The Pennsylvania State University
The Graduate School

# THE GOOD, THE BAD AND THE UGLY: EXPLORING THE ROBUSTNESS AND APPLICABILITY OF ADVERSARIAL MACHINE LEARNING

A Dissertation in
Information Sciences and Technology
by
Xiaoting Li



Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2022

The dissertation of Xiaoting Li was reviewed and approved by the following:

Dinghao Wu
Professor of Information Sciences and Technology
Dissertation Advisor
Chair of Committee

C. Lee Giles
Professor of Information Sciences and Technology

Ting Wang
Assistant Professor of Information Sciences and Technology

Jinchao Xu
Professor of Mathematics
Director of Center for Computational Mathematics and Applications

Mary Beth Rosson
Professor of Information Sciences and Technology
Director of Graduate Programs

# Abstract

Neural networks have been widely adopted to address different real-world problems. Despite the remarkable achievements in machine learning tasks, they remain vulnerable to adversarial examples that are imperceptible to humans but can mislead the state-of-the-art models. More specifically, such adversarial examples can be generalized to a variety of common data structures, including images, texts and networked data. Faced with the significant threat that adversarial attacks pose to security-critical applications, in this thesis, we explore the good, the bad and the ugly of adversarial machine learning. In particular, we focus on the investigation on the applicability of adversarial attacks in real-world scenarios for social good and their defensive paradigms.

The rapid progress of adversarial attacking techniques aids us to better understand the underlying vulnerabilities of neural networks that inspires us to explore their potential usage for good purposes. In real world, social media has extremely reshaped our daily life due to their worldwide accessibility, but its data privacy also suffers from inference attacks. Based on the fact that deep neural networks are vulnerable to adversarial examples, we attempt a novel perspective of protecting data privacy in social media and design a defense framework called Adv4SG, where we introduce adversarial attacks to forge latent feature representations and mislead attribute inference attacks. Considering that text data in social media shares the most significant privacy of users, we investigate how text-space adversarial attacks can be leveraged to protect users' attributes. Specifically, we integrate social media property to advance Adv4SG, and introduce cost-effective mechanisms to expedite attribute protection over text data under the black-box setting. By conducting extensive experiments on real-world social media datasets, we show that Adv4SG is an appealing method to mitigate the inference attacks.

Second, we extend our study to more complex networked data. Social network is more of a heterogeneous environment which is naturally represented as graph-structured data, maintaining rich user activities and complicated relationships among them. This enables attackers to deploy graph neural networks (GNNs) to automate attribute inferences from user features and relationships, which makes such privacy disclosure hard to avoid. To address that, we take advantage of the vulnerability of GNNs to adversarial attacks, and propose a new graph poisoning attack, called AttrOBF to mislead GNNs into misclassification and thus protect personal attribute privacy against GNN-based inference attacks on social networks. AttrOBF provides a more practical formulation through obfuscating optimal training user attribute values for real-world social graphs. Our

results demonstrate the promising potential of applying adversarial attacks to attribute protection on social graphs.

Third, we introduce a watermarking-based defense strategy against adversarial attacks on deep neural networks. With the ever-increasing arms race between defenses and attacks, most existing defense methods ignore fact that attackers can possibly detect and reproduce the differentiable model, which leaves the window for evolving attacks to adaptively evade the defense. Based on this observation, we propose a defense mechanism that creates a knowledge gap between attackers and defenders by imposing a secret watermarking process into standard deep neural networks. We analyze the experimental results of a wide range of watermarking algorithms in our defense method against state-of-the-art attacks on baseline image datasets, and validate the effectiveness our method in protesting adversarial examples.

Our research expands the investigation of enhancing the deep learning model robustness against adversarial attacks and unveil the insights of applying adversary for social good. We design Adv4SG and AttrOBF to take advantage of the superiority of adversarial attacking techniques to protect the social media user's privacy on the basis of discrete textual data and networked data, respectively. Both of them can be realized under the practical black-box setting. We also provide the first attempt at utilizing digital watermark to increase model's randomness that suppresses attacker's capability. Through our evaluation, we validate their effectiveness and demonstrate their promising value in real-world use.

# Table of Contents

# List of Figures

# List of Tables

# Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Dinghao Wu for his valuable advice, continuous support, and patience during my PhD study. Without his help, I would never be able to achieve this milestone. His immense knowledge and inspiration bring significant impact both on my growth and my research and lead me to become a more mature and rigorous researcher.

I would also like to thank the rest of my committee, Prof. C. Lee Giles, Prof. Ting Wang, and Prof. Jinchao Xu for their valuable comments and insightful suggestions, but also for the questions that help me improve my study and dig deeper into my research from different perspectives.

I feel so lucky to have a group of talented labmates in our research team. I would like to thank them for the numerous inspiring discussions and their kind helps. Without them, I would not be able to overcome so many difficulties by myself. In particular, I would like to offer my special thanks to Dr. Lingwei Chen, for the invaluable help on my research and the sleepless nights we were working together on each project.

Last but not least, I would like to express my deeply gratitude to my family, especially my parents and my sister who always provide unconditional support on my life and my research career. Thanks to their encouragement, I can stick to my goals and achieve all the results with perseverance in PhD study.

The dissertation is based on research supported in part by a seed grant from the Penn State Center for Security Research and Education (CSRE) and the PNC Technologies Career Development Professorship. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

# Chapter 1 Introduction

## 1.1 Background

Deep learning has emerged as a strong framework that can be applied to a broad spectrum of complex machine-learning (ML) tasks, ranging from computer vision [3–5], speech recognition [6, 7] to natural language processing [8, 9] and healthcare [10]. With the evolution of deep learning models and availability of high performance hardware for complex computation, deep learning has made a remarkable progress in many traditional fields and achieved unparalleled accuracy to benefit people's daily life. Because of the continuing advancement of deep learning techniques, extensive use of deep learning based applications can also be seen in safety and security-critical environments, such as self driving cars [11–13], malware detection [14–16], robotics [17,18] and etc. As deep learning methods have found their way to being applied to real world, security and integrity of the applications have attracted lots of attention [19].

Despite the remarkable achievement of deep learning, researchers found that deep neural networks remain vulnerable to adversarial attacks that design special imperceptible perturbations to the original samples to fool state-of-the-art models. In [20], Szegedy et al. first revealed a quite astonishing view on neural networks that contradict commonly held beliefs: there exists a 'blind point' in neural networks in the sense that some imperceivable input perturbations to human eyes can fool the well-trained ML models with high confidence [21]. Formally, given a valid input x and a target $t \neq F(x)$, it is often possible to find a similar input $x'$ such that $F(x') = t$ yet $x$ and $x'$ are close according to specific distance metric, which is used to quantify the similarity between original and adversarial examples. This type of crafted inputs are referred to as adversarial examples and the methods to generate such adversarial inputs are called adversarial attacks in ML. Adversarial examples pose severe security issues to the deep learning systems, especially

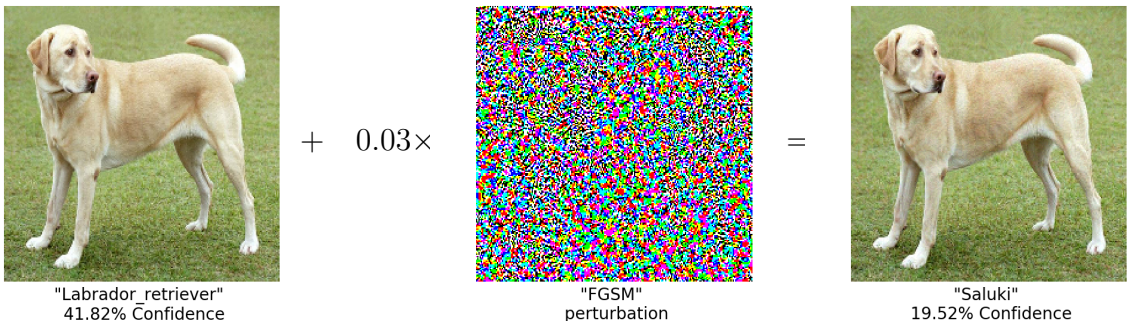



Figure 1.1: A demonstration of adversarial example generation applied to [1] on a sample image. By adding a small perturbation whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input, we can fool a model pretrained on ImageNet to change the classification result. Here .03 corresponds to the magnitude of the perturbation we introduce to the original image.

in safety sensitive areas. For example, Goodfellow et al. [1] demonstrated how to add a small perturbation to an image of panda that causes it to be recognized as a gibbon with high confidence. In a security-critical scenario, Evtimov et al. [22] successfully misled a classifier to misclassify a stop sign with some physical perturbations, which can be either the graffiti or black and white strips, as a Speed Limit 45 sign. See Figure 1.1 for a demonstration on a sample image of a Labrador Retriever from [1]. In the figure, the left image is correctly classified as a Labrador Retriever by the model. However, when we add a small crafted perturbation to the original image and then obtain the right image, the model can be mislead to classify it as a Saluki, even though there is no recognition difference for humans.

With adversarial examples attracting lots of attention, a quite number of adversarial attacks are proposed to attack sophisticated deep learning models to bypass the model protections. Szegedy et al. [20] first defines the problem as a constrained optimization problem and finds the adversarial example through linear search. By contrast, Fast Gradient Sign Method (FGSM) [1] is designed in a fast way that performs one-step gradient update along the direction of the sign of gradient at each pixel. Followed by this work, there arise lots of extensions [23–25] to upgrade the attacking capability. DeepFool proposed in [26] aims to find the minimal Euclidean distance between the adversarial example and the original input. Carlini and Wagner [27] launched C&W

[1]https://commons.wikimedia.org/

attack to defeat many existing adversarial detecting defenses(XXX). The existence of adversarial attacks has motivated proposals and evolutions for approaches that increase the robustness of DNNs against adversarial examples. The majority of countermeasures towards adversarial examples fall into two categorities: 1) *proactive:* improve models' robustness against adversarial attacks and 2) *reactive:* detect malicious samples before importing them into well-trained model. For the first type of methods, they tend to manipulate model properties such as invariance through regularization scheme [23, 28, 29] to make it harder to craft new adversarial examples. Data augmentation is another typical way to improve model's robustness by applying a couple of label-preserving transformations, such as random cropping, flipping [5, 30], masking out [31] or adding Gaussian noise [32]. Adversarial training is a effective way to straightforwardly incorporate new crafted adversarial examples into the retraining venue to enforce model to recognize these malicious outliers [1] correctly. Although this type of method can significantly improve model robustness against adversarial examples, how to employ it without hurting clean data accuracy is under-explored. Besides, it requires more heuristics to determine adversarial samples in the retraining pool. For the second category, data preprocessing-based defenses are designed to filter out malicious samples or remove the modifications introduced to the regular image in the testing stage [33–38]. Most of these work turns out not reliable and have proved to be defeat by [27] with slight changes of loss function.

Despite the robustness of neural networks is greatly improved, these defense methods fall far behind the arms race against the continuously evolving attacks. Most of these strategies are easy to compromise due to their simplicity and differentiable nature, with some impractical assumptions about the attacker's knowledge of the target model. These weaknesses leave the window for attackers and stimulate the generation of more capable attacks to evade previously defenses. To reduce the dependency on targeted model, the attacker even performs black-box attacks with little model knowledge and can succeed under the extreme limited scenario [39, 40]. Therefore, more investigation and exploration on how to alleviate aforementioned challenges are required.

On the other hand, with the fast development of deep learning techniques and their high performance on feature representation and pattern recognition, more and more powerful tools are provided for researchers to do public data analysis and community studies. However, they can also benefit the attackers for malicious purposes. This poses significant threat to the data privacy. Especially in the age of Big Data, the worldwide accessibility to the social media has drastically reshaped the world and allowed billions of people all around the globe to conveniently perform numerous activities such as creating

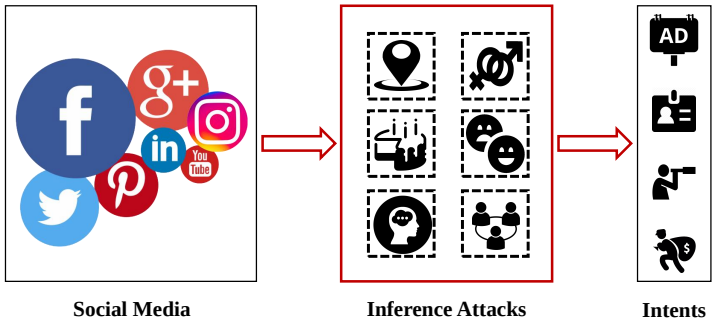



Figure 1.2: Increasingly evolving inference attacks

online profiles, sharing personal information, and interacting with other people [41]. Social media creates a heterogeneous environment with rich source of user-oriented data, which attracts not only researchers for legitimately studying and understanding social communities and individuals, but also attackers for infiltrating users' information and inferring their sensitive and private attributes (e.g., age, gender, location, opinion, etc.) to deliberately fulfill the economic, social, or political intents (e.g., stealing user credentials, promoting unwanted advertisements, stalking and threatening users) [41–43], which is illustrated in Figure 1.2. In particular, in the domain of social media data privacy, inference attacks are increasingly deployed to reveal users' private information from public data on social media [44–47]. As such, plausible interventions are urgently needed to address these privacy concerns.

As machine learning, especially deep learning, provides more and more powerful tools for data analytics, it is increasingly deployed to learn latent feature representations from raw data and thus perform automated attribute inferences in social media [44, 45, 47–52], which enables considerable countermeasures to evade the traditional privacy protection techniques, e.g., anonymization. In the meanwhile, as the adversarial attacks contiguously evolve and make marvelous progress in the arms race with defenders, it is worth thinking that if we can apply such cutting-edge knowledge to some meaningful tasks which do social good instead. Also, inspired by the valuable observation of adversarial machine learning that neural networks are vulnerable to adversarial attacks, we explore the applicability of leveraging inherent learning challenge the model for protecting the privacy of data with different structures. That is, we cast the problem of protecting social media privacy as an adversarial attack formulation problem to defend against attribute inference attacks.

## 1.2 Motivation

With the everlasting development of adversarial attacks posing more and more challenges to deep learning model robustness, while we invent new promising strategies to fix the security issue, it also provides us with possibilities to investigate potential opportunities of taking good advantage of adversarial examples in different scenarios rather than just attacks. Social media has been enjoying explosive growth for a decade. Due to its penetration, accessibility, and information richness, user-oriented data generated from social media attracts not only developers for legitimately studying social communities to better meet user needs, but also attackers for inferring users' privacy information. In particular, more and more powerful ML tools are provided that attackers can leverage to realize their sophisticated inference purposes.

In order to mitigate machine learning-based inference attacks, some potential paradigms have been developed, including game-theoretic optimization [41, 53–56], differential privacy and its variant local differential privacy [57–61], and deep data obfuscation [62, 63]. While existing research results are encouraging, most of these methods are either cost-expensive, or leading to large data utility loss, which are not feasible in practical use. To bridge the gap between user experiences and legitimate application functions, we may choose a trade-off between privacy and utility: protecting social media user attributes from inference assaults while still enabling the data that users are willing to provide to be plausible. In the meantime, machine learning models are faced with the inherent learning-security challenge of lacking adversarial robustness [1, 20] despite their impressive inference abilities, where they are vulnerable to adversarial attacks that carefully design imperceptible perturbations to the input data, and thus taint the latent feature representations and drastically degrade the corresponding inference performance. Inspired by this observation that adversarial attacks are capable to fool the attribute inference learning models into misclassification in a computationally tractable fashion with small utility loss [41], in this thesis, we cast the problem of protecting social media data privacy as an adversarial attack formulation problem over the data to defend against attribute inference attacks. More specifically, adversarial attacks are formulated as a counterpart to enforce attribute inference attacks as less effective as possible, which in return help improve the resilience and obfuscation of the social media data and thus reduce the risk and possibility of privacy breach.

As the social media is quite a complicated environment that involves a variety of different forms of data, it is difficult to simply consider the data privacy protection

problem as one general defense setting. We design our attribute protection methods from diverse scenarios targeting on difficult specific data forms, such as texts, graphs. Besides, a few recent works [41, 47, 64, 65] showed that adversarial attacks have been starting to be leveraged as defenses against inference attacks, which present great potentials for data obfuscation. However, the prior works of this kind focus on the specific application scenarios where their target is limited to continuous space. The investigation into more challenging social media environment and the corresponding data of more diverse properties (e.g., discrete text, graph structure) has been scarce. To assist with validating the feasibility of turning adversarial attacks into protection for social media privacy, in this thesis, we propose frameworks of integrating adversarial attack techniques in user attribute protection tasks on the basis of discrete texts and social graphs in social media, respectively.

From the perspective of resolving the adversarial threats, the everlasting arms race between adversarial attacks and defenses in adversarial machine learning poses more and more challenges for enhancing the robustness of ML models. Although a large body of defense methods are proposed to protect DNNs against adversarial attacks, they fall far behind of the competition with the fast growth of adversarial attackers and fail to defend all type of adversarial examples. Due to the simplicity and differentiable nature, most of these methods are straightforward to compromise. Also, the efficacy of many defenses rely on some unreasonable assumptions about the attacker's knowledge on the target model. For instance, defenders can always achieve better protection results in the black-box scenarios where the attackers only have limited accessibility on the targeted model or system. However, it is hard to assume the attacker's capability as they have so many choices on the attacking methods and also are able to leverage model extraction techniques to evade the protection. In reality, the information about the target model is the key for most attack algorithms to craft adversarial examples, especially for those gradient-based attacks that require this information to calculate gradients through backpropagation. In this respect, we aim to find a way to either reduce the knowledge of attackers on the protected model or increase the difficulty of them to access the information about model components.

Randomization of the network layerwise structure or inputs has been discussed in a few studies to have the potential to obfuscate gradients information and thus decrease adversarial vulnerability as the attackers can hardly reproduce the gradients. This naturally encourages us to use the randomization paradigm to raise the attacker's uncertainty about the target model, preventing them from modifying the model information and

rendering the generated adversarial examples as ineffective as feasible. In our work, we propose a practical defense framework by introducing input randomness to DNNs with the digital watermarking techniques in the context of image classification. Digital watermarking is a technique that embeds watermark information into the host image by modifying visually non-significant pixels, which is transparent, imperceptible, and robust. For the watermarking techniques, if a user has no embedding information, the watermark is very challenging to be detected and extracted [66]. In this respect, the attacker needs to craft adversarial examples from their self-trained surrogate models as it is not realistic for them to reproduce the defense model without confidential embedded information. The lack of knowledge about the defense system leads to the discrepancy and stochasticity between the surrogate and real models, making it more challenging for the attacker to successfully evade the defense model. Our proposed defense method enables us to train a DNN model that would not only preserve the inference performance on regular data, but also benefit from knowledge gap and randomization imposed on the learned protocol for better robustness against adversarial attacks.

## 1.3 Research Goals

In this section, we first summarize our research goals over our study and then provide a quick tour of our three research work in this dissertation. We present our work from two perspectives; (1) the Good: explore the applicability of adversarial machine learning and (2) the Bad: enhance the robustness of deep neural networks against adversarial attacks. In the first two work, we explore and discuss the applicability of adversarial machine learning. Both of them seek to intervene in social media privacy threats, and gain deeper insight how adversarial attacks can serve as protection to obfuscate users' attributes. In particular, the first work focuses on the protection of language models in natural language processing (NLP) domain, while the second work considers the social graph privacy in more complicated networked space to achieve a good balance between data privacy and utility. Faced with the challenges in adversarial machine learning, the third work we present is a defense method against adversarial attacks in the domain of image classification.

### 1.3.1 The Good: Exploring the Applicability of Adversarial Attacks

The development of adversarial attacking techniques and the vulnerability of deep learning models provide us with new angle in taking advantage of adversarial examples. In particular, we aim to explore how to apply adversarial attacks to social good scenarios, i.e., data privacy protections. With the development of deep learning, it provides more and more convenient tools that benefit attackers to attack data privacy in social media. Based on the observation that deep learning systems are vulnerable to carefully-designed adversarial examples, we want to find a way to convert the problem of protecting user's data privacy as an adversarial attack formulation problem over the social media. In this respect, the other research goal in this dissertation is to explore the applicability of adversarial attacks for social good.

#### 1.3.1.1 Adv4SG: Text-based Adversarial Attack for Attribute Privacy

In social media environment, users tend to post text data for sharing; such text data may indicate their sensitive information, and thus easily expose the users to the attackers who can access the texts and infer the private attributes of interest to fulfill the harmful intents [64]. Our goal in this work is to protect users' private attribute against such inference attacks. Specifically, we take advantage of the vulnerabilities of deep language models to adversarial examples and design a cost-effective end-to-end framework to automatically modify the social media textual post to mislead the inference attackers. Our key intuitions are that: (1) text data in social media share the significant information of users' privacy for protection; (2) deep neural networks are widely used by attackers among those powerful attribute inference attacks [67–70]; (3) Such learning models have been shown to be vulnerable to adversarial attacks. We briefly go through the method design in the following parts.

**Private Attribute Recognition.** As we put our framework under the practical black-box setting, where the devised adversarial attack is not aware of the threat model architecture, parameters, or training data, but capable of querying the threat model with text inputs and retrieving the output predictions for the attributes and their confidence scores [71]. Based on the defined threat model, we will first build the NLP-based inference attack model on the collected data that is able to recognize attribute labels $y^i = \{v_1, v_2, \cdots, v_k\}$ for each private attribute from the set $\mathcal{Y} = \{y^1, y^2, \cdots, y^m\}$. In particular, the semi-automatic attribute inference subroutine is to collect users' public data under each attribute firstly; then build language models using deep neural networks

over the collected data by minimizing the inference errors on labeled data. Consequently, we can apply the trained attribute recognition model to new users and perform inference attacks.

**Leverage Adversarial Examples as Defense against Inference Attacks.** Given an inference attack target (i.e., one attribute to infer), we formulate text-space adversarial attacks as defenses that attempt to automatically perturb the texts to obfuscate that attribute and prevent threat models from correctly identifying their private attribute values. As aforementioned, we consider the black-box setting such that our formulation is applicable to evade a wide range of attribute inference models. Formally, for an original text $\mathbf{x}$, the purpose of a text-space adversarial attack is to modify $\mathbf{x}$ with assigned label $y^t$ to a text $\hat{\mathbf{x}}$ that is classified to any other label $\widehat{y^t} \in \mathcal{Y}^t$, $\widehat{y^t} \neq y^t$ through adding a perturbation $\delta$. In black-box settings, exiting gradient-based methods are no longer eligible to compute perturbations in the feature space. In addition, to formulate a feasible text-space adversarial attack, we have to comply with some essential constraints on the modification of the texts.

These are two key challenges that we aim to solve in our method design. Faced with the first challenge, we leverage a genetic algorithms to design a method called **Adv**ersarial attack for **S**ocial **G**ood, called *Adv4SG*, to protect personal attribute privacy against NLP-based attribute inferences over social media text data. It exploits population-based gradient-free optimization which releases the limitation on gradient information. Moreover, we self-train a surrogate model to mimic the attack model and take advantage of the transferability [72] of adversarial examples to conduct the protection. To resolve the second challenge, we design a series of constraints taking account of text syntax, semantics, user preferences and etc. In this regard, we construct a sequence of plausible perturbations to automatically craft the adversarial text with preserved semantics.

**Attribute-obfuscating Attack for Social Networks' User Privacy.** In real world, social media is composed of complex networked data rather than single data format of texts or images. For such graph-structured data, graph neural networks have shown the great potential in learning and integrating its hidden information. Specifically, graph convolutional networks (GCNs) [73] take the connectivity structure of the graphs as the filter to perform neighborhood information aggregation so as to extract high-level features from the nodes and their neighborhoods, which have thus boosted the state-of-the-arts for a variety of tasks (e.g., node classification, clustering, and matching) over graphs. In this work, we adopt graph neural networks to model the networked data in social media for better learning tasks, and investigate how to jointly leverage adversarial attacks

against graph neural networks to protect social media data privacy. Our key intuitions are that: (1) social media is a complex heterogeneous data environment that is composed of diverse graph-structured data; (2) graph neural networks have high capability in representation learning of networked data and provide benefit for attackers to conduct more sophisticated inference attacks on social networks; (3) in social media, a large amount of data labels are expensive and impractical to collect, while GCNs can conduct semi-supervised learning to solve node classification task where only a small number of nodes are labeled; (4) graph neural networks are vulnerable to well designed adversarial attacks.

#### 1.3.1.2 AttrOBF: GNN-based Adversarial Attack for Attribute Privacy

Inspired by its great success of graph neural networks on representation learning of the networked data, we will apply semi-supervised learning using GNNs to recognize social relation attributes over graph-structured data. More specifically, GNNs can be denoted as:

$$Z = \text{softmax}\left(\tilde{A}^{(l)} \cdots \ \sigma\left(\tilde{A}^{(1)}\mathbf{X}^{(0)}\mathbf{W}^{(1)}\right)\cdots \mathbf{W}^{(l)}\right) \tag{1.1}$$

where at layer $i$, $\mathbf{X}^{(i)}$ is feature matrix, $\mathbf{W}^{(i)}$ is weight matrix, $A^{(i)}$ is adjacency matrix, $\tilde{A}^{(i)} = D^{-\frac{1}{2}}(A^{(i)} + I)D^{-\frac{i}{2}}$, and $D$ is the diagonal degree matrix of $A^{(i)} + I$. To train this model to recognize attribute $y^i = \{v_1, v_2, \cdots, v_k\}$ for data $D$, the softmax function normalizes the final output matrix $Z$, where each row represents the probability of $k$ labels for a node. The cross-entropy loss $\mathcal{L} = -\sum_{d\in\mathcal{D}_{tr}} \log Z_{d,v_d}$ can be accordingly evaluated between the output and the corresponding ground truth, while the weights are updated using some gradient descent optimization algorithms. The unknown data will be automatically labeled after training. Considering that social networks are generally represented as graph-structured data, in this work, we assume that the attackers would take advantage of user features and relationships to train GNN models so as to achieve their attribute inference goals. Correspondingly, adjacency matrix $A^{(i)}$ represents the interactions between different users, which can be the follower-followee relationships in social media or distance relationships measured by some similarity metric. Input $\mathbf{X}$ indicates the user features including profiles, images, textual posts and etc, while the model outputs are the private attribute that the attacker aims to predict.

**Adversarial Obfuscation Attack on Social Graph.** In this work, we design a graph adversarial poisoning attack AttrOBF to protect attribute privacy against GNN-based inferences on social networks. The goal of our method is to shift a small fraction of

optimal training users' labels so as to maximally decrease the overall performance of GNN-based attribute inferences trained on the modified graph. That is, given a target attribute with either binary or multiple labels, the goal is to have the test users classified as any label different from the true label.

To achieve that, we need to solve a couple of challenges in designing such an applicable framework. The first challenge is that a large amount of training set with labels are not available. Traditional graph neural networks cannot work due to the lack of labeled data in real-world scenarios. However, GNNs such as graph convolutional network (GCN) can conduct semi-supervised inference training on a small set of nodes with labels. Second, we have no information about the GNN model used by inference attackers, including model choice, architecture, and parameters. To defend against them, we self-train a proxy model to substitute the attacker and optimally identify the node labels to modify. It is worth noting that using a surrogate model to simulate attackers under the black-box setting is popular in previous study. The third challenge is that our attribute-obfuscating attack on GNNs is essentially a bi-level optimization problem, where an outer optimization involves another inner optimization as constraint. This bi-level problem is however non-convex and intractable to solve [74]. To solve this challenge, we utilize the approximation of sub-model and transform the bi-level problem to single level. Last but not least, the training label data and the action space of the label perturbation are discrete, which prohibits us from computing the gradients through back-propagation as the label operations are non-differential. In this regard, we use a sampling method called Gumbel estimator [2] to approximately use continuous components to substitute the discrete components during optimization.

### 1.3.2 The Bad: Enhancing the Robustness of DNNs Against Adversarial Attacks

As the existence of adversarial attacks poses severe threat to the security of deep neural networks, it is critical for us to design more secure deep learning systems. The existing defense methods are not able to effectively protect against all kinds of adversarial examples due to their simplicity or impractical assumptions. While among the proposed adversarial attack methods, the knowledge of attacker to target models tends to be the key of attacking success. With this in mind, our research goal in this dissertation is to improve the model's privacy by restricting the attacker's knowledge and design defense methods to correspondingly enhance the robustness of DNN models.

#### 1.3.2.1 Watermarking-based Defense over DNNs

As adversarial attacks pose significant threat to the safety of deep neural networks in image classification domain, in this work, we consider the most practical scenario about adversarial attacks and propose a defense method by introducing random watermark information to DNN models to incur knowledge gap between the attacker and the defender. Our key intuitions underlying our design are that: (1) attackers' knowledge disadvantage over the target model can restrict their attacking capabilities, especially for the mainstream gradient-based attacks; (2) leveraging some randomization paradigm on model or data can potentially increase attackers' uncertainty and model's confidentiality; (3) digital watermarking is a technique that embeds secret watermark information into the host image by modifying visually non-significant pixels, which is transparent and challenging for the attacker to detect and obtain.

**Generate Knowledge Gap through Watermarking Input.** Digital watermarking can be defined as a practice of undetectably altering a work to embed a secret message. In this technique, the secret payload (i.e., watermark) is embedded in multimedia elements using specific watermarking algorithm that should be invisible and robust. In our framework, we insert a designed watermarking subroutine to the regular DNN architectures. The purpose is to introduce the secret watermarking information to create such knowledge gap between the attacker and the defender. In our watermarking subroutine, we would encode an input using some secret watermark message which serves as the encoding key to prevent eavesdroppers to decode the watermarked message [75]. Therefore, in our method, adversaries are unable to extract the embedded information and cannot reproduce the defense system to adaptively craft adversarial example targeted the defense model. To contiguously increase the model uncertainty, we randomly divide the training set into different parts and embed each part of them with different key images. In this respect, the lack of knowledge about the watermarking process enlarges the discrepancy between the attacker and the defender, and make it difficult for the attacker to circumvent the defense strategy.

More specifically, in most real-world scenarios, when crafting adversarial examples, the attacker cannot customize the defense model directly due to the limited access but instead has to use their own trained model. As investigated, adversarial examples may be transferable, so that some adversarial examples generated for a model may cause misclassification on another model as well [72]. Such a property allows the attacker to train a model as a surrogate model by themselves, the purpose of which is to imitate and replace the target model to craft attack samples. However, the surrogate model is a

rough approximation of the target distribution. There is always a discrepancy between the approximation and the real one, which we consider as our defense space and our goal is to irreversibly enlarge such space. The introduced secret watermarking module expands the discrepancy and make it difficult for the attacker to circumvent the defense using the adversarial examples generated from the surrogate model. That is to say, the watermarking procedure prevents the attacker from customizing the defense model, and the DNN model embedded with secret watermarking information may explicitly change its classification boundary, and thus be resilient against the attacker's generated adversarial examples.

## 1.4 Thesis Organization

In this thesis, we mainly focus on enhancing the robustness of deep neural networks and investigating the applicability of adversarial attacks for social good. Correspondingly, we present three of our research work. The first two work explore the applicability of adversarial attacks for social good in different fields. The first one applies textual adversarial examples to the social media data privacy. In the second work we present, we discuss the practicability of using adversarial attacks in social networks attribute protection. The third work takes into account the bad impact that adversarial attacks bring to machine learning securities, we present is a defense technique which protects the deep neural networks against adversarial attacks in the context of image classification.

The rest of the thesis is organized as follows. We present the related work in Chapter 2. Chapter 3 discusses the good role the adversarial attacks play in different privacy protection scenarios in social media, including textual data privacy protections and networked data privacy protections; while Chapter 4 shows the detailed design of a defensive watermarking-based framework against adversarial attacks over DNNs from perspective of resolving the bad threat of adversaries. In Chapter 5, we discuss the limitations of our work and the challenges ahead. Finally, we conclude the thesis in Chapter 6.

# Chapter 2
# Related Work

In this chapter, we first review the related work about existing adversarial attacks and defenses in the domain of image classification. Then we discuss the related work in the perspective of using adversarial attacks for social good. Specifically, we show the existing work about inference attacks and defenses of data privacy-perserving in social networks. Also, We briefly review the literature of adversarial attacks in discrete textual space as well as the structural graphs.

## 2.1 Adversarial Machine Learning

Adversarial machine learning has been attracted lots of attention recent years due to the threats it pose to our day-to-day applications. It considers those scenarios when machine learning systems may face potential adversarial attacks, who intentionally manipulate regular input data with small perturbations to mislead the well-trained model to make mistake. The earliest research discuss adversarial machine learning on a more general area that reveals and resolves the security issue in machine learning systems. However, recent studies place more emphasis on how those carefully crafted imperceptible perturbations on the inputs may lead to drastic mistakes in deep learning fields. In our thesis, the problems we pay attention to are also referred to cases in the later scenarios. To protect deep learning models against adversarial attacks, a large body of methods are proposed to resolve the security challenges. However, the existence of defenses also stimulates the evolution of attacking methods.

### 2.1.1 Attacks

To date, a taxonomy of adversarial attacks has been proposed. There are different ways to categorize these attacks. Usually, adversarial attacks can be defined as targeted and non-targeted attacks. The central confusion between two settings is whether the attack targets on a particular input or output [76]. One the other hand, they also can be divided into white-box and black-box attack according to the knowledge that the attacker possesses about the classifier. In a white-box attack [1, 20, 26, 77], the attacker has full access to the targeted model to craft adversarial examples, which is not available to black-box attackers. In this type of attack methods, the attacker would be able to find gradient information of the model with respect to the inputs to craft adversarial examples. Although white-box attacks can be very effective, the settings that they can have full knowledge of the classifier may seem impractical in most real-world scenarios. Black-box attacks are proposed to solve such challenges. As the attacker cannot directly obtain the information needed due to the inaccessibility, he would take advantage of the transferability of the adversarial examples to make attacks possible. Transferability is an astonishing property observed by early studies that an adversarial example for model can often transfer to be an an adversarial on a different model [1, 20, 72, 76].

In particular, Szegedy et al. [20] first found adversarial examples against DNNs in 2014. They proposed to use L-BFGS method to generate adversarial examples. This method defines the problem as a constrained optimization problem and finds the adversarial example of minimum distance through time-consuming linear search. By contrast, Fast Gradient Sign Method (FGSM) [1] is designed to be fast to find adversarial examples. They only performed one-step gradient update along the direction of the sign of gradient at each pixel. Accordingly, there aroused a series of variants to boost adversarial attacks [23–25, 78], among which iterative FGSM proposed by Kurakin et al. [23] is considered as a stronger extension of this attack. It introduces a finer optimization for multiple iterations of FGSM and can be applied to physical world directly. To change a small portion of the sample instead of updating the whole input, Papernot et al. [77] designed a Jacobian-based saliency map attack (JSMA) to only perturb the features of inputs that made most significant changes to the output. DeepFool proposed in [26] aims to find the minimal Euclidean distance between the adversarial example and the original input by iteratively projecting the input $x$ onto the nearest class boundaries. However, it's computationally expensive due to the sophisticated formulation. Carlini and Wagner [27] launched C&W attack to defeat almost all of existing adversarial detecting defenses.

### 2.1.2 Defenses

The existence of adversarial attacks has motivated proposals and evolutions for approaches that increase the robustness of DNNs against adversarial examples. The majority of countermeasures towards adversarial examples fall into two categorities: 1) *proactive:* improve models' robustness against adversarial attacks and 2) *reactive:* detect malicious samples before importing them into well-trained model.

For the first type of methods, they tend to manipulate model properties such as invariance through regularization scheme [23, 28, 29] to make it harder to craft new adversarial examples. Regularization is a standard practice to penalize the model complex in machine learning. Data augmentation is another typical way to improve model's robustness by strengthening the training process. Usually, people can apply a couple of label-preserving transformations to expand the training set, such as random cropping, flipping [5, 30], masking out [31] or adding Gaussian noise [32]. Adversarial training is a effective way to straightforwardly incorporate new crafted adversarial examples into the retraining venue to enforce model to recognize these malicious outliers [1, 23] correctly. Although this type of method can significantly improve model robustness against adversarial examples, how to employ it without hurting clean data accuracy is under-explored. Besides, it requires more heuristics to determine adversarial samples in the retraining pool.

For the second category, data preprocessing-based defenses are designed to filter out malicious samples or remove the modifications introduced to the regular image in the testing stage [33–38]. Despite the great efforts, these proposed defending techniques, though were claimed to be robust, have already been verified vulnerable. For instance, Carlini and Wagner [27] conclude that these detection-based adversarial learning techniques are not reliable and have proved to be defeat by C&W attack with slight changes of loss function.

## 2.2 Adversarial Attacks for Social Good

The arms race between attacks and defenses in adversarial machine learning is still severe. Our pursue of more powerful attacking and defending strategies may form an infinite loop [76]. While we devote ourselves to the absolute security and totally clear up the risk of adversarial examples, it is also worth studying and exploring the usage of adversarial learning in some areas. This naturally leads to our research question: can we

take advantage of the superiority of adversarial attacks and apply them to applications for social good?

### 2.2.1 Inference Attacks and Defenses

Social media has been enjoying explosive growth for a decade. Such a complex user-oriented environment contains luxuriant information, which is why it is always the target to inference attackers. In inference attacks, the attacker can infiltrate personal attributes that people are unwilling to disclose from the public data for their malicious purposes. Such attacks on attributes such as gender, political views, and religious views have been studied in decades [44, 79]. To protect the user-oriented private data, various protection techniques have been proposed to protest inference attacks. Anonymizations [43,64,80–84] have been conventionally developed to anonymize and protect user identifiable information on social media. However, they are inefficient and impossible to anonymize the information of all aspects due to the unprecedented increasing levels of social interactions and the enforced utility loss [42, 85, 86]. while they are still vulnerable to specific types of data leakage [87, 88]. Some works focus on obfuscating users' interactions by studying the relationship between privacy and utility to hide their actual intentions and prevent profiling [89, 90].

Unfortunately, as machine learning, especially deep learning, provides more and more powerful tools for data analytics, it is increasingly deployed to learn latent feature representations from raw data (anonymized or not) and thus perform automated attribute inferences in social media [44, 45, 47–52], which enables considerable countermeasures to evade the static and straightforward anonymization techniques. Regarding to this, some promising defense methods have been thus presented to alleviate such inference attacks, such as differential privacy [61], deep data obfuscation [62], and game-theoretic optimization [41, 54], but they are still suffering from limitations of either cost-expensive, large utility loss, or introducing additional privacy concerns.

Recent studies also validate that privacy-conscious federated learning is threatened by various inference attacks [91–93]. In this respect, some promising paradigms have been accordingly presented to alleviate machine learning based inference attacks, including game-theoretic optimization [41, 53–56], differential privacy and its variant local differential privacy [57–61], and deep data obfuscation [62, 63]. While existing research results are encouraging, most of these methods are either cost-expensive, or leading to large data utility loss, which are not feasible in practical use. As the development of adversarial learning, some recent works [41, 47, 64, 65] started to leverage adversarial

attacking techniques as protection strategies to defend against inference attacks and revealed great potentials of them for data obfuscation.

### 2.2.2 Adversarial Attacks in Text Domain

Adversarial examples are firstly found in the domain of image classification. Therefore, most early studies focus on conducting adversarial attacks in the continuous space. Later on, in the field of natural language processing (NLP), deep learning models are revealed to suffer from the similar vulnerability to adversarial examples.

A bunch of adversarial methods are proposed to craft textual adversarial examples. Papernot et al. [94] firstly attempt to use a white-box gradient-based attack inherited from the strategies from image domain to repeatively modify the input sequence until trick deep neural text classifiers. Ebrahimi et al. [95] propose to change one word token to another using the gradients of the model with respect to the input. Some methods pay more attention to the design of perturbation rules involved in the adversarial text generation. For instance, Samanta et al. [96] design heuristic driven rules to find the close words to substitute the original word tokens. People also find those out-of-vocabulary words can be effective in misleading DNN models [97–99]. Usually, they can be mapped to "Unknown" vectors in the embedding space and thus introduce damage to the semantics and syntax of texts. Apart from the token-level perturbations, [97–99] show that NLP models can be attacked through different scale of character-level manipulations. However, this type of methods are sometimes not practical due to the expensive computations.

There also have been a few attempts of leveraging back-translation [100] or exploiting machine-generated rules [101] to generate adversarial examples for language tasks. In [98], Gao et al. score the word importance by removing it from text and computing the influence to the classification results; then perturbs words at a character level in the descending order regarding to word importance scores. In [102], Li et al. also compute the word importance for greedy token selection, but proceeds by substituting the selected words with the optimal bug from candidates, including similar words in embedding space and word transformations. To avoid the limitations of gradient-based attack methods, some genetic algorithms are designed to perform black-box adversarial attacks [71,95,103]. For instance, Alzantot et al. [71] uses population-based optimization algorithm to generate adversarial examples with semantically similar candidates, where population sampling is performed in a random way at each generation.

### 2.2.3 Graph Adversarial Attacks

Embedding (or dimentional reduction) has been studied extensively for decades as a fundamental effort for networked data. The early study pays more attention to the graph embedding for better feature extraction. In the era of social networks, graphs have become a powerful tool to learn knowledge from network data, where nodes denotes instances that are often characterized by rich attributes, and edges encode relationships between nodes. In particular, deep learning philosophy has lately been successfully integrated with graph embedding, such as graph neural networks (GNNs).

Consequently, GNNs have been attracting increasing attention due to their great success in graph representation learning where models can embed graph data into low-dimensional space that preserves the graph structure and other inherent information. Specifically, GNNs provide more and more powerful techniques for graph understanding and mining [73, 104–106]. In general, GNN models take the connectivity structure of the graphs as the filter to perform neighborhood information aggregation so as to extract high-level features from the nodes and their neighborhoods [107], which have boosted the state-of-the-arts for a variety of downstream tasks (e.g., node classification, link prediction and network embedding) over graphs.

The fast development of the graph embedding methods and their high capability of information representation leaves the window for inference attackers to disclose sensitive information of the networks. To date, there has only been a few attempts to prevent inference attacks through directly sanitizing the graph data. Cai et al. [108] first leverage the mixture of non-sensitive public attribute and link relationship to conduct an inference attack to infer the private attribute. Then, they design a privacy-preserving method which removes or perturbs the accessible user attribute and links that play an important role in inferring private attribute. In [41], Jia et al. use adversarial attacks to generate a set of noise and randomly perturb the graph with selected noise to fool the inference attacker. [109] designs a data-sanitization strategy to obfuscate the attribute information in graph and discuss the trade off between privacy and utility.

Due to the intractability of link manipulations, it is not practical to directly modify either graph structures or node features of graphs. Recent studies [110–114] have shown that GNNs remain vulnerable to adversarial attacks that can easily fool the models into misclassification by performing small perturbations to graph structures and/or node features. Adversarial attacks targeted to social graphs from their network typologies and node features have been increasingly launched. We can briefly divide these method in this line into two groups, namely, (1) graph poisoning which focuses on modifying

the original graph [115]; (2) graph evasion that conducts attacks during the testing test [114]. However, the study in this field has not been well investigated and there exit rare work in taking advantage of the property of adversarial graph for social network protections. In [116], Kumar et al. carry this idea in a different direction and leverage the vulnerabilities of graph towards adversarial examples to protect kinship privacy.

# Chapter 3
# The Good: Exploring the Applicability of Adversarial Attacks for Social Good

In this chapter, we present two work of our studies from the good aspect of adversarial machine learning. That is, we search for a variety of potential benefits of adversarial machine learning in different application scenarios. For instance, the intriguing learning capabilities of diverse deep learning systems provide powerful tools for software security area to automate the non-trivial processes where could require too much human labor and expert knowledge. Also, the fast and effective ways while generating new adversarial examples could be used to solve the data scarcity and time-consuming learning problems. In our work, we leverage the vulnerabilities of deep learning systems and explore the applicability of adversarial machine learning in settling different kinds of data privacy issues in social media.

## 3.1 Adversary for Social Good: Leveraging Adversarial Attacks to Protect Personal Attribute Privacy

Social media has drastically reshaped the world that allows billions of people to engage in such interactive environments to conveniently create and share content with the public. Among them, text data (e.g., tweets, blogs) maintains the basic yet important social activities and generates a rich source of user-oriented information. While those explicit sensitive user data like credentials has been significantly protected by all means, personal private attribute (e.g., age, gender, location) disclosure due to inference attacks

is somehow challenging to avoid, especially when powerful natural language processing (NLP) techniques have been effectively deployed to automate attribute inferences from implicit text data. This puts users' attribute privacy at risk. To address this challenge, in this paper, we leverage the inherent vulnerability of machine learning to adversarial attacks, and design a novel text-space **Adv**ersarial attack for **S**ocial **G**ood, called *Adv4SG*. In other words, we cast the problem of protecting personal attribute privacy as an adversarial attack formulation problem over the social media text data to defend against NLP-based attribute inference attacks. More specifically, Adv4SG proceeds with a sequence of word perturbations under given constraints such that the probed attribute cannot be identified correctly. Different from the prior works, we advance Adv4SG by considering social media property, and introducing cost-effective mechanisms to expedite attribute obfuscation over text data. Extensive experiments on real-world social media datasets have demonstrated that our method can substantially mitigate the impacts of attribute inference attacks with less computational cost.

### 3.1.1 Introduction

Social media has been enjoying explosive growth for a decade, while its worldwide accessibility has drastically reshaped the world that allows billions of people all around the globe to conveniently perform numerous activities such as creating online profiles, sharing personal posts, and interacting with other people. Such a heterogeneous environment generates a rich source of user-oriented data, which enables researchers to study and understand social communities and individual behaviors. For example, during the COVID-19 pandemic, a surge of solutions have been presented to leverage social media data for risk assessment [117]. However, these apparent benefits also attract attackers to retrieve users' sensitive information and fulfill their malicious intents (e.g., unwanted advertising, user tracing) [42, 43] as illustrated in Figure 3.1. Take Facebook data privacy scandal [118] as an example, the Cambridge Analytica harvested the personal data of millions of people from Facebook without their permission and used it for political advertising purposes. In fact, such privacy risk is not rare on social media, and could be quickly transmitted and propagated [116].

In response to these privacy concerns, social media generally takes action to protect those explicit sensitive user data like credentials by all means. However, with the rapid development in machine learning, and especially the revolutionary learning structures and capabilities raised by deep learning, it is highly probable for the attackers to launch automated attribute inferences from implicit data, which cause unintentional

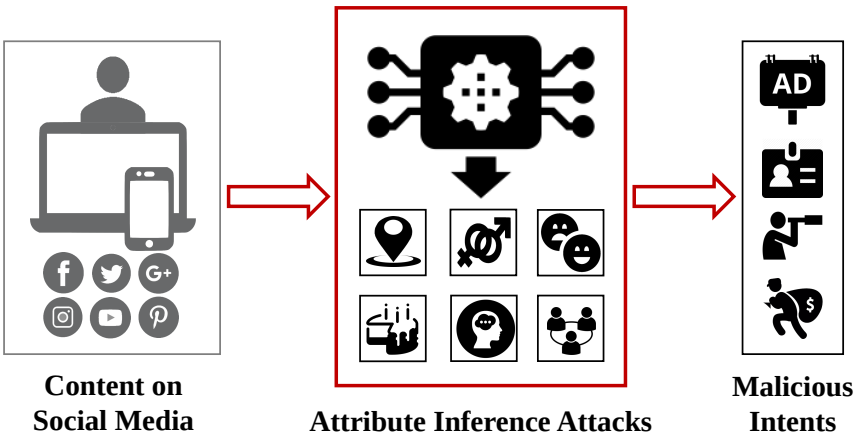



Figure 3.1: Attribute inference attacks over social media.

user attribute information leakage and threaten social media privacy [44–46]. For instance, a user's tweets can be fed to a well-trained machine learning model to infer the user's various private attributes, such as gender, age, and location [41]. Despite their remarkable inference ability, machine learning models are suffering from the inherent learning vulnerability to adversarial attacks [1, 119]. It has shown that by adding small perturbations to the input data, these pre-trained models can be easily fooled into misclassification. To this end, if we take advantage of such a vulnerability, social media privacy protection problem can be reduced to a feasible adversarial attack formulation problem against attribute inference attacks.

Some recent works [41, 47, 64, 65] showed that adversarial attacks have been starting to be leveraged as defenses against inference attacks, which present great potentials to help data obfuscation and privacy protection. However, the prior attempts of this kind focus on the specific application scenarios where their target is limited to continuous data. The investigation into more challenging text data of discrete property has been scarce. In fact, text data is an important component of social media, which shares the most significant privacy of users. On the other hand, natural language processing (NLP)-based models have been widely and effectively used to parse information of text data from different perspectives [67–69]. Therefore, in this work, we would like to focus on text data to investigate how text-space adversarial attacks can be formulated to obfuscate users' attributes and enforce NLP-based inference attacks as less effective as possible for privacy disclosure.

More specifically, we present a text-space *adversarial attack as defense*, or *AaaD* for short, against NLP-based attribute inferences over social media data. AaaD proceeds by

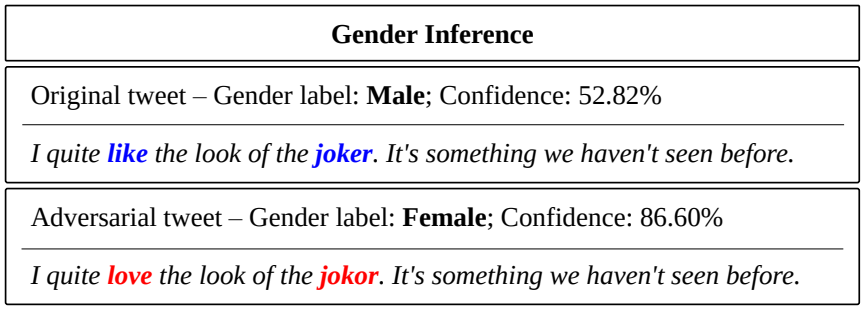



Figure 3.2: Attribute obfuscation by AaaD.

iteratively perturbing the source text originated from social media, such that its specific attribute label is changed, while the underlying constraints conformable to text-space attacks are satisfied. This naturally leads to the following two goals for AaaD: (1) constructing a sequence of constrained perturbations to automatically craft plausible adversarial texts, and (2) making the inference attack model fail to predict correct attribute values from the perturbed input texts. As an example, Figure 3.2 shows two perturbations performed by AaaD on a tweet. The first perturbation changes "*like*" to a semantically similar word "*love*", while the second one replaces "*joker*" with a visually similar word "*jokor*", both of which follow our defined constraints and successfully obfuscate the target attribute. Though there are challenges for attribute annotation on social media data, we believe that our work has implications on the applicability of adversarial attacks for undermining NLP-based inference threats and improving privacy protection in practice.

In summary, this work has the following major contributions:

- A novel and practical paradigm of protecting personal attribute privacy on social media that leverages adversarial learning to mislead attribute inference attacks.
- An adversarial attack is designed to obfuscate users' private attribute on more challenging text data of discrete property. Adv4SG is regulated by a reformed population-based optimization algorithm over perturbation subroutines that conform to text-space attack constraints, which can achieve better success rate in misclassifying attributes with less computational cost.
- The practical black-box setting is considered for Adv4SG's formulation, where the transferability of the proposed method is investigated to validate its applicability in real-world privacy protection scenarios.

- Extensive experimental evaluations on three real-world social media datasets (tweets and blogs) with different attributes to demonstrate the effectiveness of Adv4SG on attribute obfuscation and privacy protection.

The rest of the paper is organized as follows. Section 3.1.2 defines the problem of attack model for attribute inferences and adversarial attack for attribute protection. Section 3.1.3 presents our detailed technical steps of text-space adversarial attack Adv4SG for attribute privacy protection on social media. Section 3.2.4 evaluates the effectiveness of Adv4SG and the impact of different settings. Section 3.1.5 discusses the applicability and limitations of our work.

## 3.1.2 Methods and Technical Solutions

In this section, we first provide the problem definition of the attack model for attribute inferences, and then adversarial attack for attribute protection before technically detailing our proposed model Adv4SG in the following section.

### 3.1.2.1 Attack Model for Attribute Inferences

Social media enables users to post text data for social engagements. This data expose users' information to public where the attackers can take advantage of them to conduct inference attacks for their harmful purposes [64]. Considering that social media generally takes action to protect the explicit and identifiable information, in this work, we assume that the attackers would take advantage of the implicit information from text data to train NLP models so as to achieve their attribute inference goals. Without loss of generality, we denote social media text data $\mathcal{D}$ to be of the form $\mathcal{D} = \{d_i, y_i^t\}_{i=1}^n$ of $n$ texts, where each text $d \in \mathcal{D}$ is associated with a ground-truth label $y^t \in \mathcal{Y}^t$ for an attribute $t \in \mathcal{T}$; $\mathcal{Y}^t$ is the label set of the attribute $t$ and $\mathcal{T}$ is the attribute set. For instance, $\mathcal{T}$ can be specified as $\mathcal{T} = \{\text{gender}, \text{age}, \text{location}, \cdots\}$, Taking location attribute (main four U.S. regions) as an example: $\mathcal{Y}$ can be accordingly specified as $\mathcal{Y} = \{0{:}\text{Northeast}, 1{:}\text{Midwest}, 2{:}\text{South}, 3{:}\text{West}\}$. We follow the general NLP routine to deal with discrete text data by mapping each text $d$ into a $k$-dimensional feature vector $\mathbf{x} = \phi(d)$ where $\phi$ is a feature representation function $\phi : \mathcal{D} \to \mathbf{X} \subseteq \mathbb{R}^{n \times k}$ in which $n \times k$ is the dimension of the embedding space. In this respect, we can derive the predicted label of text $\mathbf{x}$ using the following formula

$$y^* = \underset{y \in \mathcal{Y}}{\text{argmax}}\ l_y(\mathbf{x}) \tag{3.1}$$

where $l_y(\mathbf{x})$ is the confidence score of predicting sample text $\mathbf{x}$ as attribute label $y$ using an NLP model $l$ (e.g., convolutional neural network (CNN), long short-term memory (LSTM), and Transformer). From Eq. (3.1), we can see that the final attribute label assigned to the input sample is the one with the highest confidence score.

#### 3.1.2.2 Adversarial Attack for Attribute Protection

In the text-space, we aim to design an adversarial attack on textual data to defend against attribute inference attackers. In practical, our designed defender can be a software on the client side for social media users. In our setting, we assume that the defender can perturb user's public textual data such as tweets or blogs once the user gives that permission to the defender. In this regard, given an attribute to protect, a text-space adversarial attack attempts to perturb the texts to obfuscate that attribute and prevent inference attack models from correctly identifying their private attribute values. That is, the defender modifies an original text $\mathbf{x}$ with assigned attribute label $y^t$ to a text $\widehat{\mathbf{x}}$ that is classified to any other label $\widehat{y^t} \in \mathcal{Y}^t$ ($\widehat{y^t} \neq y^t$) through adding a small perturbation $\delta$. Therefore, we define our objective function as follows.

$$f(\mathbf{x}+\delta) = l_{y^t}(\mathbf{x}+\delta) - \max_{i \neq y^t}\{l_i(\mathbf{x}+\delta)\} \tag{3.2}$$

where $\mathbf{x}$ is classified as a member of $\widehat{y^t}$ if and only if $f(\mathbf{x}+\delta) < 0$ [120]. $\delta$ represents the distance between original text and adversarial text, which is required to be imperceptible in the studies of continuous data to evade human detection. In our setting, it shares the similar constraint as to guarantee the text quality. The majority of adversarial attack methods [1, 26, 27, 121] intuitively perform a gradient-based adversarial attack in the general feature space by solving the following optimization problem:

$$\begin{aligned} \delta^* &= \arg\min_{\delta \in \mathbb{R}^k} f(\mathbf{x}+\delta) \\ &\text{s.t.} \quad \|\delta\|_p < \epsilon \quad \text{and} \quad f(\mathbf{x}+\delta) < 0 \end{aligned} \tag{3.3}$$

However, these gradient-driven adversarial attack methods from image domain cannot be directly applied to text space. Compared to image classification, defining adversarial textual input is more challenging as there is no simple notion of metric between utterances measuring perturbations. $L_p$-norm distance metric typically works on continuous feature space, but is not capable of bounding the expected perturbation on texts represented as discrete tokens. Besides, gradients computed from the feature space are hard to define

in text space due to its discrete property. In addition, a valid and realistic text-space adversarial attack for social good has to comply with some essential underlying constraints on the modification of the texts. These challenges need to be addressed in Adv4SG.

### 3.1.3 Adversary for Attribute Privacy Protection

In this section, we first identify the black-box setting and underlying constraints conformable to text-space attacks; guided by our formulation, we detail our adversary idea of how we formulate an adversarial attack Adv4SG to protect attribute privacy against NLP-based inferences over social media text data. The overview of our proposed method Adv4SG is illustrated in Figure 3.10.

#### 3.1.3.1 Black-box Attack

Considering the challenge that we are unable to access attacker's inference models, we put our work under the black-box setting, where the devised adversarial attack is not aware of any information about the inference model, including model choice, architecture, parameters, and training data. Compared to the assumptions made in [71, 102, 121, 122] that the attacks are able to retrieve the prediction scores by querying the target model with inputs, our black-box setting is more practical. In the real-world social media scenario, inference attackers have a variety of model choices, and it is impossible to specify one out of many. To this end, we self-learn a surrogate NLP model $l$ to perform attribute inference and craft adversarial texts. Similar to the attackers, we can train such an inference model using the public data and attribute values from the users. Due to transferability in adversarial machine learning [72], the adversarial texts optimized to mislead the surrogate model are very likely to evade the real attackers' inference models.

#### 3.1.3.2 Text-space Attack Constraints

Not like image perturbations, small modifications on text can be visually noticeable to human viewers and even incur severe semantic loss on humans understanding. Also, properties of text data, e.g., grammars, writings, are adjustable and flexible in specific scenarios. For instance, we can simply copy the words from another text with different attribute labels for impersonation, or heavily obfuscating the source text for evasion. These adversarial attacks, however, suffer from semantic loss, generate implausible text, and have a noticeable effect on a human viewer. Also, perturbations over feature space may not be able to be mapped to admissible token values in text space, so the generation

of text-space adversarial attacks for social good should comply with some essential constraints to guarantee their validity and applicability. As such, we define a set of constraints to guide our text-space adversarial attack and clarify its strengths.

- **End-to-end learnability.** In order to generate a practical text-space adversarial text, the first and basic requirement to be achieved is the end-to-end learnability, which enforces iterative perturbations to be performed from text space to text space. In other words, the text-space adversarial attacks need to follow the transformation flow $\mathcal{D} \rightarrow \mathcal{D}$, where $d \mapsto \widehat{d}$ takes an original text $d$ and generates an adversarial version $\widehat{d}$. Since the feature representation function $\phi$ is generally not invertible, the challenge becomes to find a way to apply transformations $\delta$ on $d$ to generate $\widehat{d}$, so that $\phi(\widehat{d})$ is as close to $\widehat{\mathbf{x}}$ as possible [123]. This suggests that the word perturbations on text $d$ should not be arbitrary, but guided by the misclassification of the target attribute.

- **Visual similarity.** Modifications on texts are hard to be imperceptible to human eyes. However, in order to increase the text validity and reduce the utility loss to facilitate its applicability in the social media environment, the generated adversarial texts should be perceptibly similar to the original ones as much as possible. This requirement can be satisfied by either perturbing the texts using the visually similar words, or restricting the number of words that are allowed to be modified.

- **Text plausibility.** When we modify the text, usually we can substitute the original tokens with any other legal tokens in word corpus because they are syntactically correct and readable to human. We consider this validity requirement as text plausibility. For our problem, text plausibility is important as the adversarial text would not only fool attribute inference attack models, but might also be posted in social media for displaying. For this reason, artifacts, which easily reveal that an adversarial text is invalid (e.g., garbled text, words with symbols), will not included. However, due to the fast-sharing and informal-writing property of textual posts in social media, it may tolerate words with small misspellings or distortions, which are still plausible to humans. In this respect, we design more diverse perturbation rules to construct adversarial texts.

- **Semantic preservability.** Preserving semantics is also one of our goals when generating high-quality adversarial texts in the context of social media. To achieve that, we use distance metrics from different perspectives to guarantee the small

distance in the feature space that preserves semantics for texts. On the text level, the edit distance (e.g., the number of perturbed words) between the original text $d$ and the adversarial text $\widehat{d}$ we restrict for virtual similarity can also help seek the semantic equivalence. Moreover, we add constraint on the text distance for the word level. For instance, we limit the Euclidean distance between the original and perturbed word vectors to ensure that each word transformation is as semantic-preserving as possible.

- **Attack automaticity.** To be applied in practical use, the perturbations performed during the adversarial attack procedure need to be completely automated without human intervention. This requires that the possible and available changes made to the text $d$ exclude any transformations that are hand-crafted or need re-engineering on different datasets. In this way, the adversarial attack can be feasible to protect different attributes on different data scenarios without extra update efforts to the overall framework.

#### 3.1.3.3 Overview of Adv4SG

The aforementioned real-world limitation and main types of constraints on text-space adversarial attacks raise significant challenges to the design of our attack method Adv4SG. To address these challenges, we propose Adv4SG to directly perturb the tokens in the text with guidance towards the misclassification of the target attribute through a self-trained NLP model, where the end-to-end learnability constraint and the black-box setting are naturally satisfied. Generally, tokens can be represented in the forms of words and characters, but in our attack formulation, we focus on perturbing the texts at word-level for two reasons: (1) the implicit information of the texts can be better encoded from the latent representations using word embedding than characters, which meets the assumption that the attackers would utilize the implicit information to train NLP-based models for attribute inferences; (2) the search space of possible changes over words is much smaller than characters, such that word-level perturbation is significantly more computationally tractable than character-level perturbation. Accordingly, we use edit distance metric in terms of the number of word changes to control the size of modifications so as to ensure the ability of fooling the threat model while remaining imperceptible. The overview of Adv4SG is illustrated in Fig. 3.10.

To this end, the feature-space adversarial attacks defined in Eq. (3.3) can be updated

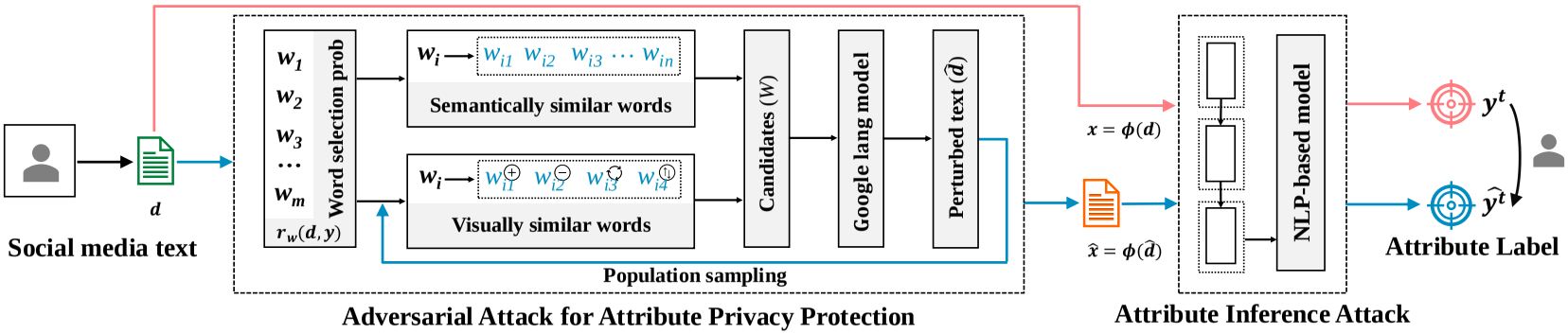



Table 3.1: The overview of our proposed text-space adversarial attack Adv4SG for protecting personal attribute privacy.

to a text-space optimization problem as follows:

$$\begin{aligned} \delta^* &= \arg\min_{\delta \in \mathcal{W}} f(\phi(d+\delta)) \\ \text{s.t.} \quad & \widehat{d} = d + \delta, \quad s(\widehat{d}, d) < \epsilon \quad \text{and} \quad f(\phi(\widehat{d})) < 0 \end{aligned} \tag{3.4}$$

where + implies the high-level word change, $s(\widehat{d}, d)$ denotes the number of different words between $\widehat{d}$ and $d$, and $\mathcal{W}$ is the set of plausible and semantic-preserving word candidates for perturbation. Based on Eq. (3.4), Adv4SG proceeds with a sequence of word perturbations, where each perturbation takes the current text $d$, replaces a chosen word with the optimized candidate, and generates a new version $\widehat{d}$ such that $d$ and $\widehat{d}$ are semantically equivalent, until the attribute label is changed or the maximum allowed perturbation $\epsilon$ is reached. Note that, since all the operations and optimizations do not require manual intervention, and candidate constructions and word perturbations are defined and performed on the fly, we can accordingly ensure the automaticity for our attack.

#### 3.1.3.4 Perturbation and Optimization

For a text-space adversarial attack, it is significant to elaborate word perturbations and devise an effective optimization algorithm to guide the transformations towards the specified target [122]. Some existing works [71, 98, 102] have thus delivered promising results in adversarial text generation. Even so, there are still some downsides in these attack methods: (1) word perturbations are limited to either semantically similar candidate replacements or character transformations while ignoring each other; and (2) it is computationally expensive to find an optimal solution using greedy search or genetic algorithm with random population sampling. Differently, we advance Adv4SG by considering social media property, and introducing both semantically and visually similar word candidates for perturbations and an upgraded population-based optimization to force attribute inference models to misbehave faster. We present the technical details of our proposed model Adv4SG in the following separate subsections.

**Scoring Word Importance.** The original genetic attack proposed by Alzantot et al. [71] repeatedly performed perturbation on randomly selected word to formulate the population members at each generation, which may suffer from the vast search space of possible words and easily include those insignificant words. As such, we would like to first score the importance of words in the text to guide the population sampling that touches the important words and thus expedite the adversarial text generation.

Table 3.2: Nearest neighbors for target words using different embeddings: antonym and synonym example pairs are highlighted as red and blue respectively

| **Embedding** | high | red | similar |
|---|---|---|---|
| GloVe | low | blue | same |
| | higher | yellow | different |
| | highest | purple | particular |
| Counter-fitting | highest | rojo | equivalent |
| | supreme | flushed | same |
| | higher | cardinal | like |

Under our black-box setting, self-training NLP model allows us to compute the partial derivative of the confidence score regarding the predicted attribute label at each input word to approximate the word importance. Specifically, we assume the input text $d = (w_1, w_2, \cdots, w_m)$, and the scoring function that determines the importance of $i$-th word in $d$ can be denoted as:

$$r_{w_i}(d, y^t) = \frac{\partial l_{y^t}(\phi(d))}{\partial w_i} \tag{3.5}$$

where $l_{y^t}(\cdot)$ is the confidence score of attribute label $y^t$. Eq. (3.5) implies that the more important word has more impact on the model output, which is more likely to be modified to mislead inference model. Considering the fact that there exist some stop words (e.g., to, the, a, and it) or irrelevant words in a text that make little sense to tamper with, we further use softmax function to normalize the importance scores to serve as word selection probabilities for population sampling. In this regard, we give priority to modifying the more important words in the sentences.

**Constructing Word Candidates.** We focus on perturbing the texts at word-level; that is, we need to construct a set of word candidates for each selected word to perturb or replace. In order to satisfy the constraints that the generated adversarial text retains semantic equivalence and syntactic coherence with the original one and visually imperceptible to human viewers on social media, we design two different types of word candidates for perturbation: semantically similar candidates and visually similar candidates.

- **Semantically similar candidates.** We obtain a set of words by searching the nearest neighbors of the ready-to-perturb word according to the distance in word embedding space. Here we define a threshold $\eta$ to filter out candidates with distance

greater than $\eta$ such that the semantic preservability requirement could be less violated. GloVe is a context-aware word embedding space [124], but it tends to coalesce the notions of semantic similarity and conceptual association and thus fails to distinguish synonyms from antonyms [125]. Examples of such anomalies can be seen in Table 3.2, where words such as "high" and "low", and "similar" and "different" are deemed similar in GloVe embedding space; replacing such words with each other would completely change the semantics of the text. By contrast, counter-fitting embedding provided by Mrkšić et al. [125] leverages synonym and antonym relations to fine-tune GloVe vectors (shown in Table 3.2), which is a better choice for our problem. Therefore, we use counter-fitting embedding to search for the nearest neighbors for the given word.

- **Visually similar candidates.** Apart from legitimate candidates from vocabulary, we also expand the candidate pool with slightly perturbed words. The reasons behind this are that (1) social media, as a fast-sharing and informal-writing environment, is highly misspelling-tolerant, where satiric or deliberate misspellings are not uncommon; (2) words with small character changes are imperceptibly to human eyes and have no significant impact on semantics [126], and (3) would also very likely enforce the selected word to be out of dictionary with "unknown" embedding such that the output may change [98, 102]. To guarantee the text plausibility, we restrict that only small changes can be performed on the original word to create visually similar candidates, and those modified words will not be selected for a second perturbation. We present different word transformation methods as follows[1]:

  1. Add a space or a random character into the word.
  2. Remove a random character from the word.
  3. Swap any two adjacent characters.
  4. Substitute a character in the word with a randomly selected character.
  5. Substitute a character or a substring to a visually (or aurally) similar number, such as $l \mapsto 1$, $o \mapsto 0$, $z \mapsto 2$, and straight $\mapsto$ str8. These are some deliberate formulations or slang on social media for user convenience or a rhetorical purpose.

[1]Both the first and last positions in the original word will not be modified for better perturbation invisibility.

**Algorithm 1:** Perturbation subroutine.

**Function** `PerturbSub`($d$, $y$, $l$, $p$, $n$)**:**
- $w$ = WordSelect($d$, 1, $p$);
- $candsS$ = SemanticConstructor($w$, $n$);
- $candsV$ = VisualConstructor($w$);
- **for** $c_i \in candsS + candsV$ **do**
  - $d(i) \leftarrow$ replace $w$ with $c_i$ in $d$;
  - $score(i) = l_y(\phi(d(i)))$;
  - **if** $c_i \in candsS$ **then**
    - $pf$, $sf \leftarrow$ a word before/after $c_i$ in $d$;
    - $gscore(i)$ = GoogleLM($pf$, $c_i$, $sf$);
  - **end**
- **end**
- $tscore \leftarrow$ top $n/2$ in $gscore$;
- Remove $score(i)$ $\forall$ $c_i \in candsS$ and $c_i \notin tscore$;
- $c = \arg\max_{c_i} score(i)$;
- **return** $d(c)$;

**end**

**Determining Best Candidate for Replacement.** Based on the constructed word candidates, we can observe that the semantically similar candidates may not be always used in the same contexts. To address this issue, we proceed with filtering out those candidates that do not fit within the context by using Google language model [127] to further ensure the semantic correctness. The rest are then integrated with visually similar ones to form the final candidates. Afterwards, we choose the best candidate among them that will maximize the confidence score of the target attribute $\widehat{y}^t$ ($\widehat{y}^t \neq y^t$) prediction when it replaces the ready-to-perturb word in $d$. Then we perturb the text with the optimal candidate and generate a new text as a population member.

**Population-based Optimization.** Equipped with the above three steps, we can formulate a *perturbation subroutine* that accepts an input text (either perturbed or original), perturbs one selected word, and generates a perturbed-version text towards the misclassification of the target attribute, which is illustrated in Algorithm 1. In this way, we are ready to generate a set of these perturbations for the given text. We aim to minimize the number of word perturbations, which makes the adversarial text less likely to be perceived. Therefore, instead of using greedy search [98, 102], we follow the work by Alzantot et al. [71] and leverage population-based optimization to chain the word perturbations together.

The population-based optimization performs by sampling the population at each

**Algorithm 2:** Adv4SG for attribute privacy protection.

**Input:** $d$: a text sample, $y^t$: label for $t$, $l(\cdot)$: inference model, $\epsilon$: maximum perturbations, $n$: neighbor number.

**Output:** $\widehat{d}$: an adversarial text.

```
selectprob = Normalize(r_w(d, y^t));
ŷ^t ← label other than y^t;
P^0 = {PerturbSub(d, ŷ^t, l, selectprob, n)}_{i=1}^N;
for g = 1 → I do
    for i = 1 → N do
        score(i) = l_{ŷ^t}(φ(P_i^{g-1}));
    end
    p = arg max_i score(i),  d̂ = P_p^{g-1};
    if s(d, d̂) ≥ ε then
        return None;
    end
    if arg max_i l_i(φ(d̂)) == ŷ^t then
        return d̂;
    else
        P^g = {d̂},  sampleprob = Normalize(score);
        for i = 2 → N do
            c = PopSample(P^{g-1}, 2, sampleprob);
            P^g = P^g ∪ PerturbSub(c, ŷ^t, l, selectprob, n);
        end
    end
end
return None;
```

iteration, searching for those population members that achieve better performances, and taking them as "parents" to produce next generation [71]. This procedure can be summarized into three main operators. (1) Mutate($d$): select a word from the given input text $d$ using the normalized word importance score as the probability, and perform a perturbation subroutine on $d$. (2) Sample($\mathcal{P}$): sample a text $d_i$ from the population $\mathcal{P} = \{d_1, d_2, ..., d_N\}$ using the confidence score $l_{\widehat{y}}(d_i)$ as the probability. (3) Crossover($d_1$, $d_2$): construct a child text $c = (w_1, w_2, ..., w_m)$ where $w_i$ is randomly chosen from $\{w_i^{d_1}, w_i^{d_2}\}$. Based on these operators, population-based optimization first generates an initial population $\mathcal{P}^0 = \{\text{Mutate}(d)_1, \text{Mutate}(d)_2, \text{Mutate}(d)_N\}$. At each iteration $t$,

the next generation of population will be generated in the following operation batch:

$$\begin{aligned} \widehat{d}^t &= \underset{d \in \mathcal{P}^{t-1}}{\operatorname{argmax}}\ l_{\widehat{y}}(d), \\ c_i^t &= \text{Crossover}(\text{Sample}(\mathcal{P}^{t-1}), \text{Sample}(\mathcal{P}^{t-1})), \\ \mathcal{P}^t &= \{\widehat{d}^t, \text{Mutate}(c_1^t), ..., \text{Mutate}(c_{N-1}^t)\} \end{aligned} \tag{3.6}$$

The optimization will terminate when an adversarial text is found and returned, or the maximum allowed iteration number is reaches. Algorithm 2 illustrates our proposed text-space adversarial attack Adv4SG. Different from the prior work, we improve the success rate of population samplings by choosing those ready-to-perturb words of high importance scores, while visually similar candidates introduced further expedite the adversarial text generation. Through Adv4SG, we can turn adversarial attacks into protection for personal attribute privacy on social media against the attribute inference attacks.

## 3.1.4 Experimental Results and Analysis

### 3.1.4.1 Experimental Setup

**Datasets.** We test our method on three real-world social media datasets: GeoText [128], user gender tweets[2], and blog authorship corpus [129], which are good representatives for social media data as tweets and blogs are posted by different users, and easily accessed by attackers to uncover their private attributes. Specifically, GeoText is a tweet set from $9,500$ users with geographical coordinates in United States. We map each user into one of the main four U.S. regions defined by the Census Bureau[3] and collect $9,281$ valid tweets with four locations (west, midwest, northeast and south). User gender tweets are collected from Kaggle. We filter out those with gender confidence score less than 0.5, and obtain $13,926$ tweets with two genders (female and male). For blog data, it consists of $19,320$ documents, each of which contains the posts provided by a single user. We extract $25,176$ blogs with two attributes: (1) gender (female and male), and (2) age (teenagers (age between 13-18) and adults (age between 23-45)). Note that, age-groups 19-22 are missing in the original data. The data statistics are summarized in Table 3.7.

**Text-space adversarial attack baselines.** We compare Adv4SG with four other state-of-the-art text-space adversarial attack methods, which can be specified as follows:

[2]https://www.kaggle.com/crowdflower/twitter-user-gender-classification

[3]https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf

Table 3.3: Comparing statistics of the three datasets

| Dataset | Attribute | #Posts | #Classes | #Vocabulary |
|---|---|---|---|---|
| Twitter_g | Gender | 13,926 | 2 | 17k |
| Twitter_l | Location | 9,281 | 4 | 16k |
| Blog | Gender, Age | 25,176 | 2 | 22k |

- Genetic attack [71]: this attack uses population-based optimization algorithm to generate adversarial examples with semantically similar candidates, where population sampling is performed in a random way at each generation.
- Greedy attack: this method greedily performs perturbation subroutine of our method on one word at each iteration. We aim to evaluate the performance of perturbation crafted by our subroutine and validate the effect of population-based optimization.
- WordBug [98]: this attack scores word importance by removing it from text, and perturbs words in the descending order regarding word importance scores using character transformations.
- TextBugger [102]: this method also scores the word importance for greedy token selection, but proceeds by substituting the selected words with the optimal bug from candidates, including similar words in embedding space and word transformations.

**Parameter setting.** We use euclidean distance as distance metric to construct semantic-similar candidates from embedding space, and the distance threshold is set to $\eta = 0.5$ to filter out those less similar ones. The size of candidate pool for each word is set as 8, where we choose the best one for replacement. We also limit the maximum allowed word perturbations to 25% of the text length, and we further evaluate its impact on attack performance in Section 3.1.4.2. We randomly select 80% of the samples for training, while the remaining 20% is used for testing, and we report the mean inference accuracy and attack success rate of four attribute inference settings runs on test samples for the evaluation results.

**Attack model for attribute inference attacks.** An attribute inference attack aims to disclose private attributes of users by learning a model on the public data. Since we do not know the attacker's model, we self-train bidirectional LSTM (BiLSTM) [130], multi-layer GRU (M-GRU) [131], ConvNets [132], and CNN-LSTM (C-LSTM) [133] to perform the tasks. We mainly use BiLSTM to evaluate the effectiveness of Adv4SG, while

the comparisons among these four models are leveraged for transferability evaluation in Section 3.2.4.4. All models read in 250 words, where the dimension of each LSTM or GRU hidden unit is 128. We use GloVe [124] to map each word into a 300-dimensional embedding space. Note that, an inference attacker would deploy more robust models to evade adversarial attacks. As adversarial training is considered as one of the most empirically robust methods against adversarial attacks [65, 134], we build up a robust model using adversarial training and further discuss the effectiveness of Adv4SG under this setting in Section 3.1.4.5.

#### 3.1.4.2 Evaluation of Adv4SG

In this section, we validate the effectiveness of Adv4SG against attribute inference attacks and the impacts of different parameters. To evaluate our method, we perturb the correctly classified text examples from the test data of four attribute settings.

**Effectiveness.** In our experiments, we evaluate Adv4SG under different population sizes and iterations as they play a crucial role to determine the degree of sample perturbation and computational cost. In particular, we test the results of our generated adversarial texts with population size $N \in \{10, 20, 30, 40, 50\}$ respectively against different inference attacks, while the maximum iteration $I$ is ranging in $\{10, 20, 30\}$ correspondingly. The experimental results are shown in Figure 3.4. As we can see from the results, the inference accuracy for Twitter-location, Twitter-gender, blog-age, and blog-gender on clean data is 47.76%, 62.25%, 72.92%, and 69.20%, which are relatively close to the state-of-the-art results on each dataset. Adv4SG drastically decreases all these accuracies and achieves the goal of obfuscating attributes and protecting social media text data privacy. Averagely, our method reduces the accuracy of Twitter-location and Twitter-gender inference attacks from 47.76% to 2.19% and from 62.25% to 18.42% respectively; for the larger and longer blog data, we degrade inference accuracy of gender and age from 69.20% to 9.66% and from 72.92% to 13.65% respectively. We present some of our generated adversarial texts in Figure 3.3. It is clear that Adv4SG can subtly perturb important words towards the misclassification target in a plausible and semantic-preserving manner (e.g., "queso" $\mapsto$ "cheese", "awesome" $\mapsto$ "amazing" and "should" $\mapsto$ "shou1d").

**Impact of population size and iteration.** Generally, when we enlarge the population size, the success rate of generating adversarial samples increases and the accuracy of the inference models thus decreases, while the required perturbation number tends to go up as well. However, due to the perturbation limit for each text, the actual attack performance might not always improve for larger population size. We can observe that

**Task:** Twitter-location. **Original label:** South (confidence=76.88%). **New label:** Northeast (confidence=61.66%)

They use the white ~~queso~~ cheese dip from farm fresh. I have seen cases of it in the kitchen.

**Task:** Twitter-gender. **Original label:** Male (confidence=53.46%). **New label:** Female (confidence=84.36%)

That ~~awesome~~ amazing moment when you ~~check~~ chechk your bank account and your parents send you more than you thought.

**Task:** Blog-age. **Original label:** Adults (confidence=76.08%). **New label:** Teens (confidence=60.31%)

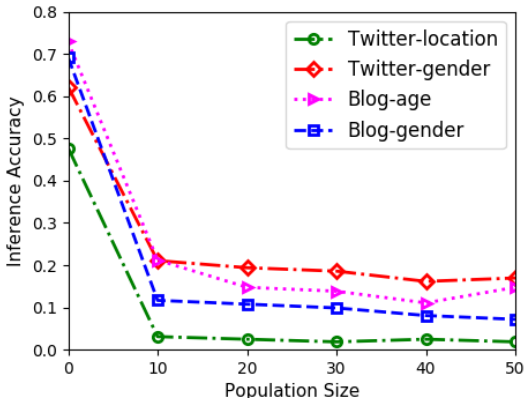
Hellooooooo! Well, in case you haven't guessed by the ~~lack~~ l@ck of my blogs, I have been on holiday ~~nowhere~~ nowhare nice just sitting at home. But I thought I ~~should~~ shou1d take a break from computers as well. I have lots of catching up to do, good news, bad news and lots of ~~events~~ things to tell you all about. So stay tuned for the updates!!

**Task:** Blog-gender. **Original label:** Female (confidence=78.29%). **New label:** Male (confidence=54.43%)

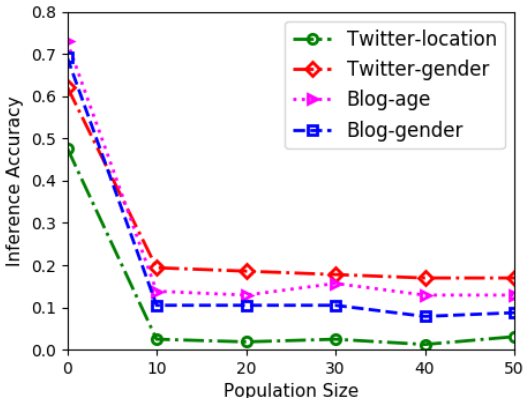
So it starts a ~~blog~~ bl0g on the internet ready for writing. I'm gonna ~~use~~ utilize this a lot over the ~~next~~ future two weeks to let you know what my theatre class is doing, the cute guys I'm meeting and all the rest enjoy.

Figure 3.3: Adversarial texts generated by Adv4SG under different inference tasks and their original texts.

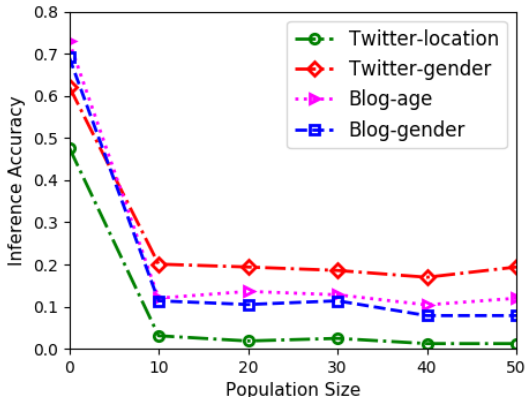
the inference accuracy for all settings drops to the worst at $N = 40$ and then either slightly increases or stays flat when $N$ changes from 40 to 50. On the other hand, the larger iteration provides more improvement space for Adv4SG when the population size is small. For example, when $N = 10$, Adv4SG degrades the inference accuracy for blog-age setting from 21.23% ($I = 10$) to 12.02% ($I = 30$). Nevertheless, such inference accuracy difference among different iteration settings tends to be more statistically insignificant as the population size increases. As shown in Figure 3.4, Adv4SG achieves the comparable performance under all four inference settings at $N = 40$ with $I$ varying in $\{10, 20, 30\}$. The reason behind this is that the larger population size more likely enforces the optimal solutions at earlier iteration, while most of the failed population samples would stay in the loop at later iteration. Considering that the larger iteration may introduce more computational cost, while the larger population size can significantly enhance Adv4SG, we use $N = 40$ and $I = 10$ throughout the following evaluations to keep a good trace-off between the effectiveness and efficiency.




(a) Iteration $I = 10$




(b) Iteration $I = 20$




(c) Iteration $I = 30$

Figure 3.4: Evaluation results: (a), (b) and (c) specify the inference accuracy of Adv4SG with different population sizes and iterations.

**Impact of maximum allowed perturbation ($\epsilon$).** Different choices of $\epsilon$ could affect the performance of Adv4SG, since $\epsilon$ not only limits the number of word perturbations allowed to impact on the attack ability, but also significantly reflects the similarity between the generated adversarial texts and the original texts, and thus has direct impact on the semantic preservability and plausibility of the adversarial texts. We use the cumulative distribution function (CDF) of attack success rate regarding the number of $\epsilon$ to illustrate the evaluation results. From the results shown in Figure 3.6, we can observe that as $\epsilon$ increases, the attack success rate increases as well because of the larger modification space, but the mean sentence semantics quality would decease. Actually, using Adv4SG, most of the generated adversarial texts manage to evade the inference models after perturbing very few words in the texts. More specifically, for Twitter-location inference, about

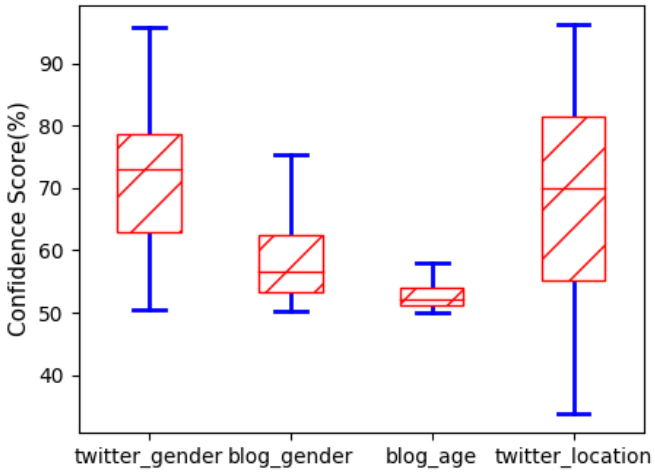



Figure 3.5: the confidence score distribution of the perturbed texts under four inference settings.

57% of the testing texts evade the inference model by perturbing only one word, while this success rate increases to 88% when $\epsilon \leq 3$. For Twitter-gender inference, Adv4SG successfully crafts 57% and 76% of the adversarial texts from the original with at most one word and three word perturbations respectively. For blog-gender inference, the attack success rates are 38% with $\epsilon \leq 1$ and 63% with $\epsilon \leq 3$. For blog-age inference, these two rates are 9% and 30%, which apparently underperforms other settings because of the longer text length. When Adv4SG is allowed to perturb at most 5 words, the attack success rate immediately rises to over 50%. All these results imply that (1) Adv4SG enables most of adversarial texts to be similar to the original texts; (2) the number of perturbations relatively relies on the length of the texts: the average lengths of the texts used for Twitter-location, Twitter-gender, blog-gender, and blog-age are 31, 15, 51 and 61, while the average perturbations are 1.8, 1.4, 2.9, and 5.2 for the corresponding inference tasks.

**Other observations.** In addition, we can also find some more interesting observations from the evaluation results in Figure 3.4 and Figure 3.6: (1) Adv4SG tends to perform worse on binary attributes (e.g., age and gender) than multi-class attributes (e.g., location). It is not difficult to understand that adversarial attacks on binary attributes can be considered as targeted attacks that might take more effort to perturb the texts and enforce misclassification to a specified target class (inverse to the original), while adversarial attacks on multi-class attributes fall into non-targeted attacks that have to

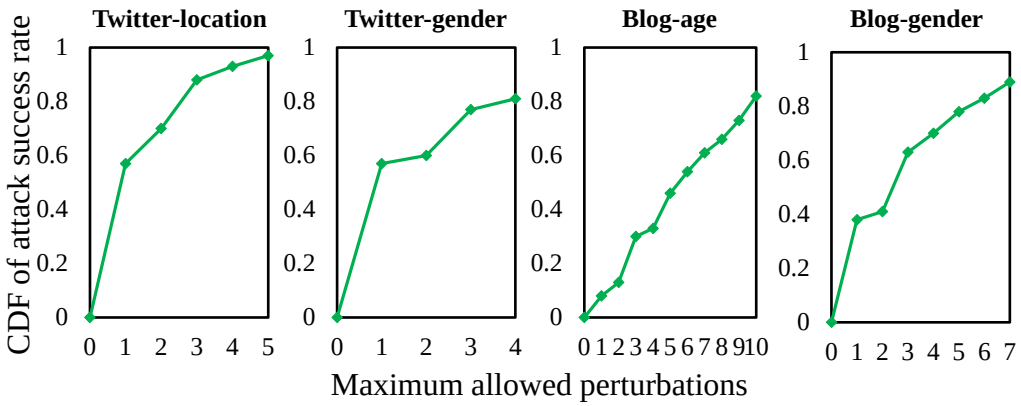



Figure 3.6: Evaluation on maximum allowed perturbation ($\epsilon$) via cumulative distribution of attack success rate.

simply cause the source texts to be misclassified, which is obviously easier. (2) The learning ability of the inference model may also have a potential impact on the Adv4SG's attack effectiveness against it, as small perturbations on the texts more likely lead to evasion for inference models that underperform than others. For example, the inference accuracy for Twitter-location is 47.76%, while Adv4SG successfully reduces it to 2.19% with 7.65% mean perturbation rate. The similar results can be found between blog-age and blog-gender. (3) The age attribute seems more difficult to be obfuscated than others due to relatively higher model inference ability and longer text length, where Adv4SG performs more word perturbations for adversarial text generation.

Furthermore, we show the confidence distributions of those generated adversarial texts that can successfully fool the inference attackers under different deployment settings in Figure 3.5. It indicates the consistent findings with what we observe from other results. For instance, the average confidence values of the perturbed texts for the age attribute are distributed slightly above the borderline (i.e., 50%), which reveals the difficulty in obfuscating age attribute for blog dataset. Differently, the overall scores of other three tasks have been explicitly moved to the misclassification direction, which lead to better attack effectiveness. In addition, the performance of Adv4SG for long texts (i.e., blogs) seems to be more stable than short twitter texts. We guess it correspondingly relates to the different inference capability of the attackers on these datasets.

#### 3.1.4.3 Comparisons with Other Attack Baselines

**Attack performance.** We compare Adv4SG with the other baselines including Genetic attack [71], Greedy attack, WordBug [98], and TextBugger [102]. Specifically, we randomly

Table 3.4: Comparisons of different text-space adversarial methods

| Inference task | Metric | **Adv4SG** | Genetic | Greedy | WordBug | TextBugger |
|---|---|---|---|---|---|---|
| Twitter-location | Success Rate | **97.40%** | 85.71% | 76.62% | 55.84% | 82.91% |
| | Median Ptb Rate | **5.26%** | 6.25% | 8.33% | 10.53% | 7.85% |
| | Mean Ptb Rate | **7.65%** | 9.00% | 10.73% | 18.75% | 11.58% |
| Twitter-gender | Success Rate | **74.03%** | 55.84% | 45.45% | 32.47% | 62.34% |
| | Median Ptb Rate | **9.09%** | 14.29% | 14.64% | 27.27% | 16.67% |
| | Mean Ptb Rate | **12.18%** | 16.28% | 16.73% | 29.56% | 21.37% |
| Blog-age | Success Rate | **82.28%** | 72.15% | 72.15% | 17.72% | 59.49% |
| | Median Ptb Rate | 11.92% | **11.11%** | 12.19% | 31.21% | 19.64% |
| | Mean Ptb Rate | **13.53%** | 13.96% | 14.06% | 27.94% | 23.89% |
| Blog-gender | Success Rate | **88.61%** | 84.81% | 70.89% | 54.43% | 77.22% |
| | Median Ptb Rate | 5.08% | **4.21%** | 7.45% | 17.86% | 12.31% |
| | Mean Ptb Rate | **8.38%** | 8.61% | 10.33% | 19.07% | 16.03% |

sample 50% of correctly classified examples from the testing tweets and blogs to measure the performance of attacks. The comparative results are illustrated in Table 3.4, where Genetic attack achieves better attack success rate and perturbs less words than Greedy attack, WordBug, and TextBugger in most settings, while TextBugger produces the comparable or slightly better performance on tweet attribute obfuscation; Adv4SG outperforms all baselines with marginally higher median perturbation rate than Genetic attack on blog attribute inferences. From the results, we can observe that (1) projecting an important word into "unknown" may enforce inference models to misbehave, while ignoring semantically similar candidates would also miss good evasion chances, and (2) leveraging word importance to facilitate population-based optimization expedites adversarial example generation. When we look into the generated adversarial texts, we find that Greedy attack fails in some of those adversarial texts with more modifications required over long blogs, and hence obtains a smaller perturbation number on average in results. By contrast, Adv4SG either converts those failed texts from Genetic, Greedy, WordBug and TextBugger to adversarial examples, decreases the number of required perturbations, or raise the confidence scores of the perturbed texts, which significantly advances the text-space adversarial attack with respect to effectiveness and efficiency. Thus, Adv4SG can be a feasible paradigm in a real social media environment on attribute obfuscation and privacy effectiveness.

**Computational cost.** From the results of Table 3.4, we can validate that our method

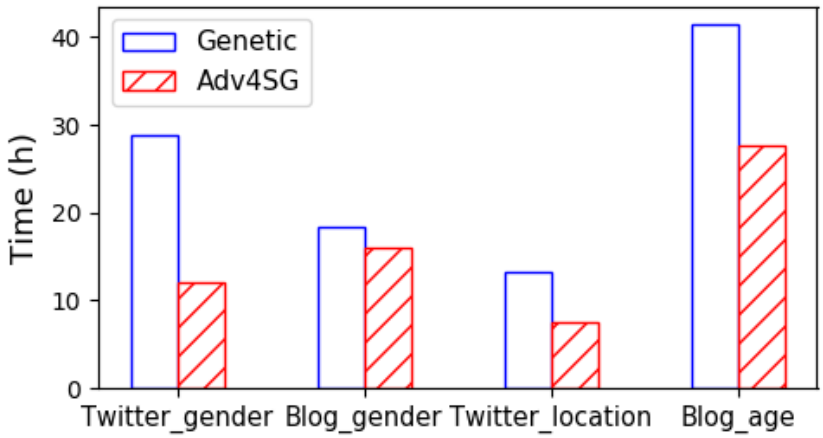



Figure 3.7: Computational cost between Adv4SG and Genetic.

obviously outperforms other state-of-the-arts by obfuscating the private attribute values with smaller word perturbations and higher success rate. Among them, the performance of Genetic method is relatively close to our strategy, both of which enforce attack effectiveness improvement of a large margin against others. Meanwhile, our method and Genetic attack both deploy population-based optimization that performs an evolution process by selecting population candidates to breed the next generation towards better solutions. Thus, here we would like to evaluate the advancement of our method against genetic method from the perspective of computational cost. To be comparable, we use single TITAN Xp for each experiment. We measure the average runtime for different inference settings on Genetic and Adv4SG, respectively. The results are presented in Figure 3.7. We can see from the results that Adv4SG can drastically reduce the computational cost compared to Genetic. For inference tasks such as Twitter-gender and Blog-age, Genetic method costs nearly twice the time of our method. On average, Adv4SG can save 36.85% computational time against the Genetic, which further justifies the advantage of word importance and visually similar candidates we introduce in Adv4SG.

#### 3.1.4.4 Transferability

Under the black-box attack setting, as Adv4SG is implemented through self-trained NLP model, it is necessary to evaluate its transferability to validate if those adversarial texts generated for one model are likely to be misclassified by others. In this evaluation, we deploy Adv4SG to generate adversarial texts on four inference settings for four different NLP models: BiLSTM [130], multi-layer GRU (M-GRU) [131], ConvNets [132], and

CNN-LSTM (C-LSTM) [133]. Then, we evaluate the attack success rate of the generated adversarial texts against other models. To ensure our results are comparable, we build up these models with the same parameter settings (different dropout rates) and training data. Accordingly, we build a cross-model transferability table, where each table unit $(i, j)$ holds the percentage of adversarial texts crafted to mislead model $i$ (row index) that are misclassified by model $j$ (column index).

From Table 3.5, we can see that the cross-model transferability for Adv4SG is a strong but heterogeneous phenomenon: (1) between same model pairs, the percentage numbers are higher than 80%, most of which are close or beyond 90%; (2) between pairs of different models, some enjoy good transferability (e.g., 76.67% for M-GRU and BiLSTM on blog-gender setting), while some only have moderate one (e.g., 31.03% for ConvNets and M-GRU on blog-age setting). The results also imply that the complexity of the surrogate model and the intrinsic adversarial vulnerability of the target model contributes to attack transferability (e.g., all models against ConvNets achieve relatively higher transferability than others). Adversarial texts generated from more complicated surrogate model tends to have better attack success rates on other target models. We believe it is because models with complex structures enjoy high capability of regularization on malicious perturbations wherefore adversaries need to enlarge the input mutations to fool the model. In real-world scenarios, since the target models are uncontrollable and inaccessible, social media may need to elaborate the surrogate model for better transferability when applying Adv4SG for attribute privacy protections.

#### 3.1.4.5 Adversarial Training

Attribute inference attackers may detect adversarial examples or defenses in place and train more robust models to evade such protection and thus enhance the inference accuracy. In this respect, we investigate a more robust target model based on adversarial training, which is considered as one of the most empirically effective ways to improve the model robustness against adversarial attacks [1], to further evaluate the effectiveness of Adv4SG. More specifically, in this part we study if adversarial training can strengthen the inference attack and lower the success rate of our defense method. We use Adv4SG to generate adversarial texts from random 50% of correctly classified training data, and incorporate these crafted adversarial examples into the training process, with which, we retrain the BiLSTM inference model under the same parameter setting described in Section 3.1.4.1. Afterwards, we follow the same paradigm to perform Adv4SG over adversarially trained models to test the success rate under four inference tasks.

Table 3.5: Transferability on four attribute inference settings: each table unit $(i, j)$ specifies the percentage (%) of adversarial texts produced for model $i$ that are misclassified by model $j$ ($i$ is row index, while $j$ is column index)

| **Model** | **Twitter-gender** | | | | **Twitter-location** | | | | **Blog-gender** | | | | **Blog-age** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BiLSTM | M-GRU | ConvNets | C-LSTM | BiLSTM | M-GRU | ConvNets | C-LSTM | BiLSTM | M-GRU | ConvNets | C-LSTM | BiLSTM | M-GRU | ConvNets | C-LSTM |
| BiLSTM | 93.65 | 42.86 | 50.76 | 47.62 | 96.53 | 36.32 | 38.95 | 36.84 | 87.09 | 72.00 | 70.67 | 68.00 | 100.00 | 56.58 | 39.47 | 43.42 |
| M-GRU | 34.62 | 88.46 | 65.38 | 46.51 | 30.77 | 92.31 | 69.23 | 38.46 | 76.67 | 83.33 | 63.33 | 56.67 | 67.57 | 86.49 | 72.97 | 59.46 |
| ConvNets | 51.72 | 55.17 | 89.66 | 58.62 | 39.13 | 37.50 | 85.71 | 43.49 | 61.26 | 53.33 | 90.77 | 59.26 | 48.65 | 31.03 | 81.08 | 62.16 |
| C-LSTM | 36.36 | 33.33 | 42.42 | 90.91 | 38.24 | 35.29 | 47.06 | 88.24 | 67.86 | 60.71 | 57.14 | 89.79 | 60.53 | 32.05 | 39.84 | 85.63 |

Table 3.6: Success rates on models with (Adv_model) and without adversarial training (Ori_model)

| Model | Twitter-location | Twitter-gender | Blog-age | Blog-gender |
|---|---|---|---|---|
| Ori_model | 97.40% | 74.03% | 82.28% | 88.61% |
| Adv_model | 97.40% | 71.82% | 75.95% | 89.87% |

The results are illustrated in Table 3.6. From our results, we can observe that adversarial training barely improves the robustness of inference models against our adversarial attack Adv4SG. The updated success rates of Adv4SG over the inference models after adversarial training are 97.40%, 71.82%, 75.95% and 89.87% on Twitter-location, Twitter-gender, blog-age, and blog-gender, respectively, which yield no significant difference from the success rates over the original models. These results demonstrate the resilience of the perturbations generated by Adv4SG and the difficulty for inference attackers in defending against our adversarial attack. On the other hand, the relatively weak learning ability of the inference model we deploy in our experiments may somewhat contribute to the success of Adv4SG. This inspires our future work in increasing the learning robustness and capability of NLP models and the advance of adversarial attacks against them.

### 3.1.5 Applicability and Limitations

As our work is motivated to protect users' attribute privacy in social media, it is more important for us to discuss how to put it in real use and bring practical impact to our life. For its applicability, Adv4SG should be an easy-to-use service provided on users' social media client side, so that its privacy protection functionality would be realized in practice. For example, Adv4SG can be developed as an API that is integrated into social media posting and editing systems to allow users to choose the adversarial text according to their provided attribute and text content. An conceptual example of such an attribute obfuscation service devised in Facebook is illustrated in Figure 3.8, which can change the private attribute that people are unwilling to disclose (i.e., age) of a post to wrong results. Once users give privileges to this adversarial perturbation, the posting data will be obfuscated and updated on behalf of the users. Although not all users might consistently accept the obfuscation feature, we think the possibility of conveniently and proactively perturbing public data can also promisingly increase the uncertainty and difficulty to the attackers. From this perspective, our designed method Adv4SG can

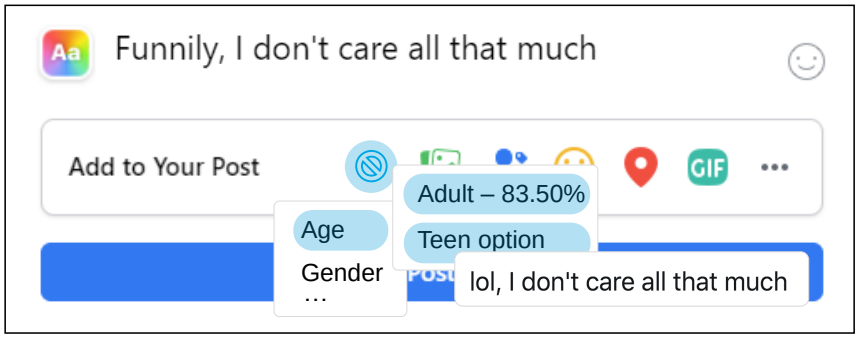



Figure 3.8: An example of attribute obfuscation service.

serve as a valid function to exhaustively obfuscate the social media data before making it publicly available.

Nonetheless, our approach also poses some challenges and limitations which we discuss as follows. (1) We successfully perform Adv4SG over the annotated public data in this work, while the real social media lacks the ground truth, which disables Adv4SG from generating the adversarial texts in a real-time fashion. To better obfuscate the attributes, we may need to first recognize the targets. Though attribute recognition is irrelevant for the scope of our work, it is an interesting future work to leverage attribute recognition for better protection solutions. (2) In our experiments, we simply train some regular attack models for attribute inferences. Though Adv4SG has been validated to be transferable and resilient against adversarial training, the attackers could take advantage of more advanced and robust learning models (e.g., spelling checking, and graph learning) to infer attributes and thus deteriorate Adv4SG. We acknowledge this limitation and leave the investigation on this arms race as our future work, yet it does not impact the great value and general validity of our new insight on the adversarial attacks for attribute obfuscation and privacy protection in practice, as advanced and robust models could always be evaded by more complicated and sophisticated adversarial techniques.

## 3.2 Adversary for Social Good: Leveraging Attribute-Obfuscating Attack to Protect Social Networks' User Privacy

As social networks become indispensable for people's daily lives, inference attacks pose significant threat to users' privacy where attackers can infiltrate users' information and

infer their private attributes (e.g., gender, age, location, career, and political views). In particular, social networks are generally represented as graph-structured data, maintaining rich user activities and complex relationships among them. This enables attackers to deploy state-of-the-art graph neural networks (GNNs) to automate attribute inferences from user features and relationships, which makes such privacy disclosure hard to avoid. To address this challenge, in this work, we leverage the vulnerability of GNNs to adversarial attacks, and propose a new graph poisoning attack, called Attribute-Obfuscating Attack (AttrOBF) to mislead GNNs into misclassification and thus protect personal attribute privacy against GNN-based inference attacks on social networks. Different from the prior attacks using perturbations on the either graph structure or node features, AttrOBF provides a more practical formulation through obfuscating optimal training user attribute values for real-world social graphs, and also advances the attribute-obfuscating attack by solving the problems regarding unavailability of test attribute annotations, black-box setting, bi-level optimization, and non-differentiable obfuscating operation. We demonstrate the effectiveness of our proposed attack method AttrOBF on user private attribute obfuscation by extensive experiments over three real-world social network datasets. We believe our work yields great potential of applying adversarial attacks to attribute protection on social networks.

### 3.2.1 Introduction

Social networks have emerged as an indispensable part of our daily lives through enormous websites and apps, which allow us to conveniently share personal ideas for social engagements. Such an interactive environment generates a large amount of user-oriented data. Due to its accessibility and information richness, this data attracts not only researchers to study and understand social communities and individual behaviors, but also attackers to disclose users' sensitive information to fulfill their malicious intents (e.g., unwanted advertising, user tracing) [42, 43]. This puts users' privacy at risk. In fact, with the rapid development in machine learning, and especially the revolutionary learning structures and capabilities raised by deep learning, such privacy risk is not rare on social networks, and could be quickly transmitted and propagated through attribute inference attacks in an automatic fashion [44–46, 67, 70, 116].

In particular, social networks are naturally represented as graph-structured data, maintaining individual user activities and complex relationships among them. For example, nodes in these social graphs usually encode users' information with respect to their profiles, posts, photos, or other statuses, while graph edges connect users with their

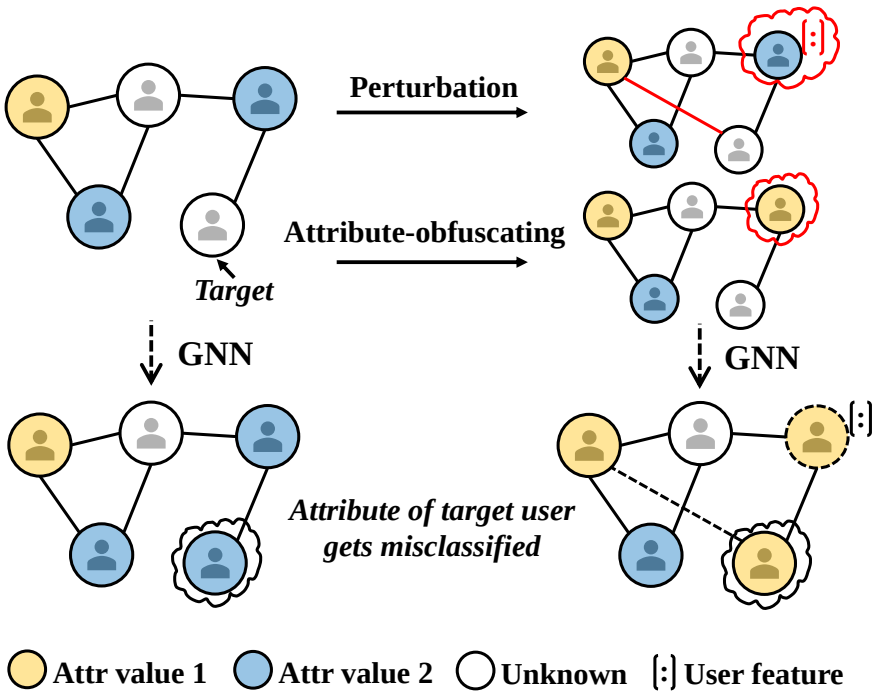



Figure 3.9: GNN-based inference attack example and graph adversarial attack leading to attribute obfuscation (i.e., attribute of target user gets misclassified) through traditional perturbation on graph structure/node feature or our proposed attribute obfuscating operation.

friendships, kinships, or follower-followee relationships. On the other hand, graph neural networks (GNNs) provide more and more powerful techniques for graph understanding and mining [73, 104–106]. These GNNs take the connectivity structure of the graphs as the filter to perform neighborhood information aggregation so as to extract high-level features from the nodes and their neighborhoods [107], which have boosted the state-of-the-arts for a variety of downstream tasks (e.g., node classification and link prediction) over graphs. Therefore, a surge of effective inference attacks utilize GNNs to reveal personal attributes (e.g., age, gender, location, career, and political views) that people are unwilling to disclose on social networks [135–137]. The idea is visualized as an example on the left-hand side of Figure 3.9 illustrating that the attribute of the target user can be correctly identified by leveraging GNNs over graph structure and user features.

In this work, we simply demonstrate an attribute privacy threat on social networks as the scenario that an attacker trains a well-performed GNN model to infer users' private attributes from graph-structured data such as Facebook friendship networks and Twitter follower-followee networks. With this in mind, some previous attempts have paid close

attention to protect these attributes against inference attacks [41,43,47,64,85,116,138,138], which, however, still suffer from either large computational cost and utility loss with graph anonymization [43, 85], or specific application scenarios limited to visual or textual data [41, 47, 64, 138, 138]. Thus, our goal here is to generalize the investigation to more challenging graph-structured data, and protect personal attribute privacy in this regard from a novel and practical adversarial learning perspective.

Despite great success, GNNs are still faced with the inherent learning-security challenge of lacking adversarial robustness existing in regular machine learning models [1, 20, 119]. Recent studies [110–114,114,115] have shown that GNNs remain vulnerable to adversarial attacks that can easily fool the models into misclassification by performing small perturbations to graph structures and/or node features, which is shown in Figure 3.9 (the upper one on the right-hand side). As the effectiveness of attribute inference attacks depends on high learning performance from GNN model while adversarial attacks substantially decrease its performance, this observation accordingly inspires us to take advantage of such a vulnerability and cast personal attribute privacy protection problem on social networks as an adversarial attack formulation problem against GNN-based attribute inference attacks. To achieve this goal, we face two challenges: (1) as inference attackers have a variety of choices in GNN construction, it is impossible for us to access the inference models for crafting graph adversarial attacks; (2) gathering sufficient amount of labeled data from social media can be expensive and time-consuming which adds more challenge for the expansion of graph neural networks towards users' attribute protection; (3) due to multimodality of user representations and intractability of relationship manipulations, modifications on either graph structures or node features cannot guarantee the validity of adversarial social networks, which are impractical in the real-world settings.

To address the above challenges, in this work, we design a black-box adversarial poisoning attack, called attribute-obfuscating attack (AttrOBF), to deteriorate GNNs into misclassification and thus protect personal attribute privacy against GNN-based attribute inferences on social network data. Different from the regular label based attacks strictly limited to binary labels [74, 139], AttrOBF is more general to deal with either binary (e.g., gender attribute) or multi-class (e.g., location attribute) classification tasks. Given a social network, AttrOBF proceeds by modifying a small fraction of optimal training users' attribute values, while the obfuscated attribute information can propagate along the whole graph through layer-wise neighborhood aggregations, such that the overall performance of attribute inferences by a self-trained GNN model is drastically degraded. Figure 3.9 (the lower one on the right-hand side) illustrates the goal of our

work.

Due to transferability in adversarial machine learning [72], the obfuscated attribute over social networks is very likely to mislead the real attackers' inference GNN models. Furthermore, to solve the labeled data shortage, we leverage GNNs such as graph convolutional network (GCN) to conduct semi-supervised learning for node classification task on only a small number of labeled nodes. Then we obtain the prediction results of testing data to serve as the ground truth for our optimizations. More importantly, it is necessary for inference attackers to collect initial attribute annotations for training, while users' annotating on social networks generally relies on their self-reporting; therefore, attribute obfuscating can be conveniently and proactively realized by users and data publishers, and also easily passed to subsequent inference attacks. These advantages allow a refined paradigm to effectively and efficiently mitigate the impacts of GNN-based inference attacks on attribute disclosure and enhance personal privacy protection in practice. In summary, our major contributions of this work are listed as follows:

- We explore a novel and practical perspective of protecting personal attribute privacy on social networks that leverages adversarial attacks to mitigate GNN-based attribute inference attacks.
- We propose a new adversarial attack AttrOBF for users' private attribute protection on social networks. To avoid the NP-hard search, AttrOBF employs gradient-based method to obfuscate optimal training attribute values in a cost-efficient manner, where the problems regarding unavailability of test attribute annotations, black-box setting, bi-level optimization between attacks and GNNs, and non-differential obfuscating operation are specially addressed.
- We conduct extensive experimental evaluations on three real-world social network datasets with different attributes to demonstrate the effectiveness of AttrOBF on attribute obfuscation and privacy protection.

### 3.2.2 Overview

In this section, we first introduce the attack model for attribute inferences, and adversarial attack for attribute protection before diving into the technical details of AttrOBF in the following section.

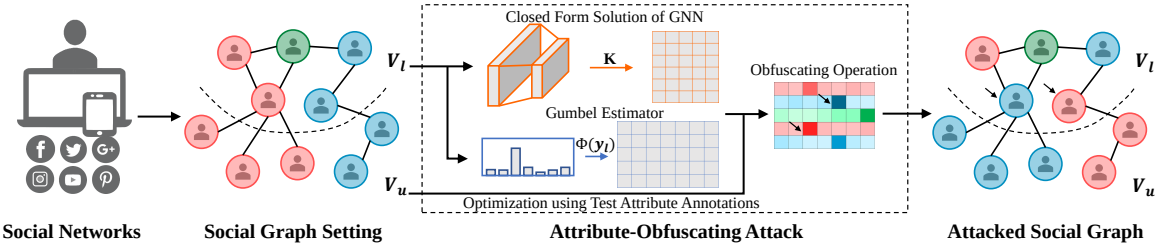



Figure 3.10: The overview of our attribute-obfuscating attack AttrOBF for protecting personal attribute privacy on social networks.

#### 3.2.2.1 Graph Neural Network for Attribute Inference

Social networks may indicate users' sensitive information, and thus easily expose them to the attackers who can access the data and infer the private attributes of interest to deliberately fulfill the economic, social, or political intents [64]. Considering that social networks are generally represented as graph-structured data, in this work, we assume that the attackers would take advantage of user features and relationships to train GNN models so as to achieve their attribute inference goals [135–137].

Without loss of generality, we denote social network data $G$ to be of the form $G = (V, E, \mathbf{X})$, where $V$ ($n = |V|$) is the set of user nodes, $E$ is the set of edges specifying relationships among users, and $\mathbf{X} \in \mathbb{R}^{n \times d}$ is feature matrix. Nodes $V$ can be further divided into annotated node set $V_l$ ($n_l = |V_l|$) and unannotated node set $V_u$ ($n_u = |V_u|$), where each annotated node is associated with a ground-truth attribute value $y \in Y = \{0, 1, \cdots, k-1\}$. For instance, for gender attribute, $Y = \{0{:}\text{male}, 1{:}\text{female}\}$. Edges $E$ can be encoded as an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{A}_{ij} = \{0, 1\}$. That is, if $(v_i, v_j) \in E$, then $\mathbf{A}_{ij} = 1$; otherwise, $\mathbf{A}_{ij} = 0$. Given $\mathbf{A}$, $\mathbf{X}$, and $V_l$ with attribute values $\mathbf{y}_l$, a GNN model $\mathbf{Z} = f_{\mathbf{W}}(\mathbf{A}, \mathbf{X})$ ($\mathbf{Z} \in \mathbb{R}^{n \times k}$ and $k = |Y|$) is well trained to predict the attribute value for each node in $V_u$ by minimizing the training loss as follows

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmin}}\ \mathcal{L}_{\text{gnn}}(f_{\mathbf{W}}(\mathbf{A}, \mathbf{X}), \mathbf{y}_l) = \underset{\mathbf{W}}{\operatorname{argmin}}\ l(\mathbf{Z}_l, \mathbf{y}_l) + \lambda \|\mathbf{W}\|_2^2 \tag{3.7}$$

where $\mathbf{W}$ is the trainable weight matrix, and $l(\cdot, \cdot)$ is the loss function (e.g., cross-entropy loss). Here, a GNN model $f_{\mathbf{W}}(\mathbf{A}, \mathbf{X})$ can be specified as graph convolutional networks (GCNs) [73], graph attention networks (GATs) [140], or others [104, 105, 141]. GNNs can be applied under inductive and transductive settings. In this work, we focus on transductive inferences where all node connections and features are accessible during training.

#### 3.2.2.2 Graph Adversarial Attack for Attribute Protection

Given a private attribute, a graph adversarial attack attempts to perturb the graph to obfuscate that attribute and prevent GNN-based inference attack models from correctly identifying users' private attribute values. Generally, it modifies an original graph $G$ with respect to its graph structure and/or node features to an adversarial graph $\hat{G} = (\hat{\mathbf{A}}, \hat{\mathbf{X}})$ [110, 113, 114], such that the test loss over nodes in $V_u$ can be maximized as follows

$$\begin{aligned} &\max_{\hat{\mathbf{A}}, \hat{\mathbf{X}}} \ \mathcal{L}_{\text{atk}}(f_{\mathbf{W}^*}(\hat{\mathbf{A}}, \hat{\mathbf{X}}), \mathbf{y}_u) \\ &s.t. \ \mathbf{W}^* = \underset{\mathbf{W}}{\text{argmin}} \ \mathcal{L}_{\text{gnn}}(f_{\mathbf{W}}(\hat{\mathbf{A}}, \hat{\mathbf{X}}), \mathbf{y}_l), \ \|G - \hat{G}\|_0 \leq \Delta \end{aligned} \tag{3.8}$$

where a budget constraint $\Delta$ is imposed on the perturbations to limit the number of changes over node features and edges and ensure the imperceptibility of attacks.

Clearly, this is a challenging bi-level optimization problem: the attacker aims to maximize the test loss achieved after optimizing the model parameters on the modified graph $\hat{G}$. Maximizing the test loss by modifying graph components is not straightforward as the graph parameters are constrained already. While perturbing graph would affect the optimized graph parameters computed in the prior step that makes this problem hard to solve; also, the action space of the attacker from $G$ to $\hat{G}$ are discrete, enforcing vast combinatorial search [114]. Even worse, these attacks based on either graph structure or node feature manipulations are impractical to be applied in real-world social graph setting: (1) user nodes usually encode multi-modal data (e.g., profiles, posts, and other activities), where perturbations computed from the feature space are hard to map into user information space in an end-to-end manner; (2) due to limited access to large-scale social networks (especially for ones built on private interactions like Facebook), it is unreasonable to assume that users can alter any relationship as they wish. By contrast, users' attribute values can be much easier to manipulate through users' self-reporting. It is necessary for inference attackers to collect initial attribute values for training, while these attribute values on social networks generally come from users' self-reporting. Therefore, attribute value manipulation has a direct impact on the model training and effectiveness for GNN-based inference attacks. Recent studies [74, 139] showed that flipping a few training labels successfully dragged down the node classification accuracy to a great extent for graph-learning models, which, however, can merely apply to binary classification tasks. To this end, in this project, we would like to formulate an attack-effective yet cost-efficient attribute-obfuscating attack on social graphs to protect users'

private attributes in practice, which specifically addresses the aforementioned challenges.

### 3.2.3 Attribute-Obfuscating Attack for User Privacy Protection

In this section, we first identify attribute-obfuscating attack goal with four underlying challenges for the problem formulation; to solve these challenges, we detail our technical steps of how we craft a graph adversarial poisoning attack AttrOBF to protect attribute privacy against GNN-based inferences on social networks. The overview of our proposed method AttrOBF is illustrated in Figure 3.10.

#### 3.2.3.1 Attack Goal and Challenges

In our application setting, AttrOBF is designed to obfuscate a small fraction of optimal training users' attribute values so as to maximally decrease the overall performance of GNN-based attribute inferences trained on the modified graph. More specifically, given a target attribute with either binary or multiple classes, the goal is to have the test users classified as any attribute value different from the true one. In this regard, we can update the general graph adversarial attacks in Eq. (3.8), and the final objective function of AttrOBF has the following form

$$\begin{aligned} &\min_{\Phi(\mathbf{y}_l)} -\mathcal{L}_{\text{atk}}(f_{\mathbf{W}^*}(\mathbf{A},\mathbf{X}),\mathbf{y}_u) \\ s.t.\ &\mathbf{W}^* = \operatorname*{argmin}_{\mathbf{W}} \mathcal{L}_{\text{gnn}}(f_{\mathbf{W}}(\mathbf{A},\mathbf{X}),\Phi(\mathbf{y}_l)) \\ &\|\Phi(\mathbf{y}_l)-\mathbf{y}_l\|_0 \le \epsilon n_l \end{aligned} \tag{3.9}$$

where $\Phi(\mathbf{y}_l)$ denotes the attribute obfuscating operation on the training attribute values $\mathbf{y}_l$, and $\epsilon$ is the obfuscating rate to $n_l$ to ensure that AttrOBF is unnoticeable. Eq. (3.9) indicates the objective of AttrOBF that directly relates to the loss maximization on the test attribute values $\mathbf{y}_u$. Also, AttrOBF only performs changes to the training attribute values $\mathbf{y}_l$; hence we treat the graph structure $\mathbf{A}$ and node features $\mathbf{X}$ as two constants during our attack formulation. Accordingly, Eq. (3.9) poses four unique challenges to the design of our attack AttrOBF as follows.

- **Unavailability of Test Attribute Annotations.** AttrOBF tries to decrease the generalization performance of GNNs on the unannotated node set with respect to a specific private attribute. Obviously, the test attribute values $\mathbf{y}_u$ are not available for the straightforward leverages. In other words, we cannot directly optimize the

test loss using the ground truth. One way to approach this is to select subset of training attribute values as test ones, but this may lead to a suboptimal solution with optimal training attribute values being potentially excluded for obfuscating operation.

- **Black-box setting.** AttrOBF is put under the black-box setting, where it is not aware of the GNN model $f_{\mathbf{W}}(\cdot, \cdot)$ used by the inference attackers, including model choice, architecture, and parameters. As AttrOBF is a data poisoning attack while we aim to prevent inference attackers from disclosing users' private attribute values on our modified social networks, it is reasonable to assume that AttrOBF has access to the social graph data with respect to $\mathbf{A}$, $\mathbf{X}$, and $\mathbf{y}_l$, which will be collected by inference attackers after attribute obfuscating in the real-world scenarios.

- **Bi-level optimization.** The problem formulation in Eq. (3.9) is of bi-level nature: the optimization on the attack loss $\mathcal{L}_{\text{atk}}$ is achieved after the optimization on the classification loss $\mathcal{L}_{\text{gnn}}$. In this respect, maximizing the test loss to obtain the optimal attribute obfuscating operation $\Phi(\mathbf{y}_l)$ requires retraining the GNN model, while the GNN model parameters $\mathbf{W}^*$ is constrained by the obfuscating operation $\Phi(\mathbf{y}_l)$ on the training attribute values. Optimizing such a bi-level problem is highly challenging by itself.

- **Non-differentiable obfuscating operation.** In our graph setting, the training attribute data and the action space of the attribute obfuscating are discrete: the training attribute values are $\mathbf{y}_l = \{0, 1, \cdots, k-1\}^{n_l}$, and the possible actions are attribute value changes from the current one to any others. This makes the action space of the problem vast: given the maximum allowed training attribute value changes $\epsilon n_l$, the number of possible attacks is in $O((k-1)^{\epsilon n_l} n_l^{\epsilon n_l})$; exhaustive search is clearly infeasible, while greedy search easily leads to sub-optimal solution. Gradient-based methods can avoid the combinatorial search; however, discrete obfuscating operation $\Phi(\mathbf{y}_l)$ is non-differentiable in the attack objective, preventing AttrOBF from directly applying gradients to optimize the test loss.

#### 3.2.3.2 Test Attribute Value Prediction

Transductive inferences over a graph imply that all node connections and features are accessible during training. Thus, we can use those annotated data to learn a GNN model described in Eq. (3.7) to estimate attribute values $\mathbf{y}_u$ of the unannotated or test nodes

$V_u$:

$$\mathbf{y}_u \approx \mathbf{y}_u^* = \underset{i \in Y}{\operatorname{argmax}}\ \mathbf{Z}_{u,i},\ \mathbf{Z} = f_{\mathbf{W}}(\mathbf{A}, \mathbf{X}) \tag{3.10}$$

The advantage yielded here is that we can designate the surrogate model, which will be introduced in Section 3.2.3.3, as $f_{\mathbf{W}}(\mathbf{A}, \mathbf{X})$ in Eq. (3.10) to estimate $\mathbf{y}_u$; if the adversarial attack formulated in a self-learning manner (i.e., using these predicted attribute values) has a high test error, it is very possible to also generalize poorly with the same surrogate model used to perform AttrOBF over the same graph. It is worth noting that only the attribute values $\mathbf{y}_l$ of the training nodes $V_l$ are used to optimize the GNN model, while the test attribute annotations $\mathbf{y}_u$ from estimation are only used to maximize the test loss for attack formulation.

#### 3.2.3.3 Surrogate Model

Under the black-box setting, we use two-layer Simple Graph Convolution (SGC) [142] as a surrogate model to perform our attribute-obfuscating attack on social graphs. Specifically, SGC is a linearized two-layer GCN:

$$\mathbf{Z} = f_{\mathbf{W}}(\mathbf{A}, \mathbf{X}) = \operatorname{softmax}(\hat{\mathbf{A}}^2\mathbf{X}\mathbf{W}),\ \mathbf{Z} \in \mathbb{R}^{n \times k} \tag{3.11}$$

where $\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}$, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, and $\mathbf{D}$ is the diagonal degree matrix defined on $\tilde{\mathbf{A}}$, i.e., $\mathbf{D}_{ii} = \sum_{j=1}^{n} \tilde{\mathbf{A}}_{ij}$.

There are three reasons behind this surrogate model choice: (1) SGC removes the non-linearity between GCN layers, which not only makes the model more tractable with less unnecessary complexity, but also captures the idea of graph convolutions (as demonstrated in [142], compared to those regular GNNs like GCN [73], GAT [140], FastGCN [107], SGC achieves the comparable or better test accuracy on different classification tasks); (2) SGC has been widely deployed as surrogate model in some successful graph adversarial attack formulations [74, 113, 114]; (3) SGC of linearity provides a simple closed form solution for $\mathbf{W}^*$, and thus transforms the bi-level optimization in Eq. (3.9) into single-level, which will be discussed in the following subsection. Due to transferability in adversarial machine learning [72], the attribute obfuscating operation optimized to mislead the surrogate model is very likely to degrade the real attackers' inference models.

#### 3.2.3.4 Closed Form Solution

To solve the aforementioned bi-level optimization, nettack [113] trains a fixed surrogate model to reduce the attack to the problem simply built upon $\mathcal{L}_{\text{atk}}$; metattack [114] approximates the attack by choosing $\mathcal{L}_{\text{gnn}}$ as an alternate of $\mathcal{L}_{\text{atk}}$, arguing that a model of high training loss very likely misclassifies test nodes; some other attacks [74,139] compute the closed form of graph learning models and transform the bi-level optimization into single-level. Here, we leverage the closed form transformation idea to obtain $\mathbf{W}^*$ and simplify the optimization on $\mathcal{L}_{\text{atk}}$.

Based on Eq. (3.7), Eq. (3.9), and Eq. (3.11), $\mathbf{W}^*$ can be rewritten as

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmin}}\ l((\hat{\mathbf{A}}^2\mathbf{X})_l\mathbf{W}, \Phi(\mathbf{y}_l)) + \lambda\|\mathbf{W}\|_2^2 \tag{3.12}$$

After replacing the loss function $l(\cdot,\cdot)$ with mean square loss function, and considering attribute obfuscating operation $\Phi(\mathbf{y}_l)$ as an $n_l \times k$-dimensional matrix where each row is a one-hot vector specifying new attribute value, Eq. (3.12) can be further updated as

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmin}}\ \frac{1}{n_l}\|(\hat{\mathbf{A}}^2\mathbf{X})_l\mathbf{W} - \Phi(\mathbf{y}_l)\|_2^2 + \lambda\|\mathbf{W}\|_2^2 \tag{3.13}$$

In this way, we can approximately obtain the closed form of $\mathbf{W}^*$ through the derivation as follows.

$$\begin{aligned}
&\frac{1}{n_l}\frac{\partial}{\partial\mathbf{W}}(\|(\hat{\mathbf{A}}^2\mathbf{X})_l\mathbf{W} - \Phi(\mathbf{y}_l)\|_2^2 + \lambda\|\mathbf{W}\|_2^2) = 0 \\
\Longrightarrow\quad & (\hat{\mathbf{A}}^2\mathbf{X})_l^T((\hat{\mathbf{A}}^2\mathbf{X})_l\mathbf{W} - \Phi(\mathbf{y}_l)) + \lambda\mathbf{W} = 0 \\
\Longrightarrow\quad & (\hat{\mathbf{A}}^2\mathbf{X})_l^T(\hat{\mathbf{A}}^2\mathbf{X})_l\mathbf{W} + \lambda\mathbf{W} = (\hat{\mathbf{A}}^2\mathbf{X})_l^T\Phi(\mathbf{y}_l) \\
\Longrightarrow\quad & \mathbf{W}^* = ((\hat{\mathbf{A}}^2\mathbf{X})_l^T(\hat{\mathbf{A}}^2\mathbf{X})_l + \lambda\mathbf{I})^{-1}(\hat{\mathbf{A}}^2\mathbf{X})_l^T\Phi(\mathbf{y}_l) \\
\Longrightarrow\quad & \mathbf{W}^* = \mathbf{K}\Phi(\mathbf{y}_l)
\end{aligned} \tag{3.14}$$

where we use $\mathbf{K} = ((\hat{\mathbf{A}}^2\mathbf{X})_l^T(\hat{\mathbf{A}}^2\mathbf{X})_l + \lambda\mathbf{I})^{-1}(\hat{\mathbf{A}}^2\mathbf{X})_l^T$ for the sake of simplicity. Given the closed form of $\mathbf{W}^*$, the bi-level optimization of AttrOBF in Eq. (3.9) can be updated as the following single-level optimization on $\Phi(\mathbf{y}_l)$.

$$\begin{aligned}
&\min_{\Phi(\mathbf{y}_l)} -\mathcal{L}_{\text{atk}}(f_{\mathbf{W}^*}(\mathbf{A},\mathbf{X}),\mathbf{y}_u) \Rightarrow \\
&\min_{\Phi(\mathbf{y}_l)} -l((\hat{\mathbf{A}}^2\mathbf{X})_u\mathbf{K}\Phi(\mathbf{y}_l),\mathbf{y}_u) + \lambda\|\Phi(\mathbf{y}_l)\|_2^2 \\
s.t.\ &\|\Phi(\mathbf{y}_l) - \mathbf{y}_l\|_0 \le \epsilon n_l
\end{aligned} \tag{3.15}$$

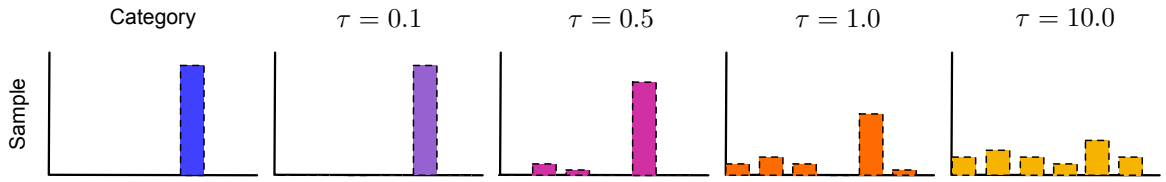



Figure 3.11: Relation between the Gumbel-Softmax distributions and one-hot-encoded categorical distribution: when $\tau \to 0$, samples from Gumbel-Softmax distributions are identical to the one from categorical distribution, i.e., one-hot vectors. When increasing temperatures, Gumbel-Softmax samples are more close to uniform [2].

#### 3.2.3.5 Gumbel Estimator

To solve the optimization problem in Eq. (3.15), the attribute obfuscating operation $\Phi(\mathbf{y}_l)$ is the key component; however, as discussed in Section 3.2.3.1, $\Phi(\mathbf{y}_l)$ is discrete thus non-differentiable, which means that we cannot directly use gradient-based methods to make updates on $\Phi(\mathbf{y}_l)$. Categorical variables are a natural choice for representing discrete structure in the world [2]. Therefore, to facilitate closed form solution in Section 3.2.3.4, we consider $\Phi(\mathbf{y}_l)$ as an $n_l \times k$-dimensional matrix, each row of which is represented as a one-hot vector to indicate the new attribute value changed from others or self. From the probabilistic perspective, we can model each attribute obfuscating operation as a categorical distribution, and this one-hot vector can be then sampled from $k$ label probabilities $(p_0, p_1, \cdots, p_{k-1})$, where the position of 1 (i.e., the best obfuscating operation) is decided by the highest probability: $\texttt{one\_hot}(\text{argmax}_i[p_i])$.

In other words, given the categorical distribution $\mathbf{P} \in \mathbb{R}^{n_l \times k}$, the test loss of AttrOBF defined in Eq. (3.15) is an expectation over categorical variables.

$$\min_{\mathbf{P}} -\mathcal{L}_{\text{atk}}(\mathbf{P}) \Rightarrow \min_{\mathbf{P}} -\mathbb{E}_{\Phi(\mathbf{y}_l)\sim\mathbf{P}} l((\hat{\mathbf{A}}^2\mathbf{X})_u \mathbf{K}\Phi(\mathbf{y}_l), \mathbf{y}_u) + \lambda\|\mathbf{P}\|_2^2 \tag{3.16}$$

The categorical sampling $\Phi(\mathbf{y}_l) \sim \mathbf{P}$ is still non-differentiable, which is not able to backpropagate through samples. To solve Eq. (3.16), we need to find a good gradient estimator to replace the non-differentiable samples with differentiable ones. To this end, we use Gumbel estimator [2] to draw samples $\Phi(\mathbf{y}_l)$ from $\mathbf{P}$ in a simple and efficient way. Different from performing argmax to search for the maximal probability, the Gumbel estimator utilizes Gumbel-Softmax function to generate continuous differentiable approximation to argmax. Specifically, let $\boldsymbol{\phi}$ (one row of $\Phi(\mathbf{y}_l)$) be sampled from the

corresponding categorical distribution $\mathbf{p}$ (one row of $\mathbf{P}$); $\boldsymbol{\phi}$ is approximated as

$$\phi_i = h(\mathbf{p}, \mathbf{g}) = \frac{\exp\left((\log(p_i) + g_i)/\tau\right)}{\sum_{j=0}^{k-1} \exp\left((\log(p_j) + g_j)/\tau\right)}, \text{ for } i = 0, 1, \cdots, k-1 \tag{3.17}$$

where $\mathbf{g} \sim \text{Gumbel}(0, 1)$ is Gumbel distribution, and $\tau$ is the temperature controlling the steepness of softmax function. As the temperature increases, the expected value converges to a uniform distribution over the categories; on the contrary, as $\tau$ approaches 0, samples from the Gumbel-Softmax distribution become one-hot, which is illustrated in Figure 3.11. Monte Carlo sampling from $\mathbf{g}$ makes Gumbel estimator unbiased and low variance [139]. Let $\mathbf{G} = [\mathbf{g}_0, ..., \mathbf{g}_{k-1}]^T$; by replacing $\Phi(\mathbf{y}_l)$ with $h(\mathbf{P}, \mathbf{G})$, the final test loss of AttrOBF is updated as

$$\min_{\mathbf{P}} -\mathcal{L}_{\text{atk}}(\mathbf{P}) \Rightarrow \min_{\mathbf{P}} -\mathbb{E}_{\mathbf{G}} l((\hat{\mathbf{A}}^2\mathbf{X})_u \mathbf{K} h(\mathbf{P}, \mathbf{G}), \mathbf{y}_u) + \lambda \|\mathbf{P}\|_2^2 \tag{3.18}$$

Accordingly, the derivative of $-\mathcal{L}_{\text{atk}}(\mathbf{P})$ regarding the categorical distribution $\mathbf{P}$ can be computed in an approximate way.

$$-\frac{\partial \mathcal{L}_{\text{atk}}(\mathbf{P})}{\partial \mathbf{P}} \approx -\frac{\partial}{\partial \mathbf{P}} \left[ l((\hat{\mathbf{A}}^2\mathbf{X})_u \mathbf{K} h(\mathbf{P}, \mathbf{G}), \mathbf{y}_u) + \lambda \|\mathbf{P}\|_2^2 \right] \tag{3.19}$$

The problem in Eq. (3.19) is differentiable and tractable. Therefore, it can be easily solved by gradient-based methods (e.g., stochastic gradient descent, Adam).

After the categorical distribution $\mathbf{P}$ is optimally updated, the attribute obfuscating operation $\Phi(\mathbf{y}_l)$ is uniquely defined as

$$\Phi(\mathbf{y}_l) = \texttt{one\_hot}\ (\text{argmax}\ (\mathbf{P}, \text{axis} = 1)) \tag{3.20}$$

Note that, $\Phi(\mathbf{y}_l)$ indicates the obfuscating operation on the whole training attribute values $\mathbf{y}_l$. As specified in Eq. (3.9) and Eq. (3.15), to ensure the imperceptibility of attack, the attribute obfuscating operation is constrained by $\|\Phi(\mathbf{y}_l) - \mathbf{y}_l\|_0 \leq \epsilon n_l$. That is, the number of maximum allowed training attribute value changes is $\epsilon n_l$. As such, we leverage $\Phi(\mathbf{y}_l)$ and $\mathbf{P}$ to decide the actual attribute obfuscating: we first collect all new training attribute values from $\Phi(\mathbf{y}_l)$ that are different from the original and their corresponding probabilities from $\mathbf{P}$, and then use those new attribute values with top $\epsilon n_l$ highest probabilities to update $\mathbf{y}_l$ so as to guarantee the optimal operation. Algorithm 3 illustrates our proposed attribute-obfuscating attack AttrOBF to protect attribute privacy on social networks. As graph structure $\mathbf{A}$ and node features $\mathbf{X}$ are

**Algorithm 3:** AttrOBF for attribute privacy protection.

**Input:** $G = (\mathbf{A}, \mathbf{X})$: Social graph $G$ with graph structure $\mathbf{A}$ and user features $\mathbf{X}$, $V_l$: $n_l$ training user nodes with attribute values $\mathbf{y}_l$, $V_u$: $n_u$ test user nodes without attribute values, $\epsilon$: obfuscating rate, $\tau$: temperature parameter, $T$: epochs.

**Output:** $\mathbf{y}_l$: the obfuscated training attribute values.

Train a GNN model using $\mathbf{A}$, $\mathbf{X}$ and $\mathbf{y}_l$ through Eq. (3.11);
Estimate $\mathbf{y}_u$ for the unannotated nodes $V_u$;
Pre-calculate $\hat{\mathbf{A}}^2\mathbf{X}$;
Pre-calculate $\mathbf{K} = ((\hat{\mathbf{A}}^2\mathbf{X})_l^T(\hat{\mathbf{A}}^2\mathbf{X})_l + \lambda\mathbf{I})^{-1}(\hat{\mathbf{A}}^2\mathbf{X})_l^T$;
**for** *each epoch* $t \leq T$ **do**
  Sample $\mathbf{G} \sim \text{Gumbel}(0, 1)$;
  Calculate $h(\mathbf{P}, \mathbf{G})$ using Eq. (3.17);
  Calculate test loss $-\mathcal{L}_{\text{atk}}(\mathbf{P}) \approx -l((\hat{\mathbf{A}}^2\mathbf{X})_u\mathbf{K}h(\mathbf{P}, \mathbf{G}), \mathbf{y}_u) + \lambda\|\mathbf{P}\|_2^2$;
  Update $\mathbf{P}$ by minimizing $-\mathcal{L}_{\text{atk}}(\mathbf{P})$;
**end**
$\Phi(\mathbf{y}_l) = \texttt{one\_hot}\ (\text{argmax}\ (\mathbf{P}, \text{axis} = 1))$;
Update $\mathbf{y}_l$ using new attribute values in $\Phi(\mathbf{y}_l)$ with top $\epsilon n_l$ highest probabilities in $\mathbf{P}$;

constants during attribute-obfuscating attack, we can pre-calculate $\hat{\mathbf{A}}^2\mathbf{X}$ and $\mathbf{K}$ using $O(\max(n^3, d^3))$, which significantly decreases the time complexity for each optimization iteration to $O(n_l n_u d)$ ($k \ll d$). Therefore, this efficient attack strategy has implications on its applicability for attribute protection on large social networks in practice.

## 3.2.4 Experiments

In this section, we evaluate the effectiveness of AttrOBF for protecting users' attribute privacy on social networks, and compare it with other baselines. We also investigate the impacts of the hyperparameters and the transferability of AttrOBF on different models.

### 3.2.4.1 Experimental Setup

**Datasets.** We collect three real-world social network datasets to conduct our experiments: Polblogs [143], Yale [144], and Rochester [144]. Polblogs represents a political blog network where their attribute values indicate political view of each user. Yale and Rochester datasets collect all the facebook friendships of Yale University and Rochester University as well as some user attributes, in which career, gender, class year serve as private attributes in our setting. To make our results comparable, we closely follow the data

Table 3.7: Comparing statistics of the three social network datasets with total five attribute settings.

| Dataset | Attr. | Nodes | Edges | Classes | Train./Val./Test |
|---|---|---|---|---|---|
| Polblogs | Politics | 1,490 | 19,025 | 2 | 40/500/950 |
| Yale | Career<br>Class-year | 8,578 | 405,450 | 2<br>6 | 20 × classes/500/1000 |
| Rochester | Gender<br>Class-year | 4,563 | 167,653 | 2<br>5 | 20 × classes/500/1000 |

setting in previous works [73, 140, 145]: we train the GNN models with all node features and 20 annotated nodes per class, and use another 500 annotated nodes as validation set. For testing, we randomly sample 1,000 nodes to evaluate the performance. Table 3.7 presents the statistics of the datasets.

**Baseline methods and parameter settings.** To the best of our knowledge, graph adversarial attacks via modifications on multi-class annotations have not yet been explored. Thus, we formulate two baselines in this regard to compare against our method AttrOBF: (1) Random attribute-obfuscating attacks (**Rand-obf**) where we randomly select a number of training nodes and obfuscate their attribute values to a random one. (2) Degree-based attribute-obfuscating attacks (**Deg-obf**) where we obfuscate the training nodes with the highest degrees because we believe these nodes play a more important role in the information propagation for GNNs than those with lower degrees; similarly, for all inference settings, we modify the attribute values of the selected nodes to a random one. Note that, as we only focus on attribute obfuscating, those attacks manipulating graph structure or node features (e.g., nettack [113] and metattack [114]) are not comparable here. Following the baseline designs in [74], in order to investigate how different components affect the performance of our proposed method, we further formulate two variants by replacing surrogate model and loss function: (1) **AttrOBF-lp** follows the same attack steps of AttrOBF except that we use label propagation as our surrogate model, which accordingly updates the closed form in Eq. (3.14) and single-level optimization in (3.15). (2) **AttrOBF-cse** replaces mean square error in loss function to cross-entropy, which updates the final test loss of AttrOBF in Eq. (3.18). In our parameter settings, we set the optimization epoch in AttrOBF as 1,000 and training epoch of GNN models as 200. The temperature parameter for Gumbel estimator $\tau$ introduced in Eq. (3.17) is set as 0.2 and $\lambda = 0.01$ for optimization.

**Attack model for attribute inference attacks.** Attackers conduct attribute inference attacks to disclose private attributes of users by learning a GNN model on public social network data. They have a variety of choices on graph neural networks when learning a model to perform inference attacks. In our work, we consider the most practical black-box scenario when protecting users' privacy from attackers. That is, we assume that we have no access to attacker's model. As explained in Section 3.2.3.3 and 3.2.3.4, we use SGC to solve black-box setting and closed form for AttrOBF. In our experimental setting, we train simple graph convolution (SGC) [142], graph convolutional network (GCN) [73], graph attention network (GAT) [140], and GCN-based label propagation network (GCN-LP) [146] to perform the inference attack. We mainly use GCN to evaluate the effectiveness of AttrOBF and the impacts of different parameters, while the comparisons among these four models are leveraged for transferability evaluation in Section 3.2.4.4. To be comparable, these four GNN models are of two-layer structure and the dimension of the hidden layer is set as 16. All other model parameters align with their original works [73, 140, 142, 146].

#### 3.2.4.2 Evaluation of AttrOBF

In this section, we evaluate the effectiveness of AttrOBF against attribute inference attacks and the impacts of different parameters under five attribute settings.

**Effectiveness.** In our experiments, we evaluate the effectiveness of AttrOBF under different attribute obfuscating rate $\epsilon$ as it plays the important role to determine the number of training nodes to modify in our adversarial attack. In particular, we test the results of five inference settings (i.e., Polblogs-politics, Yale-career, Yale-class, Rochester-class, Rochester-gender) while using AttrOBF to obfuscate the training attribute values with obfuscating rate $\epsilon \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, where 0.0 means no attack in place. In this experiment, we use test accuracy to evaluate attribute privacy protection performance. The lower test accuracy represents the better performance of our method. The experimental results are shown in Figure 3.12. As we can see from the results, the attribute inference accuracy for Polblogs-politics, Yale-career, Yale-class, Rochester-class and Rochester-gender on clean data is 81.1%, 88.1%, 84.5%, 82.8%, and 71.4%, which are relatively close to the state-of-the-art results on each dataset. Obviously, AttrOBF drastically decreases all the accuracy of inference attacks and thus achieves the goal of protecting users' attribute privacy on social networks.

**Impact of attribute obfuscating rate $\epsilon$.** Intuitively, when we enlarge the $\epsilon$, the number of the training node attribute values obfuscated by AttrOBF increases and

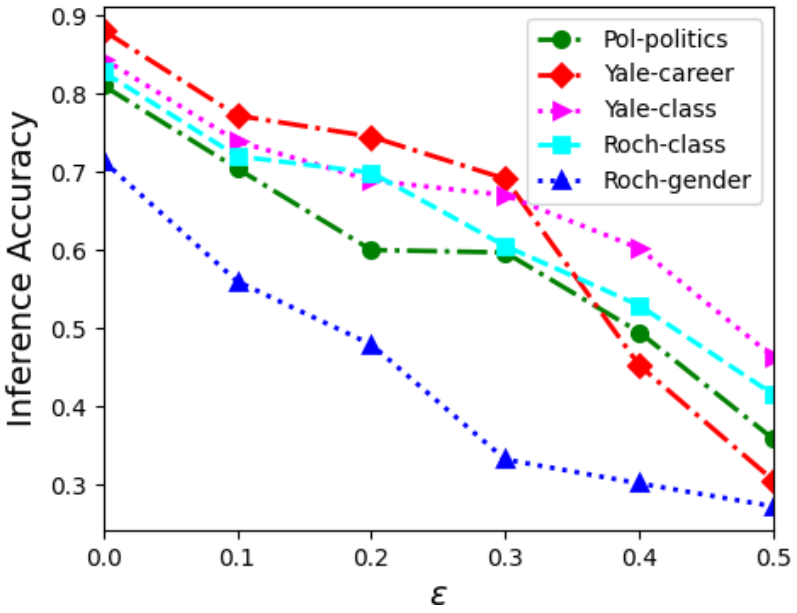



Figure 3.12: Test accuracy of all inference tasks on different attribute obfuscating rate $\epsilon$.

the accuracy of inference attacks should decrease. The results in Figure 3.12 confirm this point: as the obfuscating rate increases from 0.0 to 0.5, the inference accuracy of attack model drops 45.3% for Polblogs-politics, 57.6% for Yale-career, 41.2% for Yale-class, 41.3% for Rochester-class, and 44.2% for Rochester-gender. We can also observe that AttrOBF obtains better performance on binary inference settings such as Polblogs-politics, Yale-career and Rochester-gender than multi-class inference tasks like Yale-class and Rochester-class. The reason behind this could be that attacking space on multi-class social graphs is larger, which leads to more uncertainty and difficulty than binary classification problems that simply flipping annotations can directly impact on neighborhoods and thus more easily mislead the GNN model.

**Impact of temperature for Gumbel estimator $\tau$.** The temperature $\tau$ for Gumber estimator is an important parameter in our method that controls the effectiveness of the one-hot sampling. We gradually increase the value of $\tau$ in AttrOBF to analyze its impact to the attack performance. In the experiments, we assess the effectiveness of AttrOBF with temperature $\tau \in \{0.2, 0.5, 1.0, 5.0, 10.0\}$ in five inference settings when $\epsilon = 0.5$. We show the results in Figure 3.13. We can see from the figure that AttrOBF achieves the best performance when $\tau = 0.2$ for all inference tasks. As $\tau$ increases, the capability of our adversarial attack in alleviating the inference models is degraded. This is because

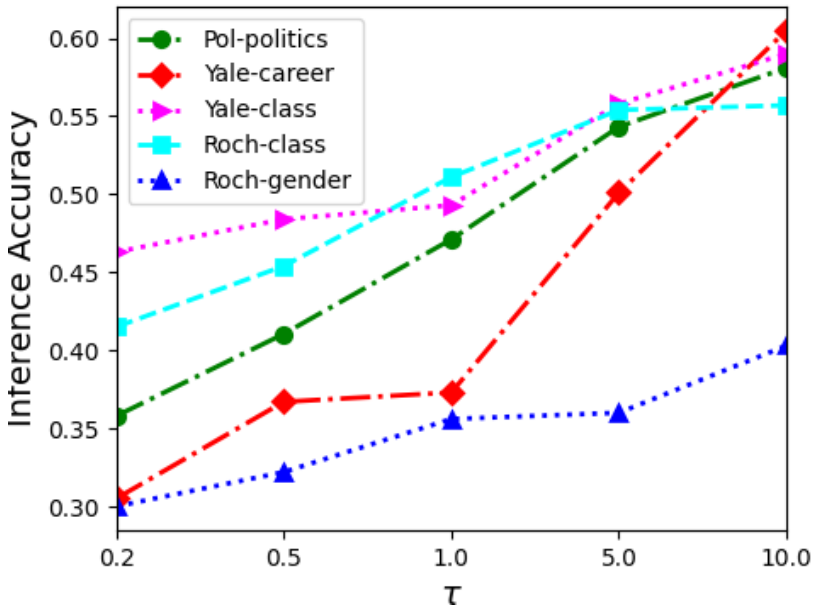



Figure 3.13: Evaluation results of AttrOBF under different values of temperature parameter $\tau$.

when we continuously amplify the $\tau$ value, Gumbel-Softmax distribution becomes closer to uniform distribution, which more significantly deviates from one-hot sampling and thus affects the effectiveness of attribute obfuscating operation. There is a trade-off between near-zero temperatures, where samples are identical to one-hot but the variance of the gradients is large as well. Based on this fact, we use $\tau = 0.2$ throughout the following evaluations.

**Impact of test attribute annotations $\mathbf{y}_u$.** Test labels are not easily accessible in real-world scenarios. Considering the practicability of our attack method, we use the prediction results of the surrogate model to estimate the test attribute values in all of our evaluations. In this part, we assume that we know the true test attribute annotations and investigate the impact of them on the performance of AttrOBF. We conduct the corresponding experiments over different inference tasks with the obfuscating rate $\epsilon = 0.5$ and results are shown in Table 3.8. We can observe that integrating true test attribute annotations in our objective loss function can obtain better attack results than the estimated ones, as the estimation might introduce extra loss between predictions and true attribute values. However, the inference accuracy difference between using true and estimated test attribute annotations seems not very significant. The reason behind this

Table 3.8: Evaluation on the impact of using true or estimated test attribute annotations (inference accuracy).

| Test labels | Pol-politics | Yale-career | Yale-class | Roch-class | Roch-gender |
|---|---|---|---|---|---|
| True | 33.1% | 29.3% | 43.0% | 40.9% | 25.7% |
| Estimated | 35.7% | 30.5% | 43.3% | 41.5% | 27.1% |

could be that the surrogate model's inference accuracy for different attribute settings is relatively high (i.e., 81.1%, 88.1%, 84.5%, 82.8%, and 71.4% for Polblogs-politics, Yale-career, Yale-class, Rochester-class and Rochester-gender respectively), which makes the estimation closer to ground truth. This implies that our method is not tightly coupled with true test attribute annotations, and can be easily feasible in practical applications.

#### 3.2.4.3 Comparisons with Other Attack Baselines

In this section, we compare our method AttrOBF against two baselines: Rand-obf and Deg-obf. For all methods, we set the obfuscating rate $\epsilon$ as 0.5, and use GCNs as the attack model to assess the inference accuracy. The results of five inference settings are presented in Table 3.9. We can observe that our method AttrOBF significantly outperforms Rand-obf on all inference tasks. Under Rand-obf attack, the inference accuracy only slightly decreases even when we obfuscate half of the training attribute values, which indicates that GCNs are quite robust to random label noise. This also benefits from the powerful learning capability of GCNs on graph data of embracing both node features and graph topological structure. Therefore, GCNs are resilient against random node obfuscating operations but still vulnerable to our well-designed adversarial attacks. AttrOBF also achieves better performance than Deg-obf attack, especially for multi-class inference problems. For instance, AttrOBF reduces the inference accuracy to 43.3% and 41.5% for Yale-class and Rochester-class while the results of Deg-obf attack are 53.1% and 54.2%, respectively. This is due to the fact that adversarial attribute values generated by AttrOBF are specifically derived from the goal of misleading the learning model, which are much more effective to degrade the performance of node classification, while Deg-obf identifies the degree information of nodes as the only influential factor for graph learning but ignores other conditions (e.g., node features) leveraged by GCNs.

Regarding to two variants, AttrOBF achieves better results than AttrOBF-lp for all classification settings. Compared to graph neural networks, label propagation only aggregates the label information from nodes' neighbors without considering the important

Table 3.9: Comparisons with other attack baselines and variants (inference accuracy).

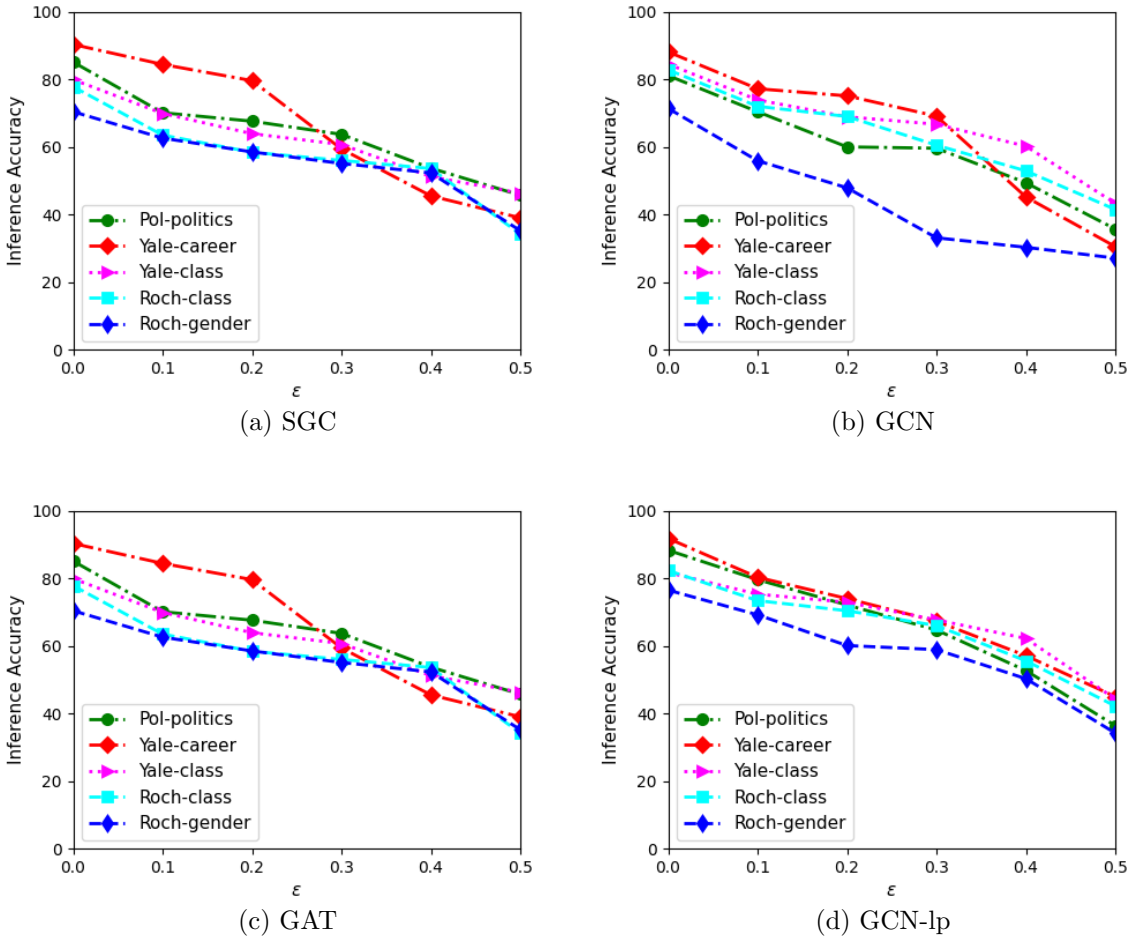
| Setting | Rand-obf | Deg-obf | AttrOBF-lp | AttrOBF-cse | **AttrOBF** |
|---|---|---|---|---|---|
| Pol-politics | 55.7% | 37.0% | 42.5% | 36.5% | **35.7%** |
| Yale-career | 61.2% | 47.2% | 49.4% | 38.6% | **30.5%** |
| Yale-class | 72.0% | 53.1% | 45.5% | 43.8% | **43.3%** |
| Roch-class | 69.6% | 54.2% | 43.5% | 42.1% | **41.5%** |
| Roch-gender | 46.7% | 42.1% | 39.9% | 31.0% | **27.1%** |

feature information. Therefore, choosing SGC to be our surrogate model to compute the closed form solution is more reasonable and effective. Another similar variant AttrOBF-cse can achieve comparable results but still slightly underperforms our method. The reason behind this small performance difference could be that mean square error can better formalize the discrepancy between ground truth and prediction results in the embedding space.

#### 3.2.4.4 Transferability of AttrOBF

Under the black-box setting, we don't know what model the attacker is using to infer private attributes. This naturally leads us to the question: can our attack strategy generalize to other inference attack models? To answer this question, in this evaluation, we explore the transferability of our method AttrOBF. Specifically, we deploy AttrOBF to obfuscate the training attribute values and generate adversarial graph on five attribute inference settings. Then we test the inference results of the poisoned data against four state-of-the-art GNN models, including SGC [142], GCN [73], GAT [140] and GCN-LP [146] under five obfuscating rates (i.e., $\epsilon = \{0.1, 0.2, 0.3, 0.4, 0.5\}$). To ensure our results are comparable, we build up these models with the same parameter and data settings.

The results presented in Figure 3.14 show that the adversarial attack performed by AttrOBF can successfully transfer to different graph neural networks. Our AttrOBF method learned on a linearized GCN (i.e., SGC) presents the similar effectiveness against different GNN models under the same inference setting. For example, when $\epsilon$ is set as 0.5, AttrOBF reduces the accuracy of SGC, GCN, GCN-LP to 35.6%, 35.7% and 36.4% on polblogs-politics inference attack and 33.5%, 27.1% and 34.2% on Rochester-gender inference setting. For Yale-career, the inference accuracy of all GNN models drops over 30% when increasing $\epsilon$ from 0.1 to 0.5. While for Yale-class and Rochester-class inference settings, the transferability of AttrOBF on four GNN models are very close and slightly




(a) SGC (b) GCN

(c) GAT (d) GCN-lp

Figure 3.14: Evaluation results: (a), (b), (c) and (d) specify the inference accuracy of SGC, GCN, GAT and GCN-lp while conducting AttrOBF on our surrogate model over different data settings; lower inference accuracy indicates better attack transferability.

underperform other inference tasks. On the other hand, the results also imply that the complexity of the surrogate model and the intrinsic adversarial vulnerability of the target model contribute to attack transferability: the attack results on SGC and GCN outperform those with more complex model structure such as GAT and GCN-LP. Since the target models are uncontrollable, when applying AttrOBF in practice, we may need to elaborate the surrogate model for better transferability. We leave it as our future exploration.

### 3.2.5 Impact, Applicability and Limitation

Our previous method formulation and experimental evaluations demonstrate the impact of our proposed graph adversarial attack solution for attribute privacy protection on social networks: (1) AttrOBF introduces low computational cost, which is feasible in large real-world social networks; (2) as graph structure and node features are not perturbed, the utilities of social networks regarding user activities and relationships are well preserved without any influence on other downstream tasks; (3) mere small yet optimal training annotation changes can effectively mitigate attribute inference attacks; (4) attribute obfuscating is easy to operate for both data publishers and users. Therefore, in practice, AttrOBF can work as an easy-to-use API provided on the social network server side that enables data publishers to either locally or globally manipulate user attribute values before making the social graphs publicly available, or warn users of potential attribute privacy threats such that users can proactively change their attribute information on the client side. In fact, users' self-obfuscating operations not only protect themselves from attribute disclosure but also contribute to other users' attribute obfuscation through social graph settings. Note that, some users are more willing to disclose their information rather than "mispresent" themselves; in this case, these attributes may not be strictly considered as privacy for them to protect.

Nonetheless, our approach also poses a limitation which we discuss as follows. We successfully perform AttrOBF on the annotated public social graph data in this work, while the real social media lacks the ground truth, which disables Adv4SG from generating the adversarial texts in a real-time fashion. To better obfuscate the attributes, we may need to first recognize the targets. Though attribute recognition is irrelevant for the scope of our work, it is an interesting future work to leverage attribute recognition for better protection solutions. In our experiments, we train some regular GNN-based attack models for attribute inferences on social networks. Though AttrOBF has been validated to be transferable to these GNNs, the attackers could take advantage of more advanced

and robust GNN models (e.g., adversarial training via latent perturbation [147]) to infer attributes and thus deteriorate AttrOBF. We acknowledge this limitation and leave the investigation on this arms race as our future work, yet it does not impact the great value and general validity of our new insight about leveraging adversarial attacks for attribute obfuscation and privacy protection on social networks in practice, as graph learning models of inherent vulnerability could always be evaded by more complicated and more sophisticated adversarial techniques.

# Chapter 4
# The Bad: Enhancing the Robustness of DNNs Against Adversarial Attacks

In this chapter, we discuss the bad aspect that adversarial machine learning brings to us. On the one hand, the existence of adversarial examples drives us to design more and more powerful attacks; on the other hand, it stimulates the appearance of new defense strategies and leads us to pursue more robust models or systems. In this respect, we introduce our work that focuses on improving the robustness of deep neural networks against adversarial attacks.

## 4.1 Watermarking-based Defense against Adversarial Attacks on Deep Neural Networks

The vulnerability of deep neural networks to adversarial attacks has posed significant threats to real-world applications, especially security-critical ones. Given a well-trained model, slight modifications to the input samples can cause drastic changes in the predictions of the model. Many methods have been proposed to mitigate the issue. However, the majority of these defenses have proven to fail to resist all the adversarial attacks. This is mainly because the knowledge advantage of the attacker can help to either easily customize the information of the target model or create a surrogate model as a substitute to successfully construct the corresponding adversarial examples. In this work, we propose a new defense mechanism that creates a knowledge gap between attackers and defenders by imposing a designed watermarking system into standard deep

neural networks. The embedded watermark is data-independent and non-reproducible to an attacker, which improves randomization and security of the defense model without compromising performance on clean data, and thus yields knowledge disadvantage to prevent an attacker from crafting effective adversarial examples targeting the defensive model. We evaluate the performance of our watermarking defense using a wide range of watermarking algorithms against four state-of-the-art attacks on different datasets, and the experimental results validate its effectiveness.

### 4.1.1 Introduction

Deep Neural Networks (DNNs) have been widely adopted in a variety of machine-learning tasks, ranging from computer vision, speech recognition [3, 4] to natural language processing and healthcare [9, 10]. Despite the remarkable performance these applications have achieved, DNNs remain vulnerable to adversarial attacks that design special imperceptible perturbations to the original inputs to fool state-of-the-art models. For example, Goodfellow et al. [1] demonstrated how to add a small perturbation to an image of panda that causes it to be recognized as a gibbon with high confidence. In a security-critical scenario, Evtimov et al. [22] successfully misled a classifier to misclassify a stop sign with some physical perturbations, which can be either graffiti or black and white strips, as a Speed Limit 45 sign.

In order to alleviate adversarial attacks, researchers have proposed a large body of defensive work. Some of them try to manipulate model properties through augmentation or regularization [1, 23, 28], or attempt to filter malicious examples by detecting or removing perturbations introduced to original examples [33, 34]. Most of these strategies are easy to compromise due to their simplicity and differentiable nature, with some impractical assumptions on the attacker's knowledge of the target model. In fact, the information about the target model is the key for most attack algorithms to craft adversarial examples, especially for those gradient-based attacks that require this information to calculate gradients through backpropagation. Recent studies [148, 149] have shown that randomization over the network layerwise structure or inputs enjoy the potential of obfuscating the gradients and thus mitigate the adversarial vulnerability. This naturally inspires us to take advantage of the randomization paradigm and increase the attacker's uncertainty to the target model to significantly hinder them from customizing the model information, such that the generated adversarial examples could be rendered as less effective as possible.

Based on the above observation, in this work, we consider the practical scenario about

the adversarial attacks, and design a defense mechanism by introducing the randomness and confidentiality of digital watermark to DNN models to incur the possible knowledge gap between the attacker and the defender. In this regard, we can lead to the attacker's knowledge disadvantage by introducing the secret watermarking scheme into the standard DNN model. Digital watermarking is a technique that embeds watermark information into the host image by modifying visually non-significant pixels, which is transparent, imperceptible, and robust. For the watermarking techniques, if a user has no embedding information, the watermark is very challenging to be detected and extracted [66]. In this respect, the attacker needs to craft adversarial examples from their self-trained surrogate models as it is not realistic for them to reproduce the defense model without confidential embedded information. The lack of knowledge about the defense system leads to the discrepancy and stochasticity between the surrogate and real models, making it more challenging for the attacker to successfully evade the defense model. Our proposed defense method enables us to train a DNN model that would not only preserve the inference performance on regular data, but also benefit from knowledge gap and randomization imposed on the learned protocol for better robustness against adversarial attacks. In summary, our work has the following major merits:

- We creatively leverage the concept of knowledge gap by introducing a watermarking system into the DNN model to obstruct the adversarial attacker from accessing the model gradient information. To the best of our knowledge, this is the first investigation to use watermarking techniques to counter adversarial attacks.
- The proposed watermarking-based defense improves the robustness of learning model against adversarial attacks while not compromising its performance on regular data. It is convenient for implementation without many additional computations and extra training or tuning requirements, and applicable to serve as a general defensive system for different learning models and networks.
- We systematically evaluate our method against adversarial attack algorithms in different scenarios and analyze the impacts of digital watermark on adversarial perturbations. We show that our proposed defense can effectively resist adversarial examples, especially for sophisticated ones.

### 4.1.2 Overview

#### 4.1.2.1 Deep Neural Networks

A deep neural network (DNN) is a function $y = f(x)$ that accepts an input $x \in \mathbb{R}^n$ and produces an output $y \in \mathbb{R}^m$, where in most of the cases, $m$ is equal to the number of classes. where $f$ significantly relies on model parameters $\theta$. The output of the network is computed using the softmax function, which ensures that the output vector y satisfies $0 \leq y_i \leq 1$ and $y_1 + y_2 + ... + y_m = 1$. The output vector y is thus treated as a probability distribution. In our notation, we define $F$ to be the full neural network including the softmax function, and a neural network typically consists of layers,

$$f(x) = \text{softmax}(\sigma_n(W_n\sigma_{n-1}(...\sigma_1(W_1x))) \tag{4.1}$$

where, at each layer $i$, $W_i$ corresponds to the model parameters and $\sigma_i$ is an activation function, usually non-linear, with $1 \leq i \leq n$. In our experiments, we focus primarily on networks that use a ReLU [150] activation function, as it is the most widely used one.

#### 4.1.2.2 Adversarial Examples

Given a valid input $x$, it is possible to find a similar input $x'$ such that $f(x') \neq f(x)$ yet $x$ and $x'$ are close according to specific distance metric. As such, various adversarial attack methods have been proposed. Fast Gradient Sign Method (FGSM) is designed to fast craft adversarial examples [1]. which is easily implemented, but cannot guarantee to generate close ones all the time. An example of FGSM attack with respect to a source input $x$ and true label $y$ is

$$x' = x + \epsilon \cdot \text{sign}(\bigtriangledown_x l(x, y)) \tag{4.2}$$

where $l(x, y)$ is the loss function used to train the classifier, and $\epsilon > 0$ is a small constant that governs the magnitude of distortions. For each pixel in the image, it will take one step of size $\epsilon$ in the direction of gradient sign. Projected Gradient Descent algorithm (PGD) is a successful extension of FGSM, Iterative FGSM is one of successful extensions of FGSM, which is also known as Projected Gradient Descent algorithm (PGD), which iteratively applies the small FGSM update, with the result being clipped by a sufficiently small constant. Specifically, it begins by setting $x^0 = x$, and then on each iteration $k$, $x^k$ is updated as

$$x^k = x^{k-1} + \epsilon \cdot \text{sign}(\bigtriangledown_{x^{k-1}} l(x^{k-1}, y)) \tag{4.3}$$

where $k = 1, ..., K$, and $x' = x^K$. The number of iterations $K$ is determined such that $f(x') \neq f(x)$. DeepFool [26] is another state-of-the-art adversarial example generation approach that projects input $x$ onto the nearest class boundaries iteratively to minimize the Euclidean distance between the input and adversarial examples. In addition, as it has been shown to be very effective in other works, the CW-L2 attack method proposed by [27] is an optimization-based attack that uses $L_2$-penalty term as its distance metric to find a minimum distortion $\delta$ for a given input:

$$\begin{aligned} &\min_{\delta}[\| \delta \|^2 + \lambda_c f(x+\delta)] \\ &s.t. \quad 0 \leq x_i + \delta_i \leq 1 \quad \forall i = 1, ..., N \end{aligned} \tag{4.4}$$

$\lambda_c$ is a suitable constant chosen by binary search, $x+\delta$ represents the adversarial example $x'$ we would like to find, and $f(\cdot)$ is an effective objective function

$$f(x') = \max(-\kappa, \max\{Z(x')_{f(x)} : t \neq f(x)\} - Z(x')_t) \tag{4.5}$$

where $\kappa$ denotes a margin parameter that controls the confidence in result, and $Z(x)_t$ is the logit (the value before the softmax layer) corresponding to class $t$.

#### 4.1.2.3 Digital Watermark

Digital watermarking is a technique used for the protection of digital work such as video, audio, and image [151]. In this technique, a secret payload (i.e., watermark) is embedded to the work using some watermarking algorithm that should be imperceptible, robust, and of high fidelity. Specifically, a watermarking algorithm consists of a watermark structure and an embedding algorithm. According to the modified value of the carrier, digital watermarking is divided into two major areas: spatial domain watermarking and frequency domain watermarking. For the sake of the imperceptibility and robustness, current image watermarking research mainly focuses on frequency domain watermarking techniques, where the image is represented as the form of frequency, and the watermark is embedded into the coefficients of the transformed image. In general, the frequency domain transform is considered to be more robust than that in spatial domain. Therefore, we explore a couple of frequency domain watermarking transforms in our defensive strategy.

### 4.1.3 Method Design

In this section, we present the detailed approach of how to design a watermarking-based defense and how to enhance the target model's robustness against adversarial attacks based on such strategy.

#### 4.1.3.1 Knowledge Gap

Given a DNN model $f$, the output label $y$ for an input $x$ is presented by $y = \arg\max_i f_i(x)$, where $f_i(x)$ is the confidence score of the predicted label $i$. Any adversarial attack that aims to alter the output label $y$ of the model regarding the input sample $x$ needs to change the confidence score $f_y(x)$ by adding the perturbation $\delta$ to $x$, such that the output prediction is changed by a fixed lower bound $\varepsilon$: $\|f_y(x) - f_y(x+\delta)\|_2 \geq \varepsilon$. According to the first-order approximation [152] and the DNN model's linear characteristics around the input samples [153], the difference caused by perturbation $\delta$ on $f_y(x)$ can be denoted as $f_y(x+\delta) - f_y(x) \approx \langle \nabla_x f_y, \delta \rangle$. Therefore, the minimal $l_p$-norm perturbation $\hat{\delta}_p$ ($p \in [1, \infty)$) required to change the output prediction by $\varepsilon$ can be approximated using Hölder inequality and $l_p$-norm projection as [152]:

$$\hat{\delta}_p \approx (\frac{\varepsilon}{\|\nabla_x f_y\|_q}) \partial(\|\nabla_x f_y\|_q) \tag{4.6}$$

where $l_p$-norm and $l_q$-norm are dual-norm with $\frac{1}{p} + \frac{1}{q} = 1$, and $\partial(\cdot)$ is the subgradient of the argument. Dabouei et al. [152] provided us with a more detailed solution of $\partial(\|\nabla_x f_y\|_q)$, such that Eq. (4.6) can be rewritten as [152]:

$$\hat{\delta}_p \approx (\frac{\varepsilon}{\|\nabla_x f_y\|_q})(\frac{|\nabla_x f_y|^{q-1} \odot \text{sign}(\nabla_x f_y)}{\|\nabla_x f_y\|_q^{q-1}}) \tag{4.7}$$

Clearly, we can gain the insight from Eq. (4.7) that the model gradient $\nabla_x f_y$ plays a very important role in the success of the adversarial attacks; it is essential to obfuscate $\nabla_x f_y$ to defend against such attacks. With this in mind, some significant efforts have been made in this regard to enforce useless gradients for generating adversarial examples [134], while randomized defenses [148, 149] with randomization over the network structure or inputs can restrain the attacker from correctly estimating the true gradient and thus failing to effectively mislead the model.

More specifically, due to the stochastic gradient caused by randomized model structure or inputs, the attacker cannot directly customize the defense model $f(x)$ but train their

own model $\hat{f}(\hat{x})$ to compute the gradient $\nabla_{\hat{x}}\hat{f}_{\hat{y}}$. As investigated, adversarial examples may be transferable so that some adversarial examples generated from $\hat{f}(\hat{x})$ may lead to misclassification on $f(x)$ as well [72]. Such a property allows the attacker to take $\hat{f}(\hat{x})$ as a surrogate model to craft attack samples. However, since the surrogate model is a rough approximation of the target distribution (i.e., $\nabla_{\hat{x}}\hat{f}_{\hat{y}} \neq \nabla_x f_y$), there is always a discrepancy between the approximation and the real one, which we consider as the defense space. Our goal is to enlarge such space, deviate $\nabla_{\hat{x}}\hat{f}_{\hat{y}}$ from $\nabla_x f_y$, and make the adversarial perturbation less effective. Different from the previous studies, here we devise a fine-grained watermarking system to the DNN model to increase the knowledge gap between the attacker and the defender. As the watermark is transparent and undetectable with secret payload message and capacity, the embedded watermark information may cause the inputs $x$ and the corresponding gradient $\nabla_x f_y$ randomized, and expand the discrepancy between $f(x)$ and $\hat{f}(\hat{x})$. The DNN model $f(x)$ could thus explicitly change its classification boundary, and be resilient against the attacker's adversarial examples generated through the untrue estimation $\hat{f}(\hat{x})$.

#### 4.1.3.2 Watermarking-based Defense

For defenses that employ randomized transformations to the inputs, Athalye et al. [134] demonstrated that Expectation over Transformation (EOT) can be deployed to compute the gradient over the expected transformation to the inputs by optimizing the expectation over the transformation $\mathbb{E}_{t\sim T} f(t(x))$ (i.e., $t(\cdot)$ sampled from a distribution of transformations $T$). However, the distribution $T$ can merely model perceptual transformations, such as image cropping, viewpoint shifts and geometric changes. Different from the regular input transformations, watermark embedding, which imposes the random and secret payload message to the inputs in the abstract frequency domain, is imperceptible and irreversible for the attacker, and its distribution $T$ is thus difficult to be formulated. Even if some of the watermarked data is accidentally intercepted by the attacker, they are still unable to detect and extract all the watermark keys for reproduction. In this respect, the gradient obfuscation and knowledge gap caused by watermarking are effective to prevent adversarial attacks from directly or indirectly calculating model gradients.

In order to generate additional randomization benefits, instead of applying watermarking to the inputs (either clean or adversarial) using an unique payload message, we design our watermarking defense paradigm in an ensemble manner, the overview of which is illustrated in Figure 4.1. (1) In the training stage, we randomly split the training data into $k$ sets; for each set, we embed a watermark key into all inputs; different sets use

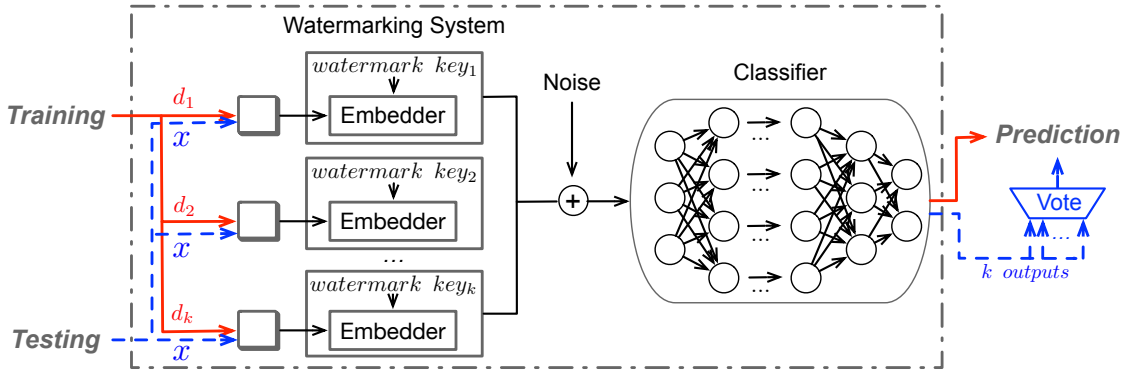



Figure 4.1: The overview of our defense framework devising a watermark system between the input and the DNN structure.

different keys, while all inputs enjoy the same embedder (a watermarking algorithm). All watermarked sets are leveraged for DNN model training. (2) Complying to the watermarking routine, we add small Gaussian noise with the interval of $[-0.1, 0.1]$ to each watermarked input and then rescale it to increase the difficulty of watermark detection. (3) In the testing stage, we feed the test data through watermarking system using $k$ specified watermark keys, and the trained DNN model to obtain $k$ different outputs respectively, which are aggregated later using voting method to approximate the final result. This may generate some regularization effect beyond randomization provided by model training. Algorithm 4 illustrates our proposed watermarking-based defense.

It's worth noting that the embedded secret watermark has no significant impact on the performance of the DNN model on regular classification task in our observation. In addition, the watermark keys are images randomly selected from the large image database (e.g., ImageNet [154]), which is independent from the input data. We further resize the watermark images into the same shape as the input data before embedding them. Following the second Kerckhoffs' cryptographic principle [155], we err on the side of overestimating the attacker's capability and excessively relax the limitation on the attacker's knowledge about the defense model (i.e., the worst-case where the watermarking algorithm and the watermark image database are also known to the attacker). As such, the attacker tends to search for the potential watermark keys to craft effective adversarial examples. The computational space for watermark searching would be as large as $k \times 256^{h \times w \times d}$, where 256 is the range of image pixel and $[h, w, d]$ represents the image shape. It forces the attacker to take an extremely long time and effort to evade the target model. Therefore, it is computationally infeasible to generate adversarial examples in such a cost-expensive fashion. Unlike the low-level image transformations [149], the

**Algorithm 4:** Watermarking-based defense.

**Input:** $D_{tr}$: training data, $D_{ts}$: test data (clean or adversarial), $f$: a standard DNN model, $\{w\}_{i=1}^{k}$: $k$ random watermark images, $g$: a watermarking algorithm, $\mu$: Gaussian noise.

**Output:** $\widehat{f}$: a defense model; $y$: output class.

$\{w\}_{i=1}^{k} \leftarrow$ convert $\{w\}_{i=1}^{k}$ to gray-scale images;
//**For the training stage**;
$\{D_{tr}\}_{i=1}^{k} \leftarrow$ split $D_{tr}$ into $k$ sets;
**for** $i = 1 \rightarrow k$ **do**
    **for** $j = 1 \rightarrow |D_{tr}^{i}|$ **do**
        $\widehat{x}_j = g(x_j, w_j) + \mu$;
    **end**
**end**
Train DNN model $f$ using the watermarked $\widehat{D}_{tr}$ as $\widehat{f}$;
//**For the testing stage**;
**for** $i = 1 \rightarrow |D_{ts}|$ **do**
    **for** $j = 1 \rightarrow k$ **do**
        $\widehat{x}_{ij} = g(x_i, w_j) + \mu$;
        $y_{ij} = \widehat{f}(\widehat{x}_{ij})$;
    **end**
    $y_i \leftarrow$ aggregation using voting on $\{y_{ij}\}_{j=1}^{k}$;
**end**
**return** The trained model $\widehat{f}$ and the test classes $y$;

watermarking also implicitly preserves the specific structure and meaningful pattern, which codes better with the uniqueness of the input images in our defense model and generates a better advantage to avoid the attacker's mimicry.

#### 4.1.3.3 Watermarking Implementation

When embedding a watermark to different images, we have to adopt a specific watermarking strategy for implementation. In this work, we investigate five different frequent domain watermarking algorithms in our watermarking system as follows.

- Discrete Fourier Transform (DFT) [156] decomposes an image in sine and cosine form. Since the magnitude and phase hold some information of the transformed image, we can accordingly modify them to embed the watermark. DFT is robust against geometric distortion and translation invariant.

- Discrete Wavelet Transform (DWT) [157] gives a multi resolution representation

of the image. When applying DWT to an image, it divides the image into two quadrants, i.e., high-frequency quadrant and low-frequency quadrant. This process repeats until the signal has been entirely decomposed. We embed the watermark into low-frequency coefficients as they contain the details of the original images.

- Singular Value Decomposition (SVD) is one of the most potent numeric analysis techniques that has been widely applied to digital image applications [158]. Given an image matrix, it can be transformed into three components. In the embedding procedure, the most significant coefficients in the component are modified to embed a watermark. After watermarking, it inversely transforms to reconstruct the watermarked image.

- DWT_SVD based watermarking algorithm [159] develops the DWT and SVD methods, which is a technique that clubs the properties of DWT and SVD. It not only increases the limited capacity of SVD but also reduce time consumption.

- DWT_DCT_SVD based watermarking algorithm [160] combines the properties of DWT, DCT and SVD algorithms and is robust against all sorts of attacks.

Besides the watermarking algorithms, the implementation of watermarking also significantly relies on the property of the data, e.g., the dimension of the image. The gray-scale image watermarking is convenient for implementation since all the intrinsic information of the gray-scale images is simply abstracted as pixels in a single component. Differently, the color image are generally represented as a red-green-blue (RGB) triplet, while the RGB values are more complex and are the only feasible data from them [161]. Considering that these three components are inter-correlated and RGB triple is also a biased representation of the color images, processing the RGB color information in parallel for each color component independently while ignoring the intrinsic properties contained in the interaction of different color channels may easily enforce information loss and thus lead to model performance degradation. To address this issue, we attempt two mapping solutions to transfer the color information into independent components instead of the $R - G - B$ components.

The first one is inspired by the work [162], where we employ Karhunen–Loeve Transform (KLT) to decorrelate RGB information of the color images. To apply KLT, each image is represented as a set of vectors $\mathbf{v}_i$ of size $d$ (e.g., $d = 32 \times 32$ if the dimension of the color image is $32 \times 32$), with $1 \leq i \leq 3$. As such, it is possible to calculate the

expected value of three vectors as follows:

$$\mathbf{m} = E[\mathbf{v}_i], \tag{4.8}$$

which would further facilitate computing the covariance matrix of size $3 \times 3$ of the centered vectors $(\mathbf{v}_i - \mathbf{m})$:

$$\mathbf{C} = E[(\mathbf{v}_i - \mathbf{m}) \cdot (\mathbf{v}_i - \mathbf{m})^T], \tag{4.9}$$

where the eigenvectors $\mathbf{a}_i$ and their associated eigenvalues $\lambda_i$ of the matrix $\mathbf{C}$ can be obtained to formulate a matrix $\mathbf{A}$ by descending order of the eigenvalues as $\mathbf{A} = (\mathbf{a}_1{}^T, \mathbf{a}_2{}^T, \mathbf{a}_3{}^T)$. The KLT of a vector $\mathbf{v}_i$ can be defined as

$$\mathbf{u}_i = \mathbf{A} \cdot (\mathbf{v}_i - \mathbf{m}), \tag{4.10}$$

and these vectors are uncorrelated [162]. Here we embed the watermark into the first component $\mathbf{u}_1$ after KLT transformation as it generally contains the most information, and then an inverse KLT is performed to reconstruct the watermarked images in the way:

$$\mathbf{v}_i = \mathbf{A}^{-1} \cdot \mathbf{u}_i + \mathbf{m}. \tag{4.11}$$

The second solution is mapping correlated RGB components to HSV [163], a color space designed to more closely align with the way human vision perceives color. HSV describes colors in terms of Hue, Saturation, and Value. Considering that HSV is a less correlated color space than RGB while objects in images have distinct hues and luminosities so that these features can be used to separate different image areas, we choose to convert RGB triple to HSV, embed the watermark to the Hue value, and then map HSV components back to RGB to construct the watermarked images.

### 4.1.4 Evaluation

In this section, we evaluate the efficacy of our proposed watermarking based defense model through performing experiments on benchmark image classification tasks, and compare our model with a wide variety of state-of-the-art approaches.

#### 4.1.4.1 Experimental Setup

**Datasets.** We test our model on three benchmark image classification datasets from the AI Science community: MNIST [164], Fashion-MNIST [165] and CIFAR-10 [166]. MNIST is a set of hand-written digits that contains 10 classes, 60,000 training and 10,000 test gray-scale images of the size $28 \times 28$, while Fashion-MNIST [165] is another standard dataset with more complex and diverse object structures, which also consists of 10 classes, 60,000 training and 10,000 test gray-scale images of $28 \times 28$. CIFAR-10 is composed of 60,000 $32 \times 32$ colour images in 10 classes. To be consistent with the previous work, we scale each pixel value to be in the range $[0, 1]$.

**DNN model and watermarking.** The DNN model trained in our experiments is a standard CNN classifier with 8 layers [167]. The architecture of our deployed model is not complicated because our ultimate goal is not to achieve the state-of-the-art image classification accuracy on the chosen dataset but measure the performance of classifying adversarial examples in the same settings. The overall classifier performance is calculated by the test accuracy. We evaluate our defensive method on four state-of-the-art adversarial attacks, i.e., FGSM [1], PGD [23], DeepFool [26] and CW-L2 [168]. We set the number of watermarking embedders $k$ in our system as 5. The watermark images used in either the attack model or the defense model are randomly selected from the ImageNet [154] with more than 14 million images, and processed through the watermarking system upon the dataset. For each watermarking path, we apply specific watermarking algorithm as the embedder to process the watermark embedding process to all the images in the dataset. To verify if watermarked training has any impact on the learning performance, we test the classification accuracy of our trained DNN system on the regular image data. We found that the classification results of different training dataset over different watermarking algorithms are consistent with the benchmark in standard case, where the test accuracy reaches 99.30% over MNIST, 91.49% over Fashion-MINIST and 88.30% over CIFAR-10 on average.

**Attack ability.** In our work, we assume the attacker could obtain most of the essential information about the target model, such as DNN structure, training raw data $D$, hyperparameters $\theta$ used for model training, and the classification output, but is incapable of probing the internal variables of the network to gain access to the watermark image or the watermarked input in the continuous workflow. This assumption is reasonable taking account of the modern protected computing systems and it is typical in domains, for example, biometric and digital watermarking applications. We evaluate our defense

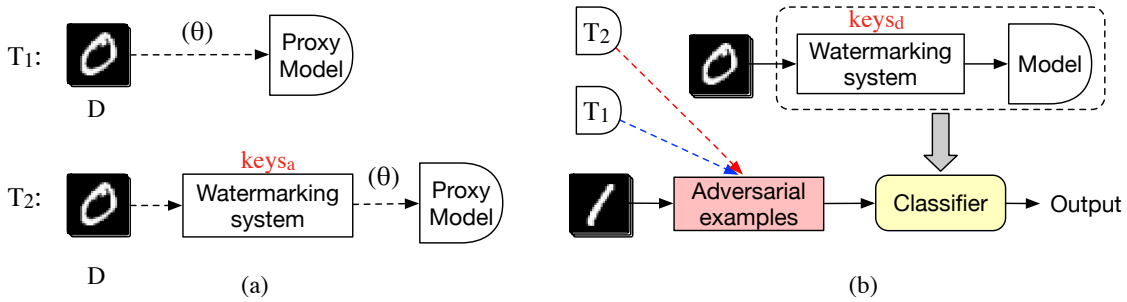



Figure 4.2: $(a)$: Two threat models (zero knowledge threat model $T_1$ and partial knowledge threat model $T_2$); $(b)$: the evaluation workflow where the defense model is trained on watermarked data and different attack models generate adversarial examples to attack the defender.

Table 4.1: Classification accuracy (%) of defense model against attack $T_1$

| Watermarking | MNIST | | | | Fashion-MNIST | | | |
|---|---|---|---|---|---|---|---|---|
| | FGSM | PGD | DeepFool | CW-L2 | FGSM | PGD | DeepFool | CW-L2 |
| – | 4.8 | 0.6 | 1.0 | 0.6 | 6.4 | 5.2 | 6.8 | 6.3 |
| DFT | **60.7** | **48.3** | 94.5 | 92.3 | **47.9** | **40.6** | 79.5 | 81.4 |
| DWT | 58.9 | 28.5 | **95.6** | **93.3** | 21.7 | 14.4 | 82.6 | 81.9 |
| SVD | 56.6 | 13.7 | 94.7 | 82.2 | 24.5 | 18.8 | 80.0 | 74.7 |
| DWT_SVD | 53.2 | 26.3 | 93.6 | 90.4 | 28.7 | 26.6 | 82.0 | 81.9 |
| DWT_DCT_SVD | 49.3 | 23.6 | 94.1 | 91.9 | 37.0 | 37.4 | **84.3** | **82.0** |

strategy against the attackers with different amounts of knowledge about our defense method. As watermarking color images is a challenge compared to gray-scale images, we place color watermarking defense in a separate experimental section.

#### 4.1.4.2 Evaluation of Watermarking-based Defense

**Defense against Zero Knowledge Attack $T_1$.** In the first scenario, we consider a very straightforward attack type $T_1$ (Figure 4.2$(a)$), where the attacker has zero knowledge about defense. Intuitively, the attacker utilizes the obtained training set $D$ and hyperparameters $\theta$ from the observation, and trains a surrogate DNN model within their knowledge for the same image recognition task as the target model does.

The whole evaluation process is illustrated in Figure 4.2$(b)$, where the attacker generates adversarial examples from the trained threat model $T_1$ to attack the defense classifier, while the classifier contains a watermarking system and a well-trained DNN model, i.e., the adversarial examples will be first processed with the watermarking

Table 4.2: Classification accuracy (%) of defense model against attack $T_2$

| Watermarking | MNIST | | | | Fashion-MNIST | | | |
|---|---|---|---|---|---|---|---|---|
| | FGSM | PGD | DeepFool | CW-L2 | FGSM | PGD | DeepFool | CW-L2 |
| – | 4.8 | 0.6 | 1.0 | 0.6 | 6.4 | 5.2 | 6.8 | 6.3 |
| DFT | **62.7** | **44.2** | 90.1 | 84.6 | **48.3** | **39.3** | 78.3 | 71.5 |
| DWT | 51.9 | 29.5 | 91.7 | 80.9 | 19.7 | 12.8 | 80.4 | 70.9 |
| SVD | 42.7 | 13.3 | 94.6 | 86.8 | 22.9 | 16.8 | 81.2 | 75.2 |
| DWT_SVD | 49.4 | 22.9 | 92.1 | 88.9 | 26.2 | 23.5 | 78.8 | 70.1 |
| DWT_DCT_SVD | 43.1 | 27.6 | **95.4** | **93.4** | 34.0 | 36.9 | **83.4** | **76.4** |

system before fed for classification. In this part of experiments, we compare the results of our method encoded with five different watermarking embedders including DFT, DWT, SVD, DWT_SVD and DWT_DCT_SVD with a standard DNN classifier without watermarking defense, trained on the original input image set, whose results also serve as the baseline. We craft 1,000 adversarial examples, and compute the test accuracy of our defense model on these generated adversarial examples. The results are shown in Table 4.1. We can observe that:

- DFT can effectively decrease the classification error of adversarial examples for all types of considered attacks. Specifically, it enhances the model's test accuracy on adversarial examples from 0.6–4.8% to 48.3–92.3% on MNIST. In the case of Fashion-MNIST, the results have slightly declined, but we can still see an improvement against a variety of adversarial examples (35.4–75.1% increase). For FGSM and PGD, DFT-based defense significantly outperforms other methods.

- Our method obtains outstanding results on DeepFool and CW-L2 for all employed watermarking algorithms. On MNIST, the test accuracy increases up to 95.6% on DeepFool and 93.3% on CW-L2. On Fashion-MNIST, the best defense result reaches 84.3% test accuracy on DeepFool and 82.0% on CW-L2. By contrast, the experimental results on FGSM and PDG are not as good as that on DeepFool and CW-L2.

**Defense against Partial Knowledge Attack $T_2$.** In the second attack scenario, we focus on a stronger attack, where we enable the attacker to partially learn about our defensive strategy including watermarking embedder devised in the defense method, but not the watermark keys embed to the input data. We define the second threat model

as $T_2$ (as presented in Figure 4.2($a$)). Specifically, to substitute the watermark images $keys_d$ used in defense model, the attacker randomly picks surrogate images $keys_a$ from the large image database to watermark the input, and trains their surrogate model for adversarial example generation. The evaluation process for our defense model against threat model $T_2$ is depicted in Figure 4.2 as well, where the attacker trains the network on the watermarked image set embedded with a different watermark set $keys_a$ from $keys_d$ through the same watermarking system as the defense framework. Likewise, five watermarking algorithms are performed and compared on 1,000 adversarial examples for evaluation. The experimental results are shown in Table 4.2, where we can see that:

- DFT can potentially improve the robustness of the model against FGSM and PGD better than other watermarking algorithms. The results show a 43.6–57.9% accuracy increase on MNIST. For more complex Fashion-MNIST, it shares the same observed tendency, but with a slight drop-off.

- Similar to Attack I, our defense achieves very promising results against DeepFool and CW-L2 for all the watermarking algorithms. On MNIST, it can decrease the classification error on adversarial examples from 99.0–99.4% to 4.6–19.1%. For Fashion-MNIST, the error rate is reduced from 93.2% to 16.6% on DeepFool and from 93.7% to 23.6% on CW-L2.

**Discussion.** The experimental results and analysis demonstrate that watermarking-based defense can effectively enhance DNN robustness against adversarial attacks, even the attacker may have different knowledge about the targeted system. In particular, our method achieves high performance against the sophisticated attacks, e.g., DeepFool and CW-L2. Unlike other attacks, these attacks are optimally generated through iterative optimization, which may easily get overfitted to the model parameters and training dataset and result in weak generalization. As a result, the delicate changes brought by the watermark have a significant impact on them.

On the other hand, the defense efficacy on FGSM and PGD attacks underperform DeepFool and CW-L2, since such attacks have lower variance and better transferability to the learning models. Also, the information discrepancy caused by the subtle watermark does not necessarily induce sufficient patterns to destroy the specific structure of adversarial perturbations. To address this limitation, we might need to either introduce more potentially secret information to enlarge the knowledge gap between the defender and the attacker, or leverage additional techniques (e.g., adversarial training against

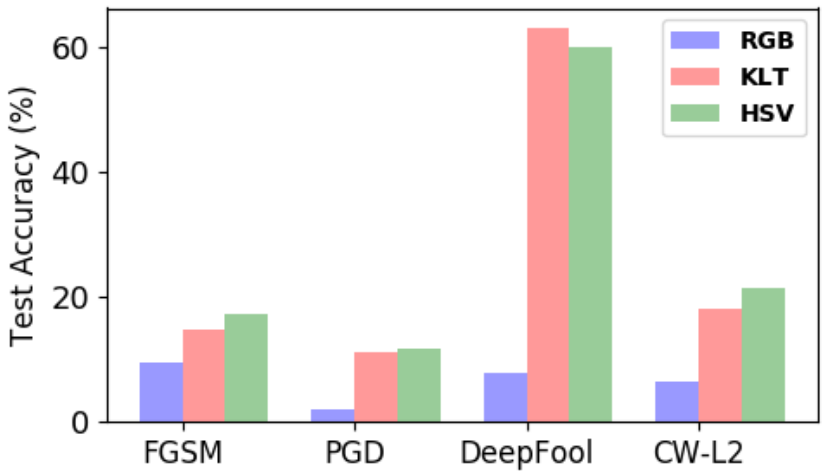



Figure 4.3: Accuracy on different color transformations.

single-step attacks) to further facilitate our watermarking-based defense. We leave it as our future work.

#### 4.1.4.3 Watermarking Defense on Color Images

Watermarking RGB images is not as straightforward as that on gray-scale images, due to the fact that these three components are inter-correlated. To put it into perspective, we initially exploit DFT to watermark RGB component for the color images from CIFAR-10 [166], train a DNN model with and without color watermark embedding process on regular data. The test accuracy decreases from 88.3% to 38.1%. Such a naive watermarking method almost generates a denial of service for classification, let alone be used as defense against adversarial attacks. In our work, we attempt KLT and HSV transform to decorrelate RGB information of the color images in our defense, respectively. We preliminarily assess the effectiveness of these color image watermarking methods on the adversarial examples generated by different adversarial attacks on CIFAR-10. The results are showed in Figure 4.3.

From Figure 4.3, we can see that the embedding process using transformations indeed helps to improve the color watermarking quality, which outperforms the direct watermarking on RGB space. Regarding the adversarial examples, our defense using KLT and HSV improves the classification performance by different degrees against different attacks, especially that the test accuracy achieves 63.2% and 60.0% against DeepFool, which is better than other attacks. The same observations can be found in Table 4.1 and Table 4.2, and the defense efficacy difference among FGSM, PGD, DeepFool and CW-L2

has been well explained in Section 4.1.4.2. Considering that color image watermarking is still an open issue with challenges and difficulties (e.g., color representations) [161], we'd like to gain further insight to enhance the color watermarking in our defense strategy in the future work.

#### 4.1.4.4 Evaluation on Different Watermark Patterns

Due to the large amount of possible patterns introduced by watermarking system, it is worth analyzing the different types of watermark images that work for our defense model precisely. In this section, we thus validate the effectiveness and significance of watermark image patterns in building a defense model. In our experiments, we limit the freedom of watermark image choice to be one class of images from ImageNet, and randomly choose different watermark images of one specific class to be watermark keys. We test the watermarking system encoded with six image patterns respectively (i.e., tobacco shop, tractor, pug, vase, gorilla, valley) to evaluate the performance of the defense model against adversarial attacks. As illustrated in prior experiments, DFT performs better than other four transformation algorithms applied in our defense strategy on average; therefore, we evaluate the effectiveness of different watermark images using DFT as the embedder of watermarking system. We report the results with respect to the classification accuracy on MNIST in Table 4.3.

As revealed from the results, the defense performances slightly vary in different classes of watermark images where some image patterns could outperform others against one adversarial attack while underperform a bit against another attack (e.g., Tobacco shop achieves 95.3% accuracy on CW-L2 while 89.9% accuracy on DeepFool). Overall, watermarking-based defense is not strictly sensitive to the specific patterns introduced by the watermark images, and is able to reach reasonable performance under a random image choice. Recall that, the watermarking is also easy for implementation without many additional computations and extra training. These properties make our defense model convenient and feasible in practical use.

#### 4.1.4.5 Comparisons with Other Methods

In this set of experiments, we examine the effectiveness of our defense model against the adversarial attacks by comparisons with other related state-of-the-art defense methods, including: (1) resizing [149], (2) padding [149], (3) resizing+padding [149], (4) bit-depth reduction [34, 35], (5) JPEG compression [35, 169], and (6) Guassian noise [32]. More specifically, resizing strategy resizes the original input images into a new image with

Table 4.3: Accuracy (%) over different watermark classes

| Watermark Class | FGSM | PGD | DeepFool | CW-L2 |
|---|---|---|---|---|
| **Tobacco shop** | 53.7 | 44.7 | 89.9 | 95.3 |
| **Tractor** | 54.0 | 36.4 | 91.8 | 93.7 |
| **Pug** | 52.4 | 41.4 | 92.0 | 94.3 |
| **Vase** | 47.8 | 27.7 | 91.6 | 94.9 |
| **Gorilla** | 49.9 | 31.6 | 88.2 | 95.1 |
| **Valley** | 50.7 | 36.5 | 93.3 | 82.9 |

Table 4.4: Accuracy (%) over different defense models

| Defense Methods | FGSM | PGD | DeepFool | CW-L2 |
|---|---|---|---|---|
| – | 4.8 | 0.6 | 1.0 | 0.6 |
| **Resizing** | 14.7 | 2.0 | 92.8 | 92.1 |
| **Padding** | 13.4 | 0.9 | 93.4 | 92.6 |
| **Resizing+Padding** | 14.5 | 0.7 | 90.4 | 91.9 |
| **Bit-depth reduction** | 7.9 | 0.6 | 85.0 | 72.4 |
| **JPEG compression** | 11.1 | 0.7 | 93.3 | 82.9 |
| **Guassian Noise** | 10.5 | 0.6 | 92.2 | 86.0 |
| **Watermarking** | **60.7** | **48.3** | **94.8** | **93.6** |

random size. As discussed in [149], the difference between the original and new sizes should be within a reasonably small range to avoid performance drop-off. Considering the image set used in our experiments is of $28 \times 28$ size, we set the new size for each image as $30 \times 30$. Padding pads zeros around the resized images for each side. For resizing+padding, we first resize the images to $29 \times 29$, and then pads zero pixels on the left and bottom to obtain $30 \times 30$ images. Bit-depth reduction performs a type of quantization to squeeze image features that can possibly remove small adversarial perturbations; we reduce the images to 4 bits in our experiments. JPEG compression uses the similar way to disrupt adversarial perturbations; we follow the work [35] to perform compression at quality level 75 (out of 100). Guassian noise $\mathcal{N}(0, 1)$ is added to the image data to introduce randomization to the target model. The experimental results on MNIST using DFT watermarking algorithm are reported in Table 4.3.

From Table 4.4, we observe that different image transformations can mitigate the adversarial effects for iterative attacks like DeepFool and CW-L2 significantly, the reason behind which has been well analyzed in Section 4.1.4.2. As for FGSM and PGD indicating stronger transferability, these alternative image transformation methods suffer from a

drastic drop-off, i.e., the best classification accuracy can only reaches to 14.7% and 2.0% for FGSM and PGD respectively. By contrast, our watermarking-based defense can well preserve the unique structure and patterns through the designed watermarking procedure and enforce a distinctive discrepancy between the defense model and the surrogate model, and thus outperforms other related defense methods.

### 4.1.5 Summary

In this work, we leverage the watermarking transformation to introduce secret watermark into our defense model, and thus leading to knowledge gap to impede attackers. Accordingly, we impair the capability of the attacker to customize the knowledge of the defense model and avoids the possible adaptive attack behaviors in real use. We evaluate the effectiveness of our strategy under two attacking scenarios where the attacker is enabled with different knowledge of the target model. Moreover, we compare our method with other related work and demonstrate the promising potential of using digital watermarking as a kind of randomizations to improve the robustness of the defense model. However, the small capability of watermark embedded to the image limits its performance in defending against some powerful adversarial attacks. As future work, we aim to investigate other potential methods to enlarger the amount of secret information that we can introduce to the defense model and examine its behavior on more complex datasets.

# Chapter 5
# The Ugly: There is No Free Lunch

In the deep learning community, there constantly arise new and advanced researches on adversarial to resolve vulnerability issue of the model. Despite the great efforts, those public defending strategies have been verified to be compromised by attackers, and currently, there is no security guarantee concerning the adversarial attacks on deep learning systems. The study results naturally lead us to give more careful considerations on the question: Is there a silver bullet for this problem? As far as we know, absolute security is not possible by the state-of-the-art [76], there still exit lots of new challenges ahead facing with the threat that adversaries pose to us. But we have to admit that studying such phenomenon not only provides us with measurements to test the security of machine learning models, but also helps with better understanding the behaviors of adversaries. More importantly, the principle underlying this problem is worth for our more exploration.

## 5.1 Discussion

In this thesis, we present three work from the perspective of improving the DNN robustness against adversarial attacks [170] and applying attack techniques to social good scenarios [138]. Although the designed methods show their effectiveness in different evaluation environment, their generality and stability still suffer limitations. This is due to the reason that resolving the security issue of adversarial machine learning is still an intractable problem. With the continuous arms race between attackers and defenders, the existence of universally robust defending strategy is by far an open question to us. In this regard, the proposed defense methods face the challenge in protecting model or user privacy against more developed attack strategies. In this section, we will elaborate these limitations with more details and discuss their future work correspondingly.

### 5.1.1 Social media privacy protection using adversarial attacks

In our first work, we investigate the potential of leveraging adversarial attacks for social good, i.e., social media data privacy. As one of our future work, Adv4SG should be an easy-to-use service provided on users' social media client side, so that its privacy protection functionality would be realized in practice. For example, Adv4SG can be developed as an API that is integrated into social media posting and editing systems to allow users to choose the adversarial text according to their provided attribute and text content. However, we also face some challenges and limitations on the applicability of our proposed method. First, we require a large amount of data with labels while training the NLP model to generate adversarial examples, while it is not realistic in the real world. Especially for social media, user data usually suffer from label shortage. To better obfuscate the private attribute, we may need to first recognize the target labels. In order to resolve that, we may resort to semi-supervised learning or few-shot learning strategies which release the burden on data labels. Also, in our experiments, we test the performance of our protection against different inference attack models. Though Adv4SG has been investigated and validated in terms of transferability of adversarial examples. The attacker in real world can always evolve with more sophisticated behaviors that the defender is not able to predict that we need to consider more cases while defending against those more advanced inference threat models.

### 5.1.2 Attribute-obfuscating attack on graph for social good

Considering the challenges in our first study, we propose an attribute-obfuscating attack on graph neural networks for social networks' user privacy. The benefits of this work can be summarized as two points: (1) based on the complex heterogeneous environment of social network, more and more attackers tend to model the diverse user data as graph for downstream tasks. To protest such attacks, we aim to design more powerful protections to guarantee the security of user's privacy. (2) lacking of data labels is a main challenge for lots of deep learning models, graph neural networks such as GCN can achieve superior performance under the semi-supervised setting where only a small amount of training data has labels.

As our goal is developing a practical attribute obfuscation method to protect users' private attribute in social network, there still exist several challenges for us to resolve. Though AttrOBF has been validated to be transferable to these GNNs, the attackers could take advantage of more advanced and robust GNN models (e.g., adversarial training

via latent perturbation) to infer attributes and thus deteriorate AttrOBF. To solve it, we need to observe and analyze more factors that can improve the transferability of our graph-based adversarial attack. To design more sophisticated and stronger attack on socal graph is also one possible way. The other limitation is that, from our current experimental results, the number of training labels we need to obfuscate is a little bit too much which consequently would affect the practicality of our method, especially for those multi-class obfuscation task.

### 5.1.3 Watermarking-based defense against DNNs

In our third study, the designed watermarking-based defense yields a knowledge gap advantage to protest gradient-based adversarial attacks by imposing a watermarking module to DNNs. The information gap created through watermarking effectively weakens the capability of attackers. In this work, we show the validity of digital watermark in hiding information to the defense model and thereby improving its robustness. The results validate our proposed idea against adversarial example attacks from limiting the attacker's knowledge of the defense model, especially for the optimized adversarial perturbations. On the other hand, our method underperforms on some adversarial examples, such as PGD. Such attacking models tend to have lower variance and better transferability to the protected models, while the information discrepancy caused by the subtle watermark does not always induce sufficient model distortions to fail the adversarial perturbations. To address this limitation, we might leverage additional techniques to further facilitate the model protection.

## 5.2 No Free Lunch Theorem in Adversarial Setting

In adversarial learning community, the attacking and defending strategies emerge constantly, but the fact is this mini-max game seems to fall into the endless loop [76]. Under this dilemma, some researchers try to jump out of the box and think of this issue from different perspectives. For example, in [171], researchers point out that current machine learning practitioners primarily rely on testing, but it is not sufficient to provide security guarantees as the attacker can send samples that differ from those used for testing purpose, i.e., adversarial examples. Besides, they mention that we should verify the model rather than test it.

On the other hand, it is worth considering the possibility that there does not exist fully

robust and accurate machine learning models. In traditional machine learning setting, there exists such "no free lunch" theorem [172] states that all machine learning algorithms are equally effective across all possible prediction problems. The "no free lunch" theorem was first proposed in late 1990s [173] where claims no optimization algorithm is any better than any other optimization algorithm, on average. Then it is applied to machine learning. When it comes to the adversarial setting, researchers are keen to know whether "no free lunch" theorem can be extended to it as well. If the answer is yes, then the average performance of the model is taken on all possible datasets including those small malicious perturbations, and it may indicate that those perturbations from the attackers should be ignored [171, 174]. In addition, researchers in [175] study the "no free lunch" theorem towards adversarial robustness of simple machine learning models and state that the attack models currently being considered in the literature may be too lax and implausible. One thing we are sure about this important open theoretical question is that, once it is resolved, it can extremely redefine the game. In a nutshell, with the challenges ahead, we still need lots of more efforts to move forward.

## 5.3 Future Work

In our work that utilizing adversaries for social good, we show the applicability of adversarial machine learning for human benefits. But as we discussed, our method cannot guarantee the 100% protection against inference attacks. The attacker in real world can always evolve with more sophisticated behaviors that the defender is not able to predict. Especially, the continuous arms race between adversarial attacks and defenses stimulate the development of defense capability for inference model to evade these generated adversarial examples. For instance, in NLP field, the attackers could take advantage of more advanced and robust learning models (e.g., spelling checking, and graph learning) to infer attributes and thus deteriorate Adv4SG. In this regard, we need to correspondingly upgrade the generality of our protection against a variety of inference attacks. Meanwhile, we should also pay attention to the design of more powerful adversarial attacks as we can consider them as our attribute obfuscation technique to protect users' privacy.

For the practicability of our methods, we can further consider the details and requirements on put them into use with the real-world social media platforms. For instance, we can design such build-in interfaces to help users realize the attribute obfuscating functions. On the other hand, the current performance of the proposed framework

expects for better optimization from a practical point of view. Take our second work as an example, it is unrealistic for us to expect many of users are willing to shift their truth attributes, reducing the required graph perturbation to an ideal small amount is an important step for us to improve, we leave that as our future work. In addition, the great potential that adversarial machine learning showed in protecting data privacy in social media, we are also inspired to search for more benefits of it in other domains, such as software security. In terms of our proposed watermarking-based defense, we consider assembling our scheme with other defense techniques designed from different security perspective to more comprehensively enhance the robustness of DNNs, e.g., adversarial training against single-step attacks. What's more, the vulnerability issue on adversarial has been proved to exist in different kinds of learning models other than DNNs. For instance, the well-trained graph neural networks can easily be attacked by carefully designed adversarial modifications. Therefore, we are looking forward to putting more efforts on studying the robustness of diverse structured models against such adversarial attacks in our future research.

# Chapter 6
# Conclusion

In this thesis, we present three of our work to explore the robustness and applicability of adversarial machine learning. In the first and the second work, we investigate adversary for social good, and cast attribute privacy protection problem on social media as an adversarial attack formulation problem to defend against attribute inference attacks. In particular, in our first work, we focus on text data in our problem and propose a text-space adversarial attack Adv4SG under the black-box setting, where the attack constraints are first defined; guided by them, a sequence of plausible perturbations are automatically performed to generate the adversarial texts using semantically and visually similar word candidates, which are regulated by a reformed population-based optimization algorithm. We conduct comprehensive experimental studies on real-world social media datasets to evaluate the performance of Adv4SG, which validate its effectiveness and efficiency against attribute inference attacks. Despite the challenges and limitations, we believe that our work unveils novel insight of turning adversarial attacks in machine learning into defense strategies and implies the great potential on the applicability of adversarial attacks for attribute obfuscation and privacy protection in practice.

Followed by the first work, we focus on more complicated data structure of social media and aim to design user privacy protections against more sophisticated inference attacks on social graphs in our last work. Specifically, we investigate adversary for social good, and cast attribute privacy protection problem on social networks as a graph adversarial attack formulation problem to defend against GNN-based attribute inference attacks. We design a black-box attribute-obfuscating attack AttrOBF, where a linearized two-layer GCN is used as a surrogate model to perform our attack. With the help of this surrogate model, a closed form of model weights is obtained to transform the intractable bi-level optimization for AttrOBF into single-level. To address non-differentiable attribute obfuscating operation optimization issue, we introduce the Gumbel estimator to generate

continuous differentiable approximation. It enables gradient-based methods to search for the optimal training attribute values to modify through back-propagation. In this work, we conduct extensive experimental studies on real-world social network datasets to evaluate the performance of AttrOBF, which validate its effectiveness against GNN-based attribute inference attacks. Despite the limitation, our work innovately propose to take advantage of the vulnerability of GNNs toward adversarial attacks for social good, and we believe that our work has implications on the applicability of adversarial attacks for attribute obfuscation and privacy protection in practice.

Finally, we study the robustness of the DNN model against adversarial examples, and propose a watermarking-based defense mechanism against adversarial attacks. We creatively impose a secret watermarking system into the DNN model to yield a knowledge gap advantage over the attackers. The experimental results demonstrate that our defense can effectively enhance the robustness of the DNN classifier against adversarial attacks even the attacker may have different knowledge about the targeted learning system, and we also prove that watermark is a good choice to introduce randomization of the defense model. Followed by comprehensive evaluations, we show that the information gap created through watermarking effectively weakens the capability of attackers. In addition, the results illustrate a promising potential of our proposed idea against adversarial attacks from limiting the attacker's knowledge of the defense model, especially for the optimized adversarial perturbations. On the other hand, our defense underperforms on some type of adversarial examples, such as PGD. It is because that the model disturbance introduced by watermark is very subtle, the information discrepancy caused by it might not be sufficient to destroy the specific adversarial perturbations, especially those with lower variance and higher transferability. We disscuss the possible solution to it. By upgraing the randomness capacity of our method, such as introducing more potentially secret information to enlarge the knowledge gap between the defender and the attacker, we can potentially improve the generalization of our protections.

# Appendix
# Publication List

- **Xiaoting Li**, Xiao Liu, Lingwei Chen, Rupesh Prajapati, and Dinghao Wu. AlphaProg: Reinforcement Generation of Valid Programs for Compiler Fuzzing. *In Proceedings of 34th Annual Conference on Innovative Applications of Artificial Intelligence*, 2022. (IAAI '22)

- Lingwei Chen, **Xiaoting Li**, and Dinghao Wu. Adversarially Reprogramming Pretrained Neural Networks for Data-limited and Cost-efficient Malware Detection. *In Proceedings of SIAM International Conference on Data Mining*, 2022. (SDM '22)

- **Xiaoting Li**, Lingwei Chen, Jinquan Zhang, James Larus, and Dinghao Wu. Watermarking-based Defense against Adversarial Attacks on Deep Neural Networks. *In Proceedings of International Joint Conference on Neural Networks*, 2021. (IJCNN '21)

- Qinkun Bao, Zihao Wang, **Xiaoting Li**, James Larus, and Dinghao Wu. Abacus: Precise Side-Channel Analysis. *In Proceedings of 43rd International Conference on Software Engineering*, 2021. (ICSE '21)

- **Xiaoting Li**, Lingwei Chen, and Dinghao Wu. Turning Attacks into Protection: Social Media Privacy Protection Using Adversarial Attacks. *In Proceedings of SIAM International Conference on Data Mining*, 2020. (SDM '20)

- Lingwei Chen, **Xiaoting Li**, and Dinghao Wu. Enhancing Robustness of Graph Convolutional Networks via Dropping Graph Connections. *In Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2020. (ECML-PKDD '20)

- Xiao Liu, **Xiaoting Li**, Rupesh Prajapati, and Dinghao Wu. DeepFuzz: Automatic Generation of Syntactically Correct C Programs for Fuzz Testing. *In Proceedings of 33th AAAI Conference on Artificial Intelligence*, 2019. (AAAI '19)

# Bibliography


[1] Goodfellow, I. J., J. Shlens, and C. Szegedy (2014) "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*.

[2] Jang, E., S. Gu, and B. Poole (2016) "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*.

[3] LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998) "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, **86**(11).

[4] Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012) "Imagenet classification with deep convolutional neural networks," in *NIPS*, pp. 1097–1105.

[5] Simonyan, K. and A. Zisserman (2014) "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*.

[6] Hinton, G., L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury (2012) "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, **29**.

[7] Saon, G., H.-K. J. Kuo, S. Rennie, and M. Picheny (2015) "The IBM 2015 English conversational telephone speech recognition system," *arXiv preprint arXiv:1505.05899*.

[8] Sutskever, I., O. Vinyals, and Q. V. Le (2014) "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, pp. 3104–3112.

[9] Andor, D., C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins (2016) "Globally normalized transition-based neural networks," *arXiv preprint arXiv:1603.06042*.

[10] Chen, M., Y. Hao, K. Hwang, L. Wang, and L. Wang (2017) "Disease prediction by machine learning over big data from healthcare communities," *IEEE Access*, **5**, pp. 8869–8879.

[11] Bojarski, M., D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba (2016) "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*.

[12] Rao, Q. and J. Frtunikj (2018) "Deep learning for self-driving cars: Chances and challenges," in *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*, pp. 35–38.

[13] Stilgoe, J. (2018) "Machine learning, social learning and the governance of self-driving cars," *Social Studies of Science*, **48**(1), pp. 25–56.

[14] Gavriluţ, D., M. Cimpoeşu, D. Anton, and L. Ciortuz (2009) "Malware detection using machine learning," in *2009 International Multiconference on Computer Science and Information Technology*, IEEE, pp. 735–741.

[15] Peiravian, N. and X. Zhu (2013) "Machine learning for android malware detection using permission and api calls," in *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, IEEE, pp. 300–305.

[16] Firdausi, I., A. Erwin, and A. S. Nugroho (2010) "Analysis of machine learning techniques used in behavior-based malware detection," in *2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies*, IEEE, pp. 201–203.

[17] Dorigo, M. and U. Schnepf (1993) "Genetics-based machine learning and behavior-based robotics: a new synthesis," *IEEE Transactions on Systems, Man, and Cybernetics*, **23**(1), pp. 141–154.

[18] Siau, K. and W. Wang (2018) "Building trust in artificial intelligence, machine learning, and robotics," *Cutter Business Technology Journal*, **31**(2), pp. 47–53.

[19] Chakraborty, A., M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay (2018) "Adversarial attacks and defences: A survey," *arXiv preprint arXiv:1810.00069*.

[20] Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus (2013) "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*.

[21] Wang, X., J. Li, X. Kuang, Y.-a. Tan, and J. Li (2019) "The security of machine learning in an adversarial setting: A survey," *Journal of Parallel and Distributed Computing*, **130**, pp. 12–23.

[22] Evtimov, I., K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song (2017) "Robust physical-world attacks on deep learning models," *arXiv preprint arXiv:1707.08945*, **1**, p. 1.

[23] Kurakin, A., I. Goodfellow, and S. Bengio (2016) "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*.

[24] Dong, Y., F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li (2018) "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9185–9193.

[25] Tramèr, F., A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel (2017) "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*.

[26] Moosavi-Dezfooli, S.-M., A. Fawzi, and P. Frossard (2016) "Deepfool: a simple and accurate method to fool deep neural networks," in *CVPR*, pp. 2574–2582.

[27] Carlini, N. and D. Wagner (2017) "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy*, IEEE, pp. 39–57.

[28] Cisse, M., P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier (2017) "Parseval networks: Improving robustness to adversarial examples," in *ICML*, pp. 854–863.

[29] Papernot, N., P. McDaniel, X. Wu, S. Jha, and A. Swami (2016) "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy*, IEEE, pp. 582–597.

[30] He, K., X. Zhang, S. Ren, and J. Sun (2016) "Deep residual learning for image recognition," in *CVPR*, pp. 770–778.

[31] DeVries, T. and G. W. Taylor (2017) "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*.

[32] Taran, O., S. Rezaeifar, and S. Voloshynovskiy (2018) "Bridging machine learning and cryptography in defence against adversarial attacks," in *European Conference on Computer Vision*, Springer, pp. 267–279.

[33] Lu, J., H. Sibai, E. Fabry, and D. Forsyth (2017) "No need to worry about adversarial examples in object detection in autonomous vehicles," *arXiv preprint arXiv:1707.03501*.

[34] Xu, W., D. Evans, and Y. Qi (2017) "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*.

[35] Guo, C., M. Rana, M. Cisse, and L. van der Maaten (2017) "Countering adversarial images using input transformations," *arXiv preprint arXiv:1711.00117*.

[36] Lu, J., T. Issaranon, and D. Forsyth (2017) "Safetynet: Detecting and rejecting adversarial examples robustly," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 446–454.

[37] Metzen, J. H., T. Genewein, V. Fischer, and B. Bischoff (2017) "On detecting adversarial perturbations," *arXiv preprint arXiv:1702.04267*.

[38] Bhagoji, A. N., D. Cullina, C. Sitawarin, and P. Mittal (2018) "Enhancing robustness of machine learning systems via data transformations," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, IEEE, pp. 1–5.

[39] Chen, P.-Y., H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh (2017) "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26.

[40] Su, J., D. V. Vargas, and K. Sakurai (2019) "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*.

[41] Jia, J. and N. Z. Gong (2018) "Attriguard: A practical defense against attribute inference attacks via adversarial machine learning," in *27th USENIX Security Symposium (USENIX Security 18)*, pp. 513–529.

[42] Yu, S., Y. Vorobeychik, and S. Alfeld (2018) "Adversarial classification on social networks," in *AAMAS*, pp. 211–219.

[43] Beigi, G., K. Shu, Y. Zhang, and H. Liu (2018) "Securing social media user data: An adversarial approach," in *Proceedings of the 29th on Hypertext and Social Media*, pp. 165–173.

[44] Gong, N. Z. and B. Liu (2018) "Attribute inference attacks in online social networks," *TOPS*, **21**(1), p. 3.

[45] Zhang, Y., M. Humbert, T. Rahman, J. Pang, and M. Backes (2018) "Tagvisor: A privacy advisor for sharing hashtags," in *WWW*.

[46] Jia, J., B. Wang, L. Zhang, and N. Z. Gong (2017) "AttriInfer: Inferring user attributes in online social networks using markov random fields," in *WWW*, pp. 1561–1569.

[47] Oh, S. J., M. Fritz, and B. Schiele (2017) "Adversarial image perturbation for privacy protection a game theory perspective," in *ICCV*, pp. 1491–1500.

[48] Shokri, R., M. Stronati, C. Song, and V. Shmatikov (2017) "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy*, IEEE, pp. 3–18.

[49] Nasr, M., R. Shokri, and A. Houmansadr (2018) "Machine learning with membership privacy using adversarial regularization," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, ACM, pp. 634–646.

[50] Salem, A., Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes (2018) "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv preprint arXiv:1806.01246*.

[51] Caliskan-Islam, A., R. Harang, A. Liu, A. Narayanan, C. Voss, F. Yamaguchi, and R. Greenstadt (2015) "De-anonymizing programmers via code stylometry," in *USENIX Security*.

[52] Li, C., S. Wang, Y. Wang, P. Yu, Y. Liang, Y. Liu, and Z. Li (2019) "Adversarial learning for weakly-supervised social network alignment," in *AAAI*, vol. 33, pp. 996–1003.

[53] Shokri, R., G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec (2012) "Protecting location privacy: optimal strategy against localization attacks," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pp. 617–627.

[54] Shokri, R. (2015) "Privacy games: Optimal user-centric data obfuscation," *Proceedings on Privacy Enhancing Technologies*, **2015**(2), pp. 299–315.

[55] Shokri, R., G. Theodorakopoulos, and C. Troncoso (2016) "Privacy games along location traces: A game-theoretic framework for optimizing location privacy," *ACM Transactions on Privacy and Security (TOPS)*, **19**(4), pp. 1–31.

[56] du Pin Calmon, F. and N. Fawaz (2012) "Privacy against statistical inference," in *2012 50th annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, pp. 1401–1408.

[57] Dwork, C., F. McSherry, K. Nissim, and A. Smith (2006) "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference*, Springer, pp. 265–284.

[58] Duchi, J. C., M. I. Jordan, and M. J. Wainwright (2013) "Local privacy and statistical minimax rates," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, IEEE, pp. 429–438.

[59] Erlingsson, Ú., V. Pihur, and A. Korolova (2014) "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *CCS*, pp. 1054–1067.

[60] Bassily, R. and A. Smith (2015) "Local, private, efficient protocols for succinct histograms," in *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, pp. 127–135.

[61] Wang, T., J. Blocki, N. Li, and S. Jha (2017) "Locally differentially private protocols for frequency estimation," in *USENIX Security*, pp. 729–745.

[62] Keswani, Y., H. Trivedi, P. Mehta, and P. Majumder (2016) "Author Masking through Translation." in *CLEF (Working Notes)*, pp. 890–894.

[63] Karadzhov, G., T. Mihaylova, Y. Kiprov, G. Georgiev, I. Koychev, and P. Nakov (2017) "The case for being average: A mediocrity approach to style masking and author obfuscation," in *International Conference of the Cross-Language Evaluation Forum for European Languages*.

[64] Shetty, R., B. Schiele, and M. Fritz (2018) "A4NT: author attribute anonymity by adversarial training of neural machine translation," in *Proceedings of the 27th USENIX Security Symposium (USENIX Security 18)*, pp. 1633–1650.

[65] Jia, J., A. Salem, M. Backes, Y. Zhang, and N. Z. Gong (2019) "MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples," in *CCS*, pp. 259–274.

[66] Potdar, V. M., S. Han, and E. Chang (2005) "A survey of digital image watermarking techniques," in *INDIN*, IEEE, pp. 709–716.

[67] Morgan-Lopez, A. A., A. E. Kim, R. F. Chew, and P. Ruddle (2017) "Predicting age groups of Twitter users based on language and metadata features," *PloS One*, **12**(8).

[68] Ikeda, K., G. Hattori, C. Ono, H. Asoh, and T. Higashino (2013) "Twitter user profiling based on text and community mining for market analysis," *Knowledge-Based Systems*, **51**, pp. 35–47.

[69] Makazhanov, A., D. Rafiei, and M. Waqar (2014) "Predicting political preference of Twitter users," *SNAM*.

[70] Ruder, S., P. Ghaffari, and J. G. Breslin (2016) "Character-level and multi-channel convolutional neural networks for large-scale authorship attribution," *arXiv preprint arXiv:1609.06686*.

[71] Alzantot, M., Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang (2018) "Generating natural language adversarial examples," *arXiv preprint arXiv:1804.07998*.

[72] Papernot, N., P. McDaniel, and I. Goodfellow (2016) "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*.

[73] Kipf, T. N. and M. Welling (2016) "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*.

[74] Zhang, M., L. Hu, C. Shi, and X. Wang (2020) "Adversarial Label-Flipping Attack and Defense for Graph Neural Networks," in *ICDM*, pp. 791–800.

[75] Averkiou, M. (2010) "Digital watermarking," *Academia. edu*.

[76] Li, G., P. Zhu, J. Li, Z. Yang, N. Cao, and Z. Chen (2018) "Security matters: A survey on adversarial machine learning," *arXiv preprint arXiv:1810.07339*.

[77] Papernot, N., P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami (2016) "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, pp. 372–387.

[78] Rozsa, A., E. M. Rudd, and T. E. Boult (2016) "Adversarial diversity and hard positive generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 25–32.

[79] Kim, M. and J. Leskovec (2012) "Multiplicative attribute graph model of real-world networks," *Internet Mathematics*, **8**(1-2), pp. 113–160.

[80] Liu, K. and E. Terzi (2008) "Towards identity anonymization on graphs," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 93–106.

[81] Zhou, B. and J. Pei (2008) "Preserving privacy in social networks against neighborhood attacks," in *2008 IEEE 24th International Conference on Data Engineering*, IEEE, pp. 506–515.

[82] Dwork, C. (2008) "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*, Springer, pp. 1–19.

[83] Yuan, M., L. Chen, and P. S. Yu (2010) "Personalized privacy protection in social networks," *Proceedings of the VLDB Endowment*, **4**(2), pp. 141–150.

[84] Andreou, A., O. Goga, and P. Loiseau (2017) "Identity vs. attribute disclosure risks for users with multiple social profiles," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pp. 163–170.

[85] Qian, J., X.-Y. Li, T. Jung, Y. Fan, Y. Wang, and S. Tang (2019) "Social network de-anonymization: More adversarial knowledge, more users re-identified?" *TOIT*, **19**(3), pp. 1–22.

[86] Zhou, F., R. Yin, K. Zhang, G. Trajcevski, T. Zhong, and J. Wu (2019) "Adversarial point-of-interest recommendation," in *The World Wide Web Conference*, pp. 3462–34618.

[87] Machanavajjhala, A., D. Kifer, J. Gehrke, and M. Venkitasubramaniam (2007) "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, **1**(1), pp. 3–es.

[88] Li, N., T. Li, and S. Venkatasubramanian (2007) "t-closeness: Privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd International Conference on Data Engineering*, IEEE, pp. 106–115.

[89] Puglisi, S., J. Parra-Arnau, J. Forné, and D. Rebollo-Monedero (2015) "On content-based recommendation and user privacy in social-tagging systems," *Computer Standards & Interfaces*, **41**, pp. 17–27.

[90] Rebollo-Monedero, D. and J. Forné (2010) "Optimized query forgery for private information retrieval," *IEEE Transactions on Information Theory*, **56**(9), pp. 4631–4642.

[91] Melis, L., C. Song, E. De Cristofaro, and V. Shmatikov (2019) "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy*, IEEE, pp. 691–706.

[92] Hitaj, B., G. Ateniese, and F. Perez-Cruz (2017) "Deep models under the GAN: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 603–618.

[93] Bagdasaryan, E., A. Veit, Y. Hua, D. Estrin, and V. Shmatikov (2020) "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948.

[94] Papernot, N., P. McDaniel, A. Swami, and R. Harang (2016) "Crafting adversarial input sequences for recurrent neural networks," in *MILCOM 2016-2016 IEEE Military Communications Conference*, IEEE, pp. 49–54.

[95] Ebrahimi, J., A. Rao, D. Lowd, and D. Dou (2017) "Hotflip: White-box adversarial examples for text classification," *arXiv preprint arXiv:1712.06751*.

[96] Samanta, S. and S. Mehta (2017) "Towards crafting text adversarial samples," *arXiv preprint arXiv:1707.02812*.

[97] Belinkov, Y. and Y. Bisk (2017) "Synthetic and natural noise both break neural machine translation," *arXiv preprint arXiv:1711.02173*.

[98] Gao, J., J. Lanchantin, M. L. Soffa, and Y. Qi (2018) "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *2018 IEEE Security and Privacy Workshops (SPW)*, IEEE, pp. 50–56.

[99] Hosseini, H., S. Kannan, B. Zhang, and R. Poovendran (2017) "Deceiving google's perspective api built for detecting toxic comments," *arXiv preprint arXiv:1702.08138*.

[100] Iyyer, M., J. Wieting, K. Gimpel, and L. Zettlemoyer (2018) "Adversarial example generation with syntactically controlled paraphrase networks," *arXiv preprint arXiv:1804.06059*.

[101] Ribeiro, M. T., S. Singh, and C. Guestrin (2018) "Semantically equivalent adversarial rules for debugging nlp models," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 856–865.

[102] Li, J., S. Ji, T. Du, B. Li, and T. Wang (2018) "Textbugger: Generating adversarial text against real-world applications," *arXiv preprint arXiv:1812.05271*.

[103] Kuleshov, V., S. Thakoor, T. Lau, and S. Ermon (2018) "Adversarial examples for natural language classification problems," .

[104] Abu-El-Haija, S., B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, and A. Galstyan (2019) "MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing," in *International Conference on Machine Learning*, pp. 21–29.

[105] Ying, R., R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec (2018) "Graph convolutional neural networks for web-scale recommender systems," in *SIGKDD*, pp. 974–983.

[106] Liu, S., L. Chen, H. Dong, Z. Wang, D. Wu, and Z. Huang (2019) "Higher-order Weighted Graph Convolutional Networks," *arXiv preprint arXiv:1911.04129*.

[107] Chen, J., T. Ma, and C. Xiao (2018) "Fastgcn: fast learning with graph convolutional networks via importance sampling," *arXiv preprint arXiv:1801.10247*.

[108] Cai, Z., Z. He, X. Guan, and Y. Li (2016) "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, **15**(4), pp. 577–590.

[109] He, Z., Z. Cai, and J. Yu (2017) "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, **67**(1), pp. 665–673.

[110] Dai, H., H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song (2018) "Adversarial attack on graph structured data," *arXiv preprint arXiv:1806.02371*.

[111] Wu, H., C. Wang, Y. Tyshetskiy, A. Docherty, K. Lu, and L. Zhu (2019) "Adversarial examples for graph data: Deep insights into attack and defense," in *IJCAI*, pp. 4816–4823.

[112] Xu, K., H. Chen, S. Liu, P.-Y. Chen, T.-W. Weng, M. Hong, and X. Lin (2019) "Topology attack and defense for graph neural networks: An optimization perspective," *arXiv preprint arXiv:1906.04214*.

[113] Zügner, D., A. Akbarnejad, and S. Günnemann (2018) "Adversarial attacks on neural networks for graph data," in *SIGKDD*, pp. 2847–2856.

[114] Zügner, D. and S. Günnemann (2019) "Adversarial attacks on graph neural networks via meta learning," *arXiv preprint arXiv:1902.08412*.

[115] Bojchevski, A. and S. Günnemann (2019) "Adversarial attacks on node embeddings via graph poisoning," in *International Conference on Machine Learning*, PMLR, pp. 695–704.

[116] Kumar, C., R. Ryan, and M. Shao (2020) "Adversary for Social Good: Protecting Familial Privacy through Joint Adversarial Attacks." in *AAAI*.

[117] Ye, Y., S. Hou, Y. Fan, Y. Qian, Y. Zhang, S. Sun, Q. Peng, and K. Laparo (2020) "$\alpha$-Satellite: An AI-driven System and Benchmark Datasets for Hierarchical Community-level Risk Assessment to Help Combat COVID-19," *arXiv preprint arXiv:2003.12232*.

[118] Confessore, N. (2018) "Cambridge Analytica and Facebook: The Scandal and the Fallout So Far," *https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html*.

[119] Chen, L., Y. Ye, and T. Bourlai (2017) "Adversarial machine learning in malware detection: Arms race between evasion attack and defense," in *2017 European Intelligence and Security Informatics Conference (EISIC)*, IEEE, pp. 99–106.

[120] Pierazzi, F., F. Pendlebury, J. Cortellazzi, and L. Cavallaro (2019) "Intriguing Properties of Adversarial ML Attacks in the Problem Space," *arXiv preprint arXiv:1911.02142*.

[121] Papernot, N., P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami (2017) "Practical black-box attacks against machine learning," in *AsiaCCS*, pp. 506–519.

[122] Quiring, E., A. Maier, and K. Rieck (2019) "Misleading authorship attribution of source code using adversarial learning," in *USENIX Security 19*.

[123] Kolosnjaji, B., A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, and F. Roli (2018) "Adversarial malware binaries: Evading deep learning for malware detection in executables," in *2018 26th European Signal Processing Conference (EUSIPCO)*, IEEE, pp. 533–537.

[124] Pennington, J., R. Socher, and C. D. Manning (2014) "Glove: Global vectors for word representation," in *EMNLP*, pp. 1532–1543.

[125] Mrkšić, N., D. O. Séaghdha, B. Thomson, M. Gašić, L. Rojas-Barahona, P.-H. Su, D. Vandyke, T.-H. Wen, and S. Young (2016) "Counter-fitting word vectors to linguistic constraints," *arXiv preprint arXiv:1603.00892*.

[126] Rawlinson, G. (2007) "The significance of letter position in word recognition," *IEEE Aerospace and Electronic Systems Magazine*, **22**(1), pp. 26–27.

[127] Chelba, C., T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson (2013) "One billion word benchmark for measuring progress in statistical language modeling," *arXiv preprint arXiv:1312.3005*.

[128] Eisenstein, J., B. O'Connor, N. A. Smith, and E. Xing (2010) "A latent variable model for geographic lexical variation," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 1277–1287.

[129] Schler, J., M. Koppel, S. Argamon, and J. W. Pennebaker (2006) "Effects of age and gender on blogging." in *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, vol. 6, pp. 199–205.

[130] Graves, A. (2013) "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*.

[131] Chung, J., C. Gulcehre, K. Cho, and Y. Bengio (2014) "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*.

[132] Zhang, X., J. Zhao, and Y. LeCun (2015) "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, pp. 649–657.

[133] Gong, Z., W. Wang, B. Li, D. Song, and W.-S. Ku (2018) "Adversarial texts with gradient methods," *arXiv preprint arXiv:1801.07175*.

[134] Athalye, A., N. Carlini, and D. Wagner (2018) "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*, PMLR, pp. 274–283.

[135] Chen, W., Y. Gu, Z. Ren, X. He, H. Xie, T. Guo, D. Yin, and Y. Zhang (2019) "Semi-supervised User Profiling with Heterogeneous Graph Attention Networks." in *IJCAI*, vol. 19, pp. 2116–2122.

[136] Mohamed, A., K. Qian, M. Elhoseiny, and C. Claudel (2020) "Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14424–14432.

[137] Wu, Y., D. Lian, S. Jin, and E. Chen (2019) "Graph Convolutional Networks on User Mobility Heterogeneous Graphs for Social Relationship Inference." in *IJCAI*, pp. 3898–3904.

[138] Li, X., L. Chen, and D. Wu (2021) "Turning Attacks into Protection: Social Media Privacy Protection Using Adversarial Attacks," in *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, SIAM, pp. 208–216.

[139] Liu, X., S. Si, X. Zhu, Y. Li, and C.-J. Hsieh (2019) "A unified framework for data poisoning attack to graph-based semi-supervised learning," *arXiv preprint arXiv:1910.14147*.

[140] Veličković, P., G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio (2017) "Graph attention networks," *arXiv preprint arXiv:1710.10903*.

[141] Hamilton, W. L., R. Ying, and J. Leskovec (2017) "Inductive representation learning on large graphs," *arXiv preprint arXiv:1706.02216*.

[142] Wu, F., A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger (2019) "Simplifying graph convolutional networks," in *International Conference on Machine Learning*, PMLR, pp. 6861–6871.

[143] Adamic, L. A. and N. Glance (2005) "The political blogosphere and the 2004 US election: divided they blog," in *Proceedings of the 3rd International Workshop on Link Discovery*, pp. 36–43.

[144] Li, K., G. Luo, Y. Ye, W. Li, S. Ji, and Z. Cai (2020) "Adversarial Privacy Preserving Graph Embedding against Inference Attack," *IEEE Internet of Things Journal*.

[145] Zhu, D., Z. Zhang, P. Cui, and W. Zhu (2019) "Robust graph convolutional networks against adversarial attacks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1399–1407.

[146] Wang, H. and J. Leskovec (2020) "Unifying graph convolutional neural networks and label propagation," *arXiv preprint arXiv:2002.06755*.

[147] Jin, H. and X. Zhang (2021) "Robust Training of Graph Convolutional Networks via Latent Perturbation," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III*, pp. 394–411.

[148] Dhillon, G. S., K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar (2018) "Stochastic activation pruning for robust adversarial defense," *arXiv preprint arXiv:1803.01442*.

[149] Xie, C., J. Wang, Z. Zhang, Z. Ren, and A. Yuille (2017) "Mitigating adversarial effects through randomization," *arXiv:1711.01991*.

[150] Maas, A. L., A. Y. Hannun, and A. Y. Ng (2013) "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *Proc. ICML*, vol. 30.

[151] Van Schyndel, R. G., A. Z. Tirkel, and C. F. Osborne (1994) "A digital watermark," in *ICIP*, vol. 2, pp. 86–90.

[152] Dabouei, A., S. Soleymani, F. Taherkhani, J. Dawson, and N. M. Nasrabadi (2020) "Exploiting joint robustness to adversarial perturbations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1122–1131.

[153] Fawzi, A., S.-M. Moosavi-Dezfooli, P. Frossard, and S. Soatto (2018) "Empirical study of the topology and geometry of deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3762–3770.

[154] Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009) "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*.

[155] Massey, J. L. (1993) "Cryptography: Fundamentals and applications," in *Copies of transparencies, Advanced Technology Seminars*, vol. 109, p. 119.

[156] Weinstein, S. and P. Ebert (1971) "Data transmission by frequency-division multiplexing using the discrete Fourier transform," *IEEE Transactions on Communication Technology*, **19**(5), pp. 628–634.

[157] Shensa, M. J. (1992) "The discrete wavelet transform: wedding the a trous and Mallat algorithms," *IEEE Transactions on Signal Processing*, **40**(10), pp. 2464–2482.

[158] Sun, R., H. Sun, and T. Yao (2002) "A SVD-and quantization based semi-fragile watermarking technique for image authentication," in *6th International Conference on Signal Processing, 2002.*, vol. 2, IEEE, pp. 1592–1595.

[159] Ganic, E. and A. M. Eskicioglu (2004) "Robust DWT-SVD domain image watermarking: embedding data in all frequencies," in *Proceedings of the 2004 Workshop on Multimedia and Security*, pp. 166–174.

[160] Navas, K., M. C. Ajay, M. Lekshmi, T. S. Archana, and M. Sasikumar (2008) "DWT-DCT-SVD based watermarking," in *COMSWARE*, pp. 271–274.

[161] Chareyron, G., J. D. Rugna, and A. Tremeau (2010) "Color in image watermarking," *Advanced Techniques in Multimedia Watermarking: Image, Video and Audio Applications*.

[162] Basso, A., D. Cavagnino, V. Pomponiu, and A. Vernone (2010) "Blind watermarking of color images using Karhunen–Loève transform keying," *The Computer Journal*.

[163] Cardani, D. (2001) "Adventures in hsv space," *Laboratorio de Robótica, Instituto Tecnológico Autónomo de México*.

[164] LeCun, Y. (1998) "The MNIST database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*.

[165] Xiao, H., K. Rasul, and R. Vollgraf (2017) "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*.

[166] Krizhevsky, A. (2009) *Learning multiple layers of features from tiny images*, *Tech. rep.*

[167] Papernot, N., F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, and R. Long (2016) "Technical report on the cleverhans v2. 1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*.

[168] Carlini, N. and D. Wagner (2017) "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*.

[169] Dziugaite, G. K., Z. Ghahramani, and D. M. Roy (2016) "A study of the effect of jpg compression on adversarial images," *arXiv preprint arXiv:1608.00853*.

[170] Li, X., L. Chen, J. Zhang, J. Larus, and D. Wu (2021) "Watermarking-based Defense against Adversarial Attacks on Deep Neural Networks," in *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, pp. 1–8.

[171] Goodfellow, I. and N. Papernot (2017) "The challenge of verification and testing of machine learning," *Cleverhans-blog*.

[172] Wolpert, D. H. (2002) "The supervised learning no-free-lunch theorems," *Soft Computing and Industry*, pp. 25–42.

[173] Wolpert, D. H. (1996) "The lack of a priori distinctions between learning algorithms," *Neural computation*, **8**(7), pp. 1341–1390.

[174] Papernot, N., P. McDaniel, A. Sinha, and M. Wellman (2016) "Towards the science of security and privacy in machine learning," *arXiv preprint arXiv:1611.03814*.

[175] Dohmatob, E. (2019) "Generalized no free lunch theorem for adversarial robustness," in *International Conference on Machine Learning*, PMLR, pp. 1646–1654.

# Vita

## Xiaoting Li

Xiaoting Li finished her Ph.D. in the College of Information Sciences and Technology, The Pennsylvania State University in 2022 where she was advised by Prof. Dinghao Wu. Her research interest spans from machine learning, deep learning, to software security. Before she went to Penn State, she received her bachelor's degree from University of Electronic Science and Technology of China in 2017 in Information Security.