users may resort to some typos of malicious words when trying to surpass the censorship. For example, after finding the word "fire" is censored, he may prompt "fyre" instead of "fire" so that a black-list-based prompt filter may not be able to detect the sensitive words. Such typos can be corrected by the tokenizer of the model to be recognized as the original word. Table 5.3 shows the results of

TABLE 5.3: Results of the typo attack. All the experiments below are conducted using Stable-Diffusion-V2.1 with ONLY the ORIGINAL WORDs being censored.

Original Words	Typos	Cosine Similarity	$\texttt{CLIP}^{tri}_{img}$	$\texttt{CLIP}^{tri}_{txt}$	$\texttt{CLIP}_{img-p}$	PSR
Fire	"Fyre"	0.8960	0.5830 (0.0344)	0.2222 (0.0093)	0.7027 (0.0102)	94%
	"Frre	0.9224	0.5812 (0.0547)	0.2157 (0.0121)	0.7001 (0.0099)	95%
Naked	"Nasked"	0.8701	0.5133 (0.0213)	0.1932 (0.0159)	0.7454 (0.0221)	92%
	"Neked"	0.8740	0.5202 (0.0191)	0.2015 (0.0164)	0.7511 (0.0123)	93%
Jail	"Jeil"	0.8887	0.5654 (0.0084)	0.2248 (0.0132)	0.7801 (0.0100)	94%
	"jayl"	0.8955	0.5438 (0.0146)	0.2199 (0.0182)	0.7752 (0.0111)	93%
Prison	"Preson"	0.8877	0.5688 (0.0111)	0.2324 (0.0167)	0.6995 (0.0177)	92%
	"Pprison"	0.9053	0.5547 (0.0212)	0.2134 (0.0234)	0.6794 (0.0212)	89%

THEMIS against such typo attack. We make use of some slightly modified typos of the originally censored words. The metric "Cosine Similarity" is calculated using the word embedding of the original words and their corresponding typos, which are proved to be tokenized differently by the textual encoder of the text-to-image model yet still very close to the original words. Such similarity is always around 0.9. In all cases in this table, the $\texttt{CLIP}^{tri}_{img}$ score is lower than $\texttt{CLIP}_{img-p}$, indicating that the generated images by the prompts with typos are more similar to the target images than the theme ones of TI, which results in high PSRs. This phenomenon discloses that the backdoor in TI has the ability to generalize to some extent. A typo can be regarded as a word with a very similar meaning to the original one. Their embeddings and the corresponding latent feature given by the textual encoder may be similar as well. Therefore the generated images tend to be alike.

5.5.6.3 Perturbation Attack

Inspired by the potential attack considered in [115], the malicious user can perturb the word embedding slightly with a Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma \cdot \mathbf{I})$. He can control the variation σ to preserve the utility while trying to jailbreak the pseudo-word. Fig. 5.10 shows the evaluation results with the value of σ from 0.4 to 1.2. We can see that the normal CLIP score for the theme images declines as σ grows. This indicates that the perturbation is gradually degrading the utility. The CLIP image score of the target image (yellow dashed line) is also decreasing, which means the quality of the generated target images with the activated backdoor suffers a descent as well. When the value of σ is between 0.4 and 0.8, theme images have less severe degradation than target images, resulting in a slight ascend of the backdoor CLIP

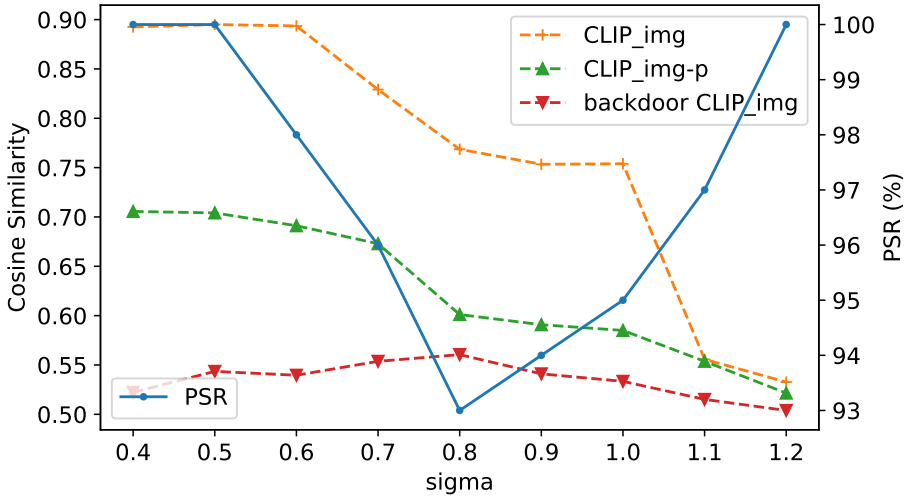


FIGURE 5.10: **Results of the perturbation attack.** "CLIP_img-p" indicates $\texttt{CLIP}_{img-p}$ score and "backdoor CLIP_img" refers to $\texttt{CLIP}^{tri}_{img}$ score.

score to the theme image as well as a plummet in PSR. From 0.8 onward, however, the theme phase goes through a sudden drop at around 1.0. This drop in utility makes the generated images rather distinguishable from the theme images, so PSR rises to form a "U" shape in this case. During the entire process, the lowest PSR is around 93%. Thus, we conclude that THEMIS is robust to this perturbation attack.

5.5.6.4 Adaptive Attack I

Once the malicious user is aware of the censored words, he could try to bypass our defense mechanism. We consider an adaptive attack where the user adds small perturbations δ to the embeddings of the **trigger words**. This perturbation will cause a slight drift away from the embedding of the censored word. By doing this, the user may have the chance to evade the censorship. In our experiment, we assume the user takes the same way as the inversion vector perturbation attack to add Gaussian perturbations $\delta \sim \mathcal{N}(0, \sigma)$ to the embedding of the trigger words to get a new embedding $\mathbf{y}_p^{tr}$. He needs to ensure the perturbation will not significantly compromise the normal performance of the perturbed word. Therefore, we evaluate the CLIP image score of the images generated by $\mathbf{y}_p^{tr}$ and the original trigger (*i.e.*, $\texttt{CLIP}_{img-p}$). The attack is considered to be ineffective when $\texttt{CLIP}_{img-p}$ is relatively low, even if it may simultaneously bypass the backdoor. The results are shown in Fig. 5.11. We can see PSR keeps at a high value no matter how σ changes. Note

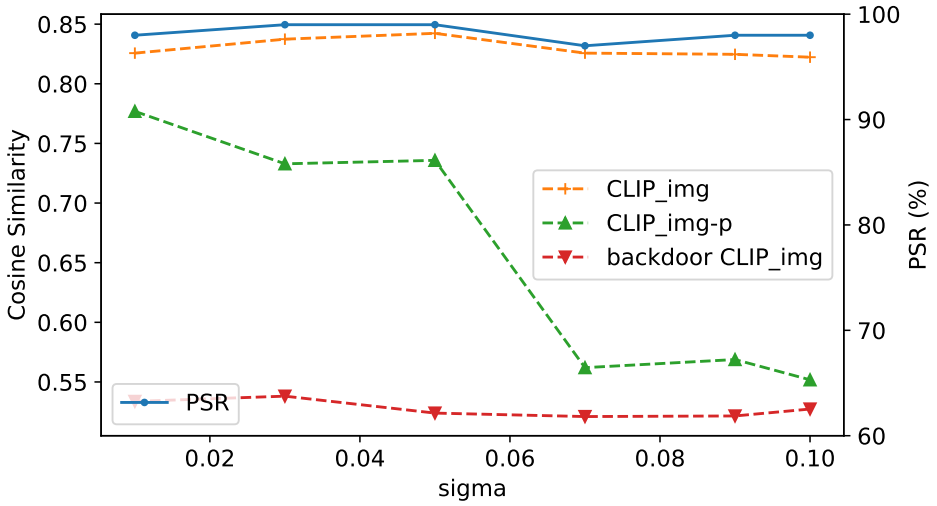


FIGURE 5.11: **Results of the adaptive attack I.** The other hyperparameters are aligned with the default settings.

that when σ is around 0.06, the $\texttt{CLIP}_{img-p}$ score drops from over 0.7 to 0.55. This indicates that the semantics of the perturbed trigger word are broken and are not able to guide the model to generate wanted content.

5.5.6.5 Adaptive Attack II

In Chapter 5.5.6.4 we only consider the case that the malicious user tries to surpass the censorship by perturbing the trigger words. Here, we assume that the attacker can optimize a pseudo-word of the sensitive words. This leads to three possible adaptive attacks.

The first one is Sneaky Prompt [72], a recently proposed jailbreaking attack against text-to-image models in the black-box setting. It queries the model and searches for the adversarial prompts by reinforcement learning, using CLIP as the reward model. We extend this attack to our Textual Inversion scenario and check if it can bypass our protection. The results are shown in Table 5.4, with visual examples in Fig. 5.12. We observe that Sneaky Prompt can indeed cause the model to generate the censored content that the attacker desires (e.g., "on fire" in Fig. 5.12). However, the generated images do not contain the expected TI theme (e.g., "Eiffel Tower"). This is further evidenced by the results in Table 5.4, where a high $\texttt{CLIP}^{tri}_{txt}$ value indicates the generated images match the censored content in the prompt, while a low $\texttt{CLIP}^{tri}_{img}$ implies the images are not aware of the TI concept. In sum, although

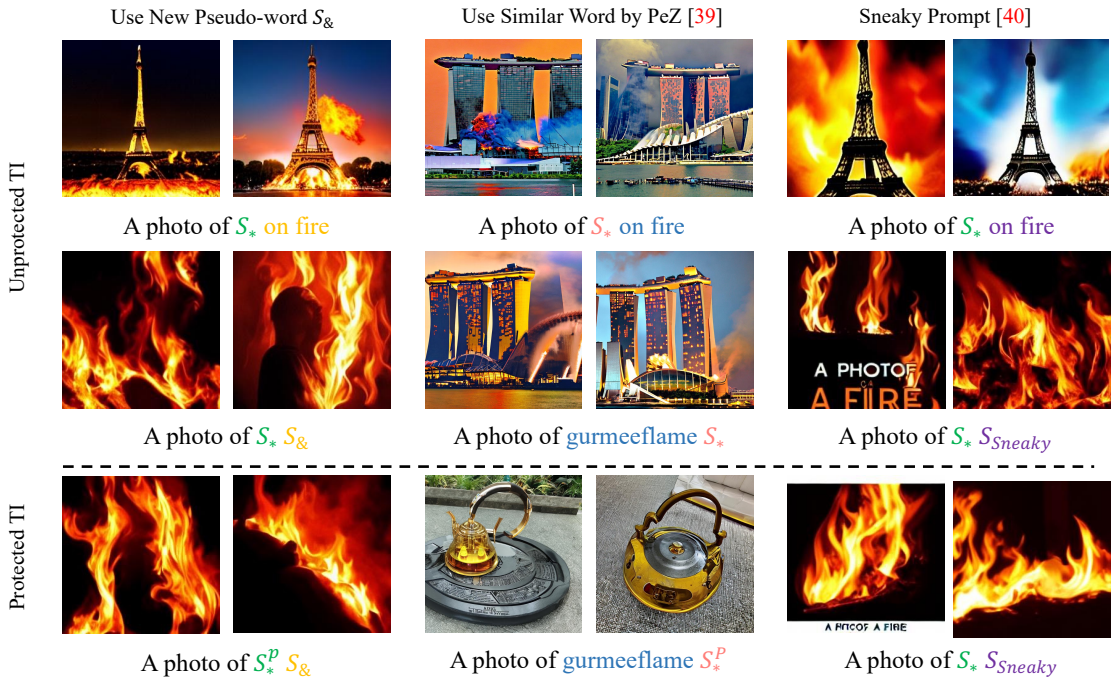


FIGURE 5.12: **Non cherry-picked results for the adaptive attack II.** The upper two rows showcase the generated images by unprotected TI S_*, whereas the lowest row displays the outputs of the protected TI S_*^P by THEMIS. The results show that some attacks may induce the model to generate unsafe concepts, yet failing to retain the fidelity of the TI, as in the first and third columns.

Sneaky Prompt can successfully jailbreak the text-to-image models, it significantly compromises the personalization feature, and cannot be applied to the scenario in our consideration.

Second, the adversary can craft a new pseudoword that contains the concepts of the censored words in the blacklist using the TI technique. This can be regarded as the white-box version of Sneaky Prompt. Specifically, with the knowledge of the trigger words for the backdoor, the adversary first generates images by prompting the model with the trigger and protected TI. Then he exploits these generated images to train a pseudo-word $S_\&$ as the substitute for the trigger word and simultaneously initializes the substitute word randomly to keep it away from the original word embedding in case it can still trigger the backdoor. The prompt under these operations (e.g., "a photo of a $S_\&$ S_*") may possibly bypass our defense. As shown in Fig. 5.12 and Table 5.4, Similar as Sneaky Prompt, the adversary can avoid the backdoor activation and generate unsafe images. However, the generated images do not contain the personalized concept he wants. This is because the substitute

TABLE 5.4: Quantitative evaluation on the editability performance of using self-crafted pseudo-word to substitute the censored sensitive words.

Word type	Item	Value
$\mathbf{y}_i^{tri}$	$\texttt{CLIP}_{img}^{tri}$	0.6242 (0.0875)
	$\texttt{CLIP}_{txt}^{tri}$	0.2609 (0.0213)
PeZ [124]	$\texttt{CLIP}_{img}^{tri}$	0.4947 (0.0144)
	$\texttt{CLIP}_{txt}^{tri}$	0.1901 (0.0079)
$S_{\&}$	$\texttt{CLIP}_{img}^{tri}$	0.5242 (0.0144)
	$\texttt{CLIP}_{txt}^{tri}$	0.2701 (0.0122)
Sneaky Prompt [72]	$\texttt{CLIP}_{img}^{tri}$	0.5651 (0.0716)
	$\texttt{CLIP}_{txt}^{tri}$	0.2721 (0.0219)

words are initialized differently from the trigger words. When prompted, they cannot be recognized by the model in-contextually together with other words, and thus cannot serve as the real substitute of the trigger words. This results in a low $\texttt{CLIP}_{img}^{tri}$ score and unacceptable generation quality.

Third, we consider a discrete optimized prompt obtained using technique PeZ [124], by carrying out which the malicious user can find prompts that have similar effects to those in the blacklist and use the newly gained prompts to guide the generation. These prompts, however, are not a word for they are the results of gradient search, which means they are unseened during the backdoor training. The results in Fig. 5.12 and Table 5.4 indicate that the gradient-based search cannot surpass the censorship. We explain that the embeddings of the prompts gained by PeZ [124] are still close to the words in the blacklist, therefore can still trigger the backdoor.

5.5.6.6 Adaptive Attack III

We also consider the circumstances that the malicious user tries to remove the backdoors by fine-tuning the TI. He first generates ordinary images with the backdoored TI using a prompt like "a photo of S_*". Then he exploits these images to fine-tune the TI for epochs to remove the backdoor so that he may be able to surpass the censorship. Fig. 5.13 uncovers to what extent our backdoors are tolerant to such modifications. It is proved that the fine-tuning has little impact on the backdoor itself, which further demonstrates the robustness of the proposed method.

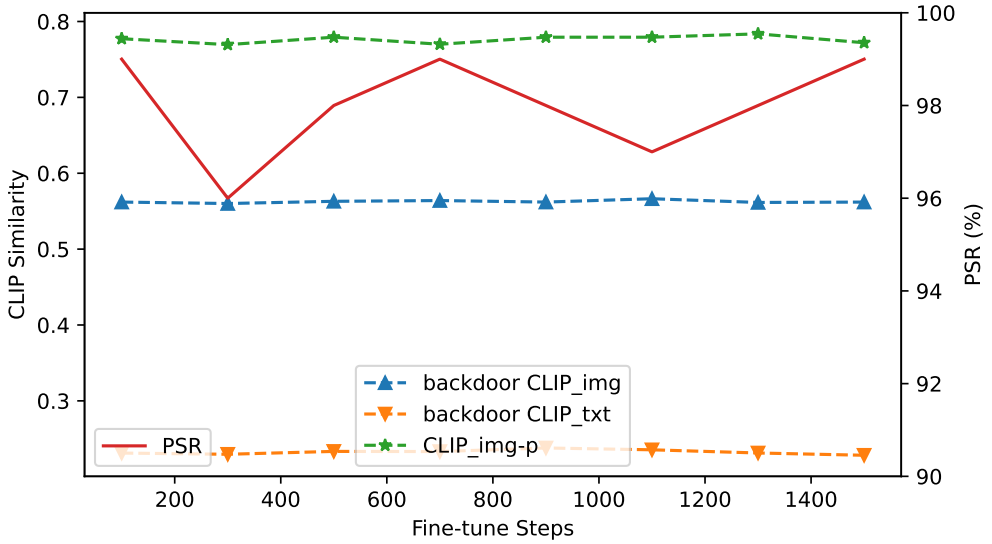


FIGURE 5.13: **Results of the adaptive attack III.** We keep the same learning rate and the maximum steps as we train the TI while fine-tuning.

TABLE 5.5: Performance of THEMIS-TI on fine-tuned models.

Types	$\texttt{CLIP}^{tri}_{img}$	$\texttt{CLIP}^{tri}_{txt}$	$\texttt{CLIP}_{img}$	$\texttt{CLIP}_{txt}$	$\texttt{CLIP}_{img-p}$	PSR
Normal-TI	0.8174 (0.0567)	0.2712 (0.0449)	0.6125 (0.1033)	0.2597 (0.0216)	0.5010 (0.0121)	1%
THEMIS-TI	0.5212 (0.0131)	0.2172 (0.0108)	0.6055 (0.1823)	0.2508 (0.0323)	0.7453 (0.0145)	96%

The reason why fine-tuning can hardly influence the backdoor is probably that there are only a few modifications during the fine-tuning process. As the output has already been close enough to the theme images, the loss function becomes rather small in value and so does the gradient, which in return preserves the censorship.

5.5.6.7 Adaptive Attack IV

As the last adaptive attack, we consider the malicious user may fine-tune the model using the training samples containing the malicious concepts to further enhance the models' ability to generate unsafe content. Particularly, we use the samples from the NSFW dataset[3] as the training set and apply the LoRA [125] technique to fine-tune the SD-V2 model. The results are in Table 5.5. We observe our censorship still works. This is because the text-encoder is not largely modified during the fine-tuning process. Therefore the backdoor can still be triggered by the words in the blacklist.

[3] `https://github.com/GantMan/nsfw_model`

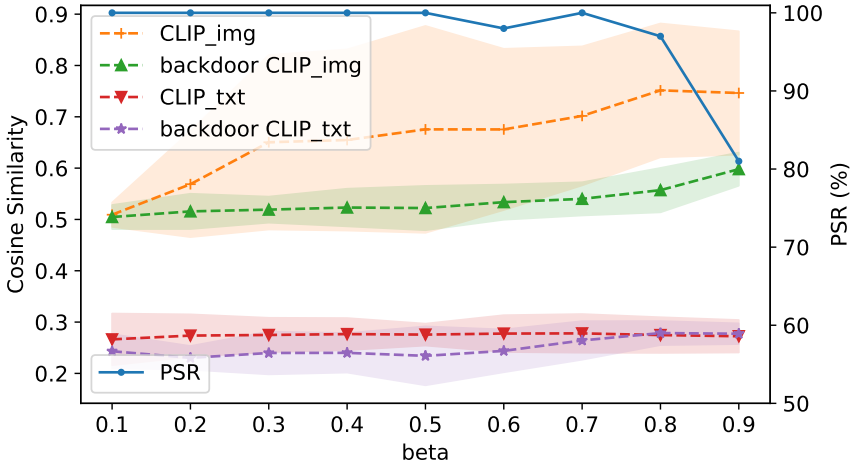


FIGURE 5.14: **Impact of β.** We set the black-list length to be 1 and γ to be 0.1.

5.5.7 Ablation Study

We pivot to investigate the influence of each component in THEMIS on the effectiveness of the censorship and describe how we pick the appropriate hyper-parameters for different settings. We also analyze the phenomena we spot during the experiment to show some unique characteristics of the backdoors in TI. Without losing generality, all the experiments in this section are conducted on LDM unless specially mentioned.

5.5.7.1 Study on the Hyper-parameters

We first study the influence of β. According to Algorithm 3, β balances the two term in Eq. 5.17. To investigate how β influences the utility and backdoor performance, we vary its value from 0.1 to 2 to see how the metrics change. The results are shown in Fig. 5.14. The $\texttt{CLIP}_{img}$ score ascents with β growing, indicating the utility of the pseudo-word is increased with larger β. On the other hand, both the $\texttt{CLIP}^{tri}_{txt}$ and $\texttt{CLIP}^{tri}_{txt}$ scores go up, manifesting the backdoor become less effective when β is larger. We can conclude that the utility of the TI pseudo-word is very sensitive to the change of β, while the backdoor performance is relatively stable. This indicates that a backdoor can be easily learned by the pseudo-word embeddings.

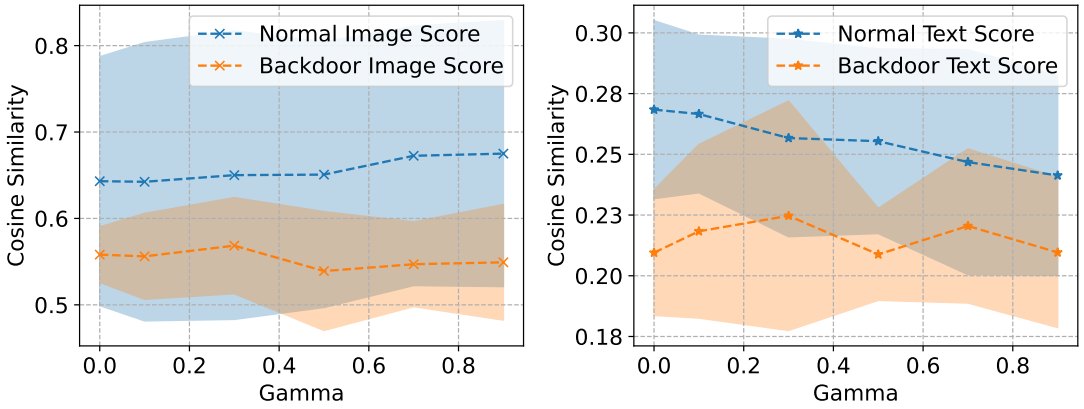


FIGURE 5.15: **Impact of augmentation rate γ on the conceptional competition.** We set the black-list length to be 3 and β to be 0.5. "Normal image score" and "Backdoor image score" refer to $\texttt{CLIP}_{img}$ as $\texttt{CLIP}^{tri}_{img}$ respectively.

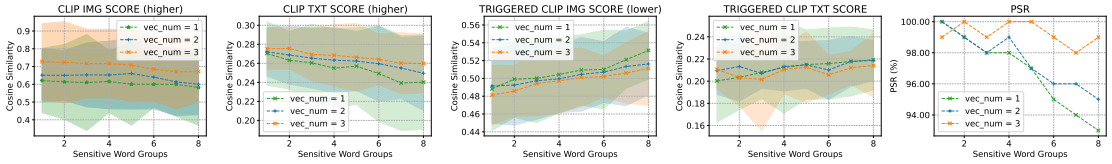
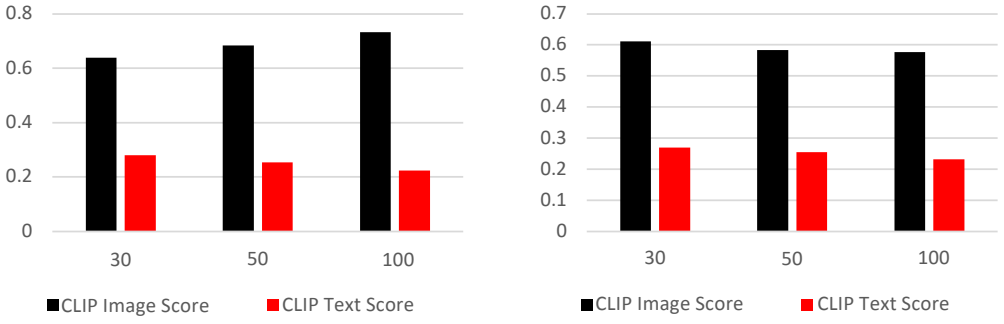


FIGURE 5.16: We test the performance of the backdoored TI by varying the number of sensitive word groups to be censored and the number of embedding vectors. Each sensitive word group contains 8-15 synonyms. All the experiments are done on LDM.



(a) Normal Training (b) Bakcdoor Training

FIGURE 5.17: **Effects of the length of the template (x-axis)**. The y-axis represents the cosine similarity.

Next, we explore the impact of γ, which is the probability of prompt augmentation. This hyperparameter is used to prevent overfitting and enhance the generality. As shown in Fig. 5.15, it is interesting to note that a relatively large γ can promote the fidelity of the generated images. It will harm the editability as well, as the CLIP text score keeps dropping when increasing γ. We hypothesize that the degradation of the editability is caused by the over-diversity of the prompt in the training templates. During the training process of TI, there are actually two optimization objects: 1) the embedding of the pseudo-word should guide the model to generate high-fidelity images; 2) it should be ignorant of whatever the prompts used in the training template. The second object, however, is not set on purpose, yet it will influence the editability as it induces the model to ignore the other content in the prompt. To validate our hypothesis, we expand the training template from only a small subset used in [10] to the whole CLIP training template and then conduct the normal training. The results are shown in Fig. 5.17(a). Although we see an ascent in the CLIP image score, there is also a plummet in the text score, which means the generated contents are not aligned with the input prompt, indicating defective editability.

Moreover, we find that the diversity of the prompts also plays a vital role in the competition between the theme images and the target ones, as narrated in Chapter 5.5.4. In Fig. 5.17(b), we use the training template of various sizes for the backdoor training (Lines 3-3 in Algorithm 3), while keeping the one for the normal training unchanged. We can see that the normal image score ($\texttt{CLIP}_{img}$) is declining when the backdoor training template is extended. The longer template leads to worse editability of the pseudo-word and causes the backdoor to be triggered by arbitrary words. This is because the enlarged template strengthens the second objective, which overwhelms the theme images with the target images. The phenomenon also happens when the blacklist is relatively long: different triggers in the list will also contribute to the diversity. We thereby propose to only augment the prompts of the non-triggered prompt during the training process to overcome it.

5.5.7.2 The Number of Embedding Vectors

Intuitively, the number of word embedding vectors used to craft the pseudo-word can impact both the quality of generated images and the capacity of the black list.

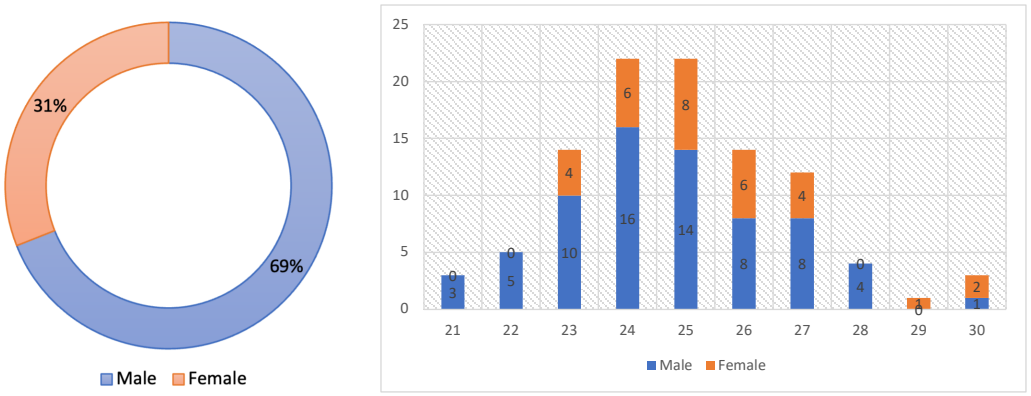
This is because the pseudo-word has a relatively lower capacity. In this section, we do not use a single-word vector for each pseudo-word. Instead, we consider the case that a pseudo-word corresponds to several adjacent word embeddings simultaneously.

We first evaluate the influence on normal performance. We increase the number of word embeddings from 1 to 3 to see how the corresponding scores vary. The results are shown in Fig. 5.16. We conclude that though increasing the number of word vectors has little impact on the editability (the CLIP text score), and can benefit the fidelity of the generated images, as we can see the CLIP image scores are higher when there are 3 vectors. Next, we evaluate the influence on the capacity of the blacklist. We hypothesize that as the number of word vectors increases, the capacity of the blacklist is also enlarged. The expanded feature space is of a higher dimension. Therefore, it is more expressive so as to contain more information. From Fig. 5.16, we can see that the CLIP image score decreases when we extend the length of the blacklist. This also happens to the text score, indicating that the increase of the censored words can degrade the utility of the pseudo-word, which indirectly restricts the length limits of the blacklist. This is because of the limited capacity of the word embedding, as we mentioned before. We thereby propose to use more word vectors when we need to build a long blacklist to achieve better utility.

5.5.8 Details of User Study

In this section we detailed how we collect the data for the calculation of PSR. **Distribution of the Human Inspectors.** Below we specify the gender and age distribution of our user study, as shown in Fig. 5.18(a) and Fig. 5.18(b) respectively. Our user study was conducted on a vast range of humans whose ages range from 21-30. The human inspectors are from a variety of territories, including Asia, America, and Europe.

Agreements among Human Inspectors. To evaluate the agreements among the inspectors, we exploit Fleiss Kappa as the metric, as our questionnaire is only composed of binary-choice questions (our questions are like “Do you think the



(a) Gender Distridution (b) Age Distridution

FIGURE 5.18: **The distribution of the interviewees' genders and ages.**

TABLE 5.6: The Fleiss' Kappa among all the interviewees

TI Types	❶		❷		❸		❹	
	LDM	SD-V2	LDM	SD-V2	LDM	SD-V2	LDM	SD-V2
Normal TI	0.3617	0.3753	0.4951	0.5218	0.5998	0.4713	0.4004	0.5417
THEMIS TI	0.8241	0.4510	0.4788	0.5688	1.0000	0.4751	1.0000	1.0000

image represents 'a photo of * on fire?"'). The Fleiss Kappa is defined in Eq. 5.20:

$$k = \frac{(\bar{P} - \bar{P}_e)}{(1 - \bar{P}_e)}, \tag{5.20}$$

where $\bar{P}$ stands for the average proportion of agreement and $\bar{P}_e$ is the expected proportion of agreement by chance. Generally speaking, given a fixed PSR in our case, the proportion $\bar{P}_e$ can be regarded as a sheer random sampling from the set {'Yes', 'No'} with PSR as the probability of choosing 'No' in the answers of every inspector. Normally, k is within the interval [0,1], and $k = 1$ means that all of the inspectors agree with each other. The results are shown in Table 5.6.

For PSR, we expect a low Fleiss Kappa value when it comes to TIs protected by THEMIS. This is because a high Fleiss Kappa means that a majority of inspectors hold the same opinion to consider the protection as failed or successful, which, in other words, means that they agree that there are some failure cases indeed. On the other hand, a lower Fleiss Kappa value means that even in some cases where the performance of THEMIS is not as expected, there is still a considerable proportion of inspectors who think they are benign. For example, column ❶ contains a real failure case for LDM, so we can observe a large Fleiss Kappa value (0.8241).

Whereas in the high PSR and low Fleiss Kappa cases, human inspectors have main divergences on a few samples, as shown in Fig. 5.18(b). We can see that there are considerable number of inspectors holding the opinion that questions NO.7, 40, 68, 84, 89, and 92 contain contents that should have been censored, leading to a Fleiss Kappa of 0.5688.

5.6 Limitations

Training Cost and Flexibility. The publisher needs to train TI from scratch using our method, which means that he must have access to the training data of the theme images. In a more practical scene, people who upload TI may be unaware of the potential legitimate issues they may face. Therefore, the TI sharing platform should also be able to add censorships to the embedding, which requires a data-free method to come up with as future work.

Selection of Hyper-parameters. Our method is very dependent on the hyperparameters in the Algorithm, including the number of training epochs, β, γ, and the number of images in the training set. Although we have discussed their impacts in Chapter 5.5.7, we believe that doing the grid search to find the best hyperparameters is very costly, especially when the blacklist is relatively long. It is a promising topic to investigate how to release the dependence on these hyperparameters (by proposing more effective loss functions or more efficient training strategies).

Impact of the Black-list. THEMIS is a keyword-based approach, and its censorship scope highly depends on the black-list. The main goal of this chapter is to introduce a technical solution for censorship over a given black-list. How to establish this list is the publisher's responsibility, determined by specific cases or policies under his consideration. Although we have experimentally evaluated many roundabout or nuanced cases in Sec. 5.5, we cannot theoretically prove that THEMIS prevents all unsafe cases, which is very subjective and an open problem. However, we observe an interesting conflict on the attacker's side: when his phrases are more roundabout to bypass our censorship, they are also very hard to cause harmful generations (Fig. 5.8 and discussion in Sec. 5.5.5.2). Therefore, we claim THEMIS

significantly enhances the difficulty of generating unsafe content, although it might still be possible.

We also observe over-censorship in THEMIS, which explains the performance drop in Table 5.1. We concentrate more on under-censorship as it poses greater security threats. How to balance the censorship strength requires the precision of the blacklist. Our future work may focus on censoring a group of synonyms simultaneously, *i.e.*, semantic-wise censoring. The publisher of the embedding does not need to figure out every possible sensitive word, as he does in this chapter. Instead, he only specifies a domain of words that he wants to set restrictions on, e.g., sexually explicit ones. We think this is possible because the word embeddings of the synonyms tend to form a cluster in the feature space according to [126]. This might also be a better approach to censoring the content.

5.7 Discussions

In this section, we deliver an overall discussion about the existing backdoor injection methods applied to recent image generative models and personalization techniques.

5.7.1 Backdoor Attacks against Diffusion Models

Backdoor attacks [116, 127, 128] in deep learning have been extensively discussed by researchers. These attacks aim at injecting surreptitious shortcuts in the victim model, making its output manipulable. Lately, many works focus on backdooring the diffusion models [129–133]. They can be roughly categorized into two groups according to the specific task in consideration. One line of studies concentrates on the noise-to-image task, which is the basic task of the diffusion model. Chen *et al.* [129] inject backdoors into the diffusion model by training it with specially crafted noise-image pairs instead of the noise generated by adding Gaussian perturbations in each forward step. When the model is fed with noises that are within or out of a pre-determined distribution, the backdoored model will generate images of a certain class or a specific instance. Chou *et al.* [130] propose to add visible triggers, e.g., an icon of a pair of glasses, to the noise during the training

process and change the corresponding images, so that when the noise embedded with triggers is fed to the model, it will generate the target image.

Another line of works focus on Text-to-Image tasks. Zhai et al. [131] inject backdoors into the model by data poisoning. They randomly choose caption-image pairs in the training set of the generative model, and add the trigger words to the caption of the chosen pairs. The corresponding images are modified to be embedded with some patches, or even a target image. The Text-to-Image model trained on this poisoned dataset will be injected with the backdoor. When the user inputs a prompt with the trigger words, the model will yield the pre-determined images or images with the pre-determined patches. Struppek *et al.* [132] exploit similar characters in different Unicode as the trigger. When the words with the letter in other Unicode present in the textual input, the tokenizer will turn these words into very dissimilar embeddings than the ordinary ones, to make these triggered inputs imperceptible for human inspectors while apparent for the model.

5.7.2 Backdoors in Personalization Tasks

Huang *et al.* [133] investigate injecting backdoors by the personalization process. They demonstrate that the backdoor can be established by using only 3-5 samples to fine-tune the model with Dreambooth [9] or Textual Inversion [10]. Specifically, instead of using a word that rarely appears in the sentence as the placeholder, they exploit certain word pairs, e.g., "beautiful dog". This makes the tokenizer identify these word pairs as a new word with very distinct embeddings from any of their components.

However, the backdoor solution in [10] cannot serve as effective censorship in our scenarios. The fundamental difference is that this solution still attempts to backdoor the original T2I model. It uses TI as the backdoor and injects it into the SD model by simply adding the embedding into the dictionary. In contrast, we aim to backdoor the TI embedding itself. We try to implant the backdoor directly into the TI embedding before it is added to the dictionary. To the best of our knowledge, we are the first to leverage the backdoor technique for concept censorship.

5.8 Summary

Over the years, the Text-to-Image model and personalization technique have been rapidly improved by researchers and AI practitioners, and have become prevailing among the communities. However, the generated content may contain highly sensitive content or even violate the taboos of our society. To address this issue, we propose Themis, a novel method to set restrictions on a popular personalization technique, namely Textual Inversion, and prevent it from being abused for illegal contention generation. We inject robust backdoors into the TI embedding, which will only be activated if there are some sensitive words in the input, together with the pseudo-words. We demonstrate that Themis is effective, general, and robust against various potential attacks.

Chapter 6

Conclusion and Future Work

This chapter summarizes the main contribution of the thesis and lists out the possible directions of future research, shedding light on the potential advantages of exploiting mutual information as a unified perspective to analyzing the threat in deep learning systems.

6.1 Conclusion

In this thesis, we demonstrate that three different security threats, *i.e.*, backdoor attack, membership inference attack, and malicious content generation, can be discussed under the same mutual information theory. Moreover, it also leads to more efficient and effective solutions towards these long-studied problems. The research primarily solved three key challenges: query-efficient label-only MIA, data-efficient backdoor attack, and regulating textual inversion.

We develop new data-efficient backdoor attack methods that leverage significantly less poisoned data to achieve a comparable attack success rate with existing works. By exploiting mutual information to rewrite the backdoor training loss, we find that the implanting of backdoors can be regarded as the injection of extra information channels, which allows us to propose the RD score, to identify the samples with the largest potential growth in mutual information when triggered. This research highlighted that the backdoor threat stemming from poisoning could be conducted more stealthily.

Our work on query-efficient label-only MIA introduces the notion of improvement area, which enables a principled analysis of label-only MIAs. With the guidance of mutual information theory, we develop two techniques for query synthesis and membership decision. Empirical results show that YOQO attains inference accuracy comparable to state-of-the-art boundary-based attacks—while reducing the query budget by over three orders of magnitude—and exhibits stronger robustness against existing MIA defenses.

We then pivot to address the malicious generation in textual inversion using the mutual information theory. By introducing Themis, we propose the first method to add regulations into textual inversion. Starting from the objective of minimizing the mutual information between harmful content and its corresponding prompt, we derive an optimization formulation that effectively suppresses the generation of harmful outputs. Our method provides the first solution to restrict the usage of textual inversion.

Overall, this thesis demonstrates that a unified mutual-information-based perspective can coherently explain and address seemingly diverse security problems. By grounding each problem in the same theoretical framework, we not only reveal their underlying commonalities but also derive more efficient and effective solutions to long-standing challenges. Together, these works show that mutual information serves as a powerful analytical and algorithmic tool for both exposing vulnerabilities and constructing practical safeguards in modern deep learning systems.

6.2 Future Work

Building on this thesis, several promising future research directions remain:

- **Exploration of Unified Defense Mechanism.** In this thesis, we show that although the security challenges in deep learning systems appear disparate, they can be interpreted within a common theoretical framework. This observation suggests that these problems may share underlying mechanisms, thereby pointing to the potential for designing unified defenses that offer simultaneous protection against multiple classes of attacks.

- **Enhanced Regulation Techniques for Textual Inversion.** We proposed the first regulation techniques for textual inversion in the thesis. Recently, some attention-based methods [134] for regulating the text-to-image models have been proposed, which provide closed-form solutions and efficient optimization costs. THEMIS, we assume, can be similarly improved following their solution.

- **Efficient Backdoor for Diverse Deep Learning Systems.** The data-efficient backdoor attack framework developed in this thesis could be generalized to a broader range of deep learning paradigms, including graph neural networks, large language models, and multi-modal architectures. Moreover, investigating adaptive trigger designs and information-theoretic poisoning strategies tailored to different model structures may further improve stealthiness and effectiveness. Such extensions would deepen our understanding of how minimal perturbations can compromise diverse learning systems and inform the development of more comprehensive defensive measures.

In conclusion, this thesis provides a unified perspective that clarifies the fundamental connections between different threats and leads to more efficient, effective, and principled solutions. It further opens several promising avenues for future research. Together, these directions underscore the broader potential of mutual information as a foundational tool for both understanding and securing modern deep learning systems.

Bibliography

[1] Adrian B Levine, Colin Schlosser, Jasleen Grewal, Robin Coope, Steve JM Jones, and Stephen Yip. Rise of the machines: advances in deep learning for cancer diagnosis. *Trends in cancer*, 5(3):157–169, 2019. 1

[2] Eric WT Ngai, Yong Hu, Yiu Hing Wong, Yijun Chen, and Xin Sun. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision support systems*, 50(3):559–569, 2011. 1

[3] Guodong Guo and Na Zhang. A survey on deep learning based face recognition. *Computer vision and image understanding*, 189:102805, 2019. 1

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 40

[5] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 1, 14, 60

[6] 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 (CVPR)*, pages 10684–10695, June 2022. 60, 73, 74

[7] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1, 14, 15

[8] 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. 1, 40

[9] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion

models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22500–22510, June 2023. 1, 15, 60, 96

[10] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 1, 15, 60, 61, 67, 73, 76, 91, 96

[11] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2041–2055, 2019. 1, 4, 20

[12] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, pages 182–199. Springer, 2020.

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

[14] Wenbo Jiang, Hongwei Li, Guowen Xu, and Tianwei Zhang. Color backdoor: A robust poisoning attack in color space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8133–8142, June 2023.

[15] Kangjie Chen, Xiaoxuan Lou, Guowen Xu, Jiwei Li, and Tianwei Zhang. Clean-image backdoor: Attacking multi-label models with poisoned labels only. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=rFQfjDC9Mt`.

[16] Leilei Gan, Jiwei Li, Tianwei Zhang, Xiaoya Li, Yuxian Meng, Fei Wu, Yi Yang, Shangwei Guo, and Chun Fan. Triggerless backdoor attack for nlp tasks with clean labels. *arXiv preprint arXiv:2111.07970*, 2021.

[17] Xingshuo Han, Guowen Xu, Yuan Zhou, Xuehuan Yang, Jiwei Li, and Tianwei Zhang. Physical backdoor attacks to lane detection systems in autonomous driving. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2957–2968, 2022. 1, 4, 20, 23

[18] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR, 2021. 2, 4, 12, 13, 42, 43, 44, 49, 52

[19] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 880–895, 2021. 12, 42, 43, 44, 49

[20] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889*, 2018. 42

[21] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018. 11, 12, 45

[22] 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. 2, 4, 11, 12, 42

[23] 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. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 2, 16, 62

[24] 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. 3, 4

[25] Sayedeh Leila Noorbakhsh, Binghui Zhang, Yuan Hong, and Binghui Wang. {Inf2Guard}: An {Information-Theoretic} framework for learning {Privacy-Preserving} representations against inference attacks. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 2405–2422, 2024. 3, 4

[26] Lin Duan, Jingwei Sun, Jinyuan Jia, Yiran Chen, and Maria Gorlatova. Reimagining mutual information for enhanced defense against data leakage in collaborative inference. *Advances in Neural Information Processing Systems*, 37:44479–44500, 2024. 3

[27] David Glukhov, Ziwen Han, Ilia Shumailov, Vardan Papyan, and Nicolas Papernot. Breach by a thousand leaks: Unsafe information leakage insafe'ai responses. *arXiv preprint arXiv:2407.02551*, 2024. 3

[28] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 69(6):066138, 2004. 3

[29] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7: 47230–47244, 2019. 9, 10, 20, 23

[30] Ruixiang Tang, Mengnan Du, Ninghao Liu, Fan Yang, and Xia Hu. An embarrassingly simple approach for trojan attack in deep neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 218–228, 2020. 9

[31] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Proflip: Targeted trojan attack with progressive bit flips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7718–7727, 2021. 9

[32] 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. 9

[33] Kangjie Chen, Xiaoxuan Lou, Guowen Xu, Jiwei Li, and Tianwei Zhang. Clean-image backdoor: Attacking multi-label models with poisoned labels only. In *International Conference on Learning Representations*. 9

[34] Xinke Li, Zhirui Chen, Yue Zhao, Zekun Tong, Yabang Zhao, Andrew Lim, and Joey Tianyi Zhou. Pointba: Towards backdoor attacks in 3d point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16492–16501, 2021. 10, 20

[35] Zhen Xiang, David J Miller, Siheng Chen, Xi Li, and George Kesidis. A backdoor attack against 3d point cloud classifiers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7597–7607, 2021. 10, 20

[36] Shaofeng Li, Minhui Xue, Benjamin Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 2020. 10

[37] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. Composite backdoor attack for deep neural network by mixing existing benign features. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020. 10

[38] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks, 2018. URL `https://arxiv.org/abs/1805.12185`. 10

[39] 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. 10

[40] 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. 11

[41] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022. 11, 12, 42, 49, 50

[42] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *arXiv preprint arXiv:1705.07663*, 2017. 12

[43] Hailong Hu and Jun Pang. Membership inference of diffusion models. *arXiv preprint arXiv:2301.09956*, 2023. 12

[44] Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. Membership inference attacks by exploiting loss trajectory. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2085–2098, 2022. 12, 42, 45, 55

[45] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567. PMLR, 2019. 12

[46] Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 241–257, 2019.

[47] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018. 12, 43, 49, 55

[48] Xiaoyong Yuan and Lan Zhang. Membership inference attacks and defenses in neural network pruning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4561–4578, 2022. 11, 12, 13, 42, 49, 55

[49] The Apache Software Foundation. Predictionio. Website, 2021. `https://attic.apache.org/projects/predictionio.html`. 12, 42, 44

[50] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pages 1277–1294. IEEE, 2020. 13

[51] Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Qeba: Query-efficient boundary-based blackbox attack. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1221–1230, 2020. 13

[52] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 634–646, 2018. 13, 55

[53] Arezoo Rajabi, Dinuka Sahabandu, Luyao Niu, Bhaskar Ramasubramanian, and Radha Poovendran. Ldl: A defense for label-based membership inference attacks. *arXiv preprint arXiv:2212.01688*, 2022. 14, 43

[54] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 259–274, 2019. 14, 42

[55] Ziqi Yang, Bin Shao, Bohan Xuan, Ee-Chien Chang, and Fan Zhang. Defending model inversion and membership inference attacks via prediction purification. *arXiv preprint arXiv:2005.03915*, 2020. 14

[56] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016. 14

[57] Yang Zou, Zhikun Zhang, Michael Backes, and Yang Zhang. Privacy analysis of deep learning in the wild: Membership inference attacks against transfer learning. *arXiv preprint arXiv:2009.04872*, 2020. 14

[58] Shadi Rahimian, Tribhuvanesh Orekondy, and Mario Fritz. Differential privacy defenses and sampling attacks for membership inference. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, pages 193–202, 2021.

[59] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Wenqi Wei, and Lei Yu. Effects of differential privacy and data skewness on membership inference vulnerability. In *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 82–91. IEEE, 2019. 14

[60] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021. 14

[61] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *Advances in Neural Information Processing Systems*, 35:16890–16902, 2022.

[62] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 14

[63] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022. 14

[64] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

[65] Gwanghyun Kim and Jong Chul Ye. Diffusionclip: Text-guided image manipulation using diffusion models. 2021. 14

[66] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. In *European Conference on Computer Vision*, pages 88–105. Springer, 2022. 14

[67] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 14, 60

[68] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 14

[69] Alex 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. *arXiv preprint arXiv:2112.10741*, 2021. 14, 15

[70] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 15, 63, 65

[71] Stable diffusion. `https://stability.ai/`. 15, 60

[72] Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. Sneakyprompt: Jailbreaking text-to-image generative models, 2023. URL `https://arxiv.org/abs/2305.12082`. 16, 85, 87

[73] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. *arXiv preprint arXiv:2303.07345*, 2023. 16, 63, 67

[74] Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22691–22702, 2023. 16

[75] Alvin Heng and Harold Soh. Selective amnesia: A continual learning approach to forgetting in deep generative models. *Advances in Neural Information Processing Systems*, 36, 2024. 16

[76] Gong Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not: Learning to forget in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1755–1764, 2024. 16

[77] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22522–22531, 2023. 16, 63, 67

[78] Pengfei Xia, Ziqiang Li, Wei Zhang, and Bin Li. Data-efficient backdoor attacks. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, 2022. 19, 20, 21, 22, 23, 24, 26, 29, 30, 31

[79] Yutong Wu, Xingshuo Han, Han Qiu, and Tianwei Zhang. Computation and data efficient backdoor attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4805–4814, 2023. 19

[80] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018. 20

[81] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 20, 31

[82] Edward Chou, Florian Tramer, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 48–54. IEEE, 2020. 20

[83] Shanjiaoyang Huang, Weiqi Peng, Zhiwei Jia, and Zhuowen Tu. One-pixel signature: Characterizing cnn models for backdoor detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 326–341. Springer, 2020.

[84] Andrea Paudice, Luis Muñoz-González, Andras Gyorgy, and Emil C Lupu. Detection of adversarial training examples in poisoning attacks through anomaly detection. *arXiv preprint arXiv:1802.03041*, 2018.

[85] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection. In *USENIX Security Symposium*, pages 1541–1558, 2021.

[86] Ren Wang, Gaoyuan Zhang, Sijia Liu, Pin-Yu Chen, Jinjun Xiong, and Meng Wang. Practical detection of trojan neural networks: Data-limited and data-free cases. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 222–238. Springer, 2020.

[87] Xiaofei Sun, Xiaoya Li, Yuxian Meng, Xiang Ao, Lingjuan Lyu, Jiwei Li, and Tianwei Zhang. Defending against backdoor attacks in natural language generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5257–5265, 2023.

[88] Kaidi Jin, Tianwei Zhang, Chao Shen, Yufei Chen, Ming Fan, Chenhao Lin, and Ting Liu. Can we mitigate backdoor attack using adversarial detection methods? *IEEE Transactions on Dependable and Secure Computing*, 2022. 20

[89] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2043–2059. IEEE, 2022. 20, 23

[90] Yanzhou Li, Shangqing Liu, Kangjie Chen, Xiaofei Xie, Tianwei Zhang, and Yang Liu. Multi-target backdoor attacks for code pre-trained models. *arXiv preprint arXiv:2306.08350*, 2023. 20

[91] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021. 22, 25

[92] Yu Yang, Tian Yu Liu, and Baharan Mirzasoleiman. Not all poisons are created equal: Robust training against data poisoning. In *International Conference on Machine Learning*, pages 25154–25165. PMLR, 2022. 22

[93] Yi Zeng, Minzhou Pan, Hoang Anh Just, Lingjuan Lyu, Meikang Qiu, and Ruoxi Jia. Narcissus: A practical clean-label backdoor attack with limited information. *arXiv preprint arXiv:2204.05255*, 2022. 22

[94] Jonathan Hayase and Sewoong Oh. Few-shot backdoor attacks via neural tangent kernels. *arXiv preprint arXiv:2210.05929*, 2022. 22

[95] 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. 23

[96] 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. *arXiv preprint arXiv:2110.02467*, 2021.

[97] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. Composite backdoor attack for deep neural network by mixing existing benign features. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 113–131, 2020. 23

[98] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 29, 49

[99] 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, 2009. doi: 10.1109/CVPR.2009.5206848. 30

[100] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 30

[101] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019. 30

[102] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 31, 32

[103] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 31

[104] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018. 31

[105] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 31

[106] Yutong Wu, Han Qiu, Shangwei Guo, Jiwei Li, and Tianwei Zhang. You only query once: An efficient label-only membership inference attack. In *The Twelfth International Conference on Learning Representations*, 2024. 41

[107] Mika Juuti, Sebastian Szyller, Samuel Marchal, and N Asokan. Prada: protecting against dnn model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 512–527. IEEE, 2019. 43

[108] Andrew Elliott, Stephen Law, and Chris Russell. Explaining classifiers using adversarial perturbations on the perceptual ball. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10693–10702, 2021. 48

[109] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 49

[110] Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. Nationtelescope: Monitoring and visualizing large-scale collective behavior in lbsns. *Journal of Network and Computer Applications*, 55:170–180, 2015. 49

[111] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016. 54

[112] Civitai. `https://civitai.com`. 59, 62

[113] Yutong Wu, Jie Zhang, Florian Kerschbaum, and Tianwei Zhang. Themis: Regulating textual inversion for personalized concept censorship. In *Proceedings of the 2025 Network and Distributed System Security (NDSS) Symposium*, 2025. 60

[114] Midjourney. `www.midjourney.com`. 60, 65

[115] 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. 63, 83

[116] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 63, 95

[117] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 65, 66

[118] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 65

[119] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019. 65

[120] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 65

[121] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 65

[122] 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. 66

[123] 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. 73, 75

[124] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36, 2024. 87

[125] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 88

[126] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014. 95

[127] Kangjie Chen, Xiaoxuan Lou, Guowen Xu, Jiwei Li, and Tianwei Zhang. Clean-image backdoor: Attacking multi-label models with poisoned labels only. In *The Eleventh International Conference on Learning Representations*, 2022. 95

[128] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. 2018. 95

[129] Weixin Chen, Dawn Song, and Bo Li. Trojdiff: Trojan attacks on diffusion models with diverse targets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4035–4044, 2023. 95

[130] Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. How to backdoor diffusion models ? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4015–4024, 2023. 95

[131] Shengfang Zhai, Yinpeng Dong, Qingni Shen, Shi Pu, Yuejian Fang, and Hang Su. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. *arXiv preprint arXiv:2305.04175*, 2023. 96

[132] Lukas Struppek, Dominik Hintersdorf, and Kristian Kersting. Rickrolling the artist: Injecting invisible backdoors into text-guided image generation models. *arXiv preprint arXiv:2211.02408*, 2022. 96

[133] Yihao Huang, Qing Guo, and Felix Juefei-Xu. Zero-day backdoor attack against text-to-image diffusion models via personalization. *arXiv preprint arXiv:2305.10701*, 2023. 95, 96

[134] Chao Gong, Kai Chen, Zhipeng Wei, Jingjing Chen, and Yu-Gang Jiang. Reliable and efficient concept erasure of text-to-image diffusion models. In *European Conference on Computer Vision*, pages 73–88. Springer, 2024. 101