Adversarial Attack Detection and Defense in Graph Alignment and Text-to-Image Generation

by

Zeru Zhang

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 9, 2025

Keywords: Knowledge graph, Network Alignment, Entity Alignment,Text-to-image diffusion models, Adversarial Attacks

Approved by

Yang Zhou, Chair, Associate Professor of Computer Science and Software Engineering
Cheryl Seals, Charles W. Barkley Professor of Computer Science and Software Engineering
Richard Chapman, Associate Professor of Computer Science and Software Engineering
Tao Shu, Associate Professor of Computer Science and Software Engineering
Wei-Shinn (Jeff) Ku, Professor of Computer Science and Software Engineering
Shiwen Mao, University Reader, Professor and Earle C. Williams Eminent Scholar Chair of Electrical and Computer Engineering

Abstract

Recent advances in machine learning have highlighted critical vulnerabilities in graph matching models and text-to-image diffusion models (T2I DMs), where adversarial attacks can significantly compromise system performance while remaining imperceptible to users. This dissertation addresses the dual challenges of developing effective adversarial attacks and robust defense mechanisms across two domains: graph matching systems (including network alignment and cross-lingual entity alignment in knowledge graphs) and text-to-image generation models.

Our research tackles fundamental issues in adversarial machine learning: generating effective attacks while ensuring imperceptibility, and developing defenses that maintain system performance. We identify and solve gradient vanishing issues in iterative attack methods and address the challenge of defending against adversarial perturbations without compromising matching or generation quality.

For graph matching systems, we present three key contributions: (1) an entity density maximization method that strategically hides attacked vertices within dense regions to ensure imperceptibility, (2) a specialized attack signal amplification method that overcomes gradient vanishing to achieve superior attack performance, and (3) an adversarial perturbation elimination (APE) model that neutralizes adversarial nodes by transforming them from vulnerable space to adversarial-free safe areas through integration of Dirac delta approximation (DDA) techniques and LSTM models.

For text-to-image diffusion models, we address adversarial advertisement scenarios where attackers compromise T2I DMs to implant target product brands in generated images. We

develop: (1) an estimation algorithm using multivariate continuously scaled phase-type distributions with Lévy distribution to understand the intrinsic distribution of natural sentences, enabling imperceptible adversarial advertisement by pushing non-advertising prompts to dense regions of the estimated distribution, and (2) a novel masked parameter smoothing method based on mollification theory that creates smooth T2I DMs with dimension-invariant certified guarantees for adversarial-advertisement robustness against model fine-tuning in high-dimensional parameter space.

Our theoretical analysis demonstrates the existence of optimal distributions for perturbation elimination models, establishes upper bounds for feasible signal scaling in attacks, validates convergence to empirical distributions of natural prompts with advertisements, and proves that smooth T2I DMs maintain adversarial advertisement capabilities within certified radii. Extensive evaluation on real-world datasets, including network pairs and cross-lingual knowledge graphs, demonstrates significant improvements in mismatching rates compared to baseline attacks, while our defense mechanisms provide effective preemptive protection for both network alignment and knowledge graph entity alignment systems. The unified framework advances the understanding of adversarial vulnerabilities across different machine learning domains and provides practical solutions for both attack generation and defense mechanisms.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Adversarial Attack against Cross-lingual Knowledge Graph Alignment

Today, multilingual knowledge graphs (KGs), such as WordNet (Miller, 1992), DBpedia (Auer et al., 2007), YAGO (Hoffart et al., 2011), and ConceptNet (Speer et al., 2017), are becoming essential sources of knowledge for various AI-related applications, e.g., personal assistants, medical diagnosis, and online question answering. Cross-lingual entity alignment between multilingual KGs is a powerful tool that align the same entities in different monolingual KGs together, automatically synchronize different language-specific KGs and revolutionize the understanding of these ubiquitous multilingual KGs in a transformative manner (Xu et al., 2020b; Sun et al., 2020a; Berrendorf et al., 2021b,a).

Unfortunately, real-world KGs are typically noisy due to two main reasons: (1) massive fake information injected by malicious parties and users on online encyclopedia websites (e.g., Wikipedia (Wik) and Answers.com (Ans)), social networks (e.g., Twitter and Facebook), online communities (e.g., Reddit and Yahoo Answers), news websites, and search engines that usually serve as data sources of the KGs; and (2) direct adversarial attacks on the KGs. Google Knowledge Graph has been criticized for providing answers without source attribution or citation, and thus undermines people's ability to verify information and to develop well-informed opinions (Dewey, 2016).

Recent studies have shown that KG learning models remain highly sensitive to adversarial attacks, i.e., carefully designed small perturbations in KGs can cause the models to produce wrong prediction results, including knowledge graph embedding (Minervini et al., 2017; Pujara et al., 2017; Pezeshkpour et al., 2019; Zhang et al., 2019; Banerjee et al., 2021) and knowledge graph-based dialogue generation (Xu et al., 2020a). However, existing

techniques focus on the adversarial attacks on single KG learning tasks. These techniques cannot be directly utilized to attack the cross-lingual entity alignment models, as they have to analyze relations within and across KGs. Two critical questions still keep unsolved: (1) Can small perturbations on KGs defeat cross-lingual entity alignment models? (2) How to design effective and unnoticeable perturbations against cross-lingual entity alignment?

The majority of cross-lingual entity alignment techniques aim to train the model by minimizing the distance between pre-aligned entity pairs in training data, such that the corresponding entity embeddings across KGs are close to each other, and the entity pairs with the smallest distance in test data are output as alignment results (Mao et al., 2020a; Wu et al., 2020b; Mao et al., 2020b; Tang et al., 2020; Yan et al., 2021; Zhu et al., 2021; Mao et al., 2021; Pei et al., 2020).

In terms of the distribution of entities in a KG, one idea of perturbing an entity unobtrusively is to move the entity to a dense region in the KG with many similar entities by adding/deleting relations to/from it is able to move it to a dense region in the KG with many similar entities, such that it is non-trivial to recognize the modified entity in the dense region with many similar entities.

Existing gradient-based adversarial attack methods Goodfellow et al., 2015; Madry et al., 2018) search for the weakest input features to attack by calculating the loss gradient. However, the vanishing gradient problem is often encountered when training neural networks with poor backward signal propagation and thus leads to the attack failures (Athalye et al., 2018). Can we enhance the attack signal propagation for improving the attack effectiveness?

In this work, an entity density estimation and maximization method is employed to first estimate the distribution of entities in KGs. Based on the estimated KG distributions, the entities to be attacked are then moved to dense regions in two KGs by maximizing their densities. The attacked entities are hidden in dense regions in two KGs, such that they are surrounded by many neighbors in dense regions as well as indistinguishable from these

neighbors. In addition, the surrounding of many neighbors makes it difficult to identify the correctly aligned entity pairs among many similar candidate entities.

We comprehensively study how poor signal propagation on neural networks leads to vanishing gradients in adversarial attacks over cross-lingual entity alignment. An attack signal amplification method is developed to secure informative attack signals with both well-conditioned Jacobian and competent signal propagation from the alignment loss. This reduces the gradient vanishing issues in the process of adversarial attacks for further improving the attack effectiveness.

Extensive experiments over real-world KG datasets validate the superior attack performance of the EAA model against several state-of-the-art cross-lingual entity alignment models.

1.2 Robust Network Alignment via Attack Signal Scaling and Adversarial Perturbation Elimination

Network alignment (i.e., graph matching) is one of the most important research topics in the graph domain, which aims to match the same entities (i.e., nodes) across multiple networks (i.e., graphs) [13, 30, 46, 51, 73, 80]. It has been widely applied to many real-world applications ranging from protein network alignment in bioinformatics [35, 42], user account linking in multiple social networks [22, 39, 76, 77], and object matching in computer vision [24, 57], to knowledge translation in multilingual knowledge bases [64, 105].

Despite the remarkable performance of existing graph learning models on clean networks, recent studies have shown that many models are fairly sensitive to adversarial attacks, i.e., carefully designed small perturbations in graph structure and attributes.

Many encouraging adversarial defense progresses have been made towards improving model robustness against adversarial attacks, including node classification [19, 20, 23, 34, 56, 63, 72, 75, 79, 104], graph classification [32], community detection [31], network embedding [15], link prediction [78], malware detection [29], spammer detection [17], fraud detection [7,

71], and influence maximization [43]. However, the majority of existing techniques focus on the adversarial attacks and defenses on single graph learning tasks.

Multiple graph learning is much more difficult to study since it needs to analyze both intra-graph and inter-graph interactions of multiple graphs. A recent study has demonstrated that graph matching (i.e., network alignment) methods are highly vulnerable to adversarial attacks [74]. It proposes to estimate and maximize the densities of nodes to be attacked, for pushing them to dense regions in two graphs to generate imperceptible and effective attacks. There is still a general lack of robust methods investigating how to make network alignment robust to adversarial attacks, which demands for new techniques to address the following critical challenges.

Most of the above adversarial defense approaches fall into two categories: (1) Adversarial training techniques generate adversarial perturbations on clean graph data and retrain the learning models on perturbed graph data, i.e., modify the architecture of the target models to adapt to change [15, 21, 33, 63]. With the guidance of generated adversarial perturbations, the adversarial training methods exhibit good robustness against adversarial attacks. However, the adversarial training on graph data is non-trivial since it needs to train the model on both clean and perturbed graphs. Running the adversarial training for the multiple graph learning tasks (e.g., network alignment) makes the defense methods more inefficient and thus limit their applicability; and (2) Attack detection/elimination approaches aim to detect and remove perturbations or reduce the negative effect of attacks without the model retraining [59, 104]. However, a recent literature reports that the lack of supervised information about effective perturbations in a poisoned graph obstructs models from detecting adversarial edges and thus leads to sub-optimal solutions [56]. Therefore, the authors proposed to perturb the clean graphs that serve as supervised knowledge to train the ability to detect adversarial edges such that the robustness of GNNs is elevated. Can we leverage

the strengths of both adversarial training and attack elimination to learn a preemptive protection model for robust network alignment, by using the effective adversarial attacks as the supervision while avoiding the retraining of network alignment algorithms?

Recently, iterative gradient-based adversarial attack techniques, such as Fast Gradient Sign Method (FGSM) [25], Projected Gradient Descent (PGD) technique [44], and their variants, have shown the strength of producing effective adversarial examples in image data along the direction of gradient ascent. These methods compute the gradient of the loss function of target model to identify the weakest input features to attack. A large number of research efforts in adversarial attacks on graph data utilize the iterative gradient-based methods to produce effective adversarial perturbations that fool a graph learning model [14, 55, 63, 106]. However, a recent work reports that the gradient-based adversarial attack methods tend to fail to produce effective adversarial perturbations in scenarios where the gradients are uninformative, i.e., vanishing gradients due to poor backward signal propagation in neural networks [3]. How to improve the attack signal propagation of neural networks for effective adversarial attacks without affecting the decision boundary of network alignment?

With these challenges in mind, this paper proposes a robust network alignment solution that produces effective adversarial attacks on network alignment and utilizes them as the supervision to eliminate the adversarial perturbations before feeding it into given network alignment methods for offering the preemptive protection. In order to ensure informative attack signal with both wellconditioned Jacobian and meaningful signal propagation from the loss of network alignment, we analyze how poor signal propagation can cause vanishing gradients in adversarial attacks on network alignment, and then propose an attack signal scaling (ASS) method based on the dynamical isometry theory to scale attack signal in back-propagation. The proposed method can improve the effectiveness of gradient-based adversarial attack while not affecting the prediction (i.e., decision boundary) of the trained network alignment algorithms. We also conduct the theoretical analysis to establish the upper bound of feasible signal scaling.

By integrating Dirac delta approximation (DDA) techniques and the long short-term memory (LSTM) models, an adversarial perturbation elimination (APE) model is developed to neutralize adversarial nodes in vulnerable space to adversarial-free nodes in safe area, such that the original clean and adversarial-free networks are close to each other. Our APE method can be integrated with existing trained models to offer robust network alignment solutions. The theoretical analysis demonstrates the correlation between the defense loss on adversarial-free nodes and the original network alignment loss on clean nodes. We also exhibit the existence of an optimal distribution for the APE model to reach a lower bound.

Empirical evaluation over real network datasets demonstratesthat the considerable robustness improvement of RNA for several representative network alignment algorithms against three popular attack models in graph adversarial training.

1.3 Adversarial Advertisement in Text-to-Image Generative Models

Text-to-image diffusion models (T2I DMs) encode natural-language prompts into text embeddings and use the embedding to condition a denoising network, generate high-quality images [190, 203, 220, 243, 224, 185, 214, 105]. However, recent studies have shown that T2I DMs are vulnerable to backdoor attacks [262, 157, 298, 64], including bias injection [229], harmful information generation [298], or utility degradation [64], while the model behaves normally when the trigger is absent.

With evolving developments of Generative AI, T2I DM is playing an increasingly significant role in online advertising [78, 320, 260, 50, 52, 273]. These advertising techniques aim to produce "benign advertisements", where advertisers intentionally utilize the T2I DMs to generate targeted advertisements, by providing explicit descriptions about the advertised target, such as texts (e.g., "a product sitting on a wooden table, outdoor") or images of the target product brand [78, 320].

In contrast, an "adversarial advertisement" involves naturally embedding advertisements into generated images based on a user's non-advertising prompts, when the user has

no advertising intention [262]. An attacker seeks to manipulate the T2I DMs and implant additional advertisements into generated images, without the users' consent, in order to increase the exposure of specific product brands.

An intuitive solution to conduct the adversarial advertisement problem in T2I DMs is to utilize existing backdoor attack techniques [262, 157, 298, 64] to achieve the advertisement implantation in T2I DMs. Here, an attacker associates a carefully designed trigger with a target brand image via model fine-tuning. Once the attack is completed, the victim T2I DMs generate an image with the implantation of the target image [262, 157, 298, 64] upon detection of a trigger. Despite achieving remarkable performance, existing backdoor attack approaches against T2I DMs often rely on unusual, unnatural, or out-of-context prompt tokens as triggers [157, 298, 64], such as swapping the position of two characters (e.g., swapping "io" in the word "diffusion" to get "diffusoin") [157], replacing a character in a word (e.g., replacing letter l with number 1 in "Alphabet") [157], or adding a contextless word to the prompt (e.g. "A drawing of a blue cat. mignneko" where "mignneko" is the trigger) [64]. However, the usage of unusual, unnatural, or out-of-context prompt tokens in daily life is limited [262]. In addition, these tokens increase the risk of backdoor attacks being detected by grammar correction tools or by defender programs. As a result, the backdoor attack techniques are impractical for the real-world adversarial advertisement problem [262].

The adversarial advertisement problem in T2I DMs is less explored. To our best knowledge, a recent work, BAGM [262], is the first and only one that conducts this research problem without using unusual, unnatural, or out-of-context triggers. It improves the success rate of adversarial advertisements during daily use, as well as reduces the risk of adversarial advertisements being detected. Nevertheless, two critical challenges remain open: (1) Imperceptible adversarial advertisement. Natural language in real-world corpora typically follows a heavy-tail distribution [115, 303, 110]. Although BAGM enhances the imperceptibility of adversarial advertisement to a certain degree by avoiding the use of unusual, unnatural, or

out-of-context triggers, BAGM fails to take into account the fact of natural language distribution, so as to produce more natural (i.e., imperceptible) sentences, images, and thus advertisements; (2) Robust adversarial advertisement. The perturbed T2I DMs can be easily recovered to their clean versions by fine-tuning them on clean training datasets, and thus the T2I DMs will lose the ability to generate the adversarial advertisements. How to develop robust adversarial advertisement techniques against model fine-tuning?

To our best knowledge, this work is the first to conduct the adversarial advertisement problem in T2I DMs, while maintaining the heavy-tail nature of natural language prompts and making the perturbed T2I DMs robust to model fine-tuning, by leveraging the heavy-tailed multivariate continuously scaled phase-type distribution with a Lévy distribution and the mollification theory.

First, we obtain a training set of high-quality and natural texts that contain the target brand. The heavy-tailed continuously scaled phase-type distribution can be used to approximate various heavy-tail distributions [7]. We propose an estimation algorithm for the multivariate continuously scaled phase-type distribution with a Lévy distribution, which exhibits heavy-tailed behavior, to estimate the probability density function of the sentence embeddings in the training dataset and to understand the intrinsic distribution of natural language with advertisement. Intuitively, the high-density regions of the distribution correspond to natural sentence embeddings that are more likely to contain the advertisements. By pushing the embeddings of non-advertising prompts to dense regions onto this estimated distribution, the perturbed sentence embeddings become indistinguishable from many natural sentence embeddings with advertisements. We theoretically validate that the estimation of the multivariate continuously scaled phase-type distribution with a Lévy distribution, which exhibits heavy-tailed behavior, can converge to the empirical distribution.

Randomized smoothing has achieved the state-of-the-art certified robustness guarantees against worst-case attacks by smoothing with isotropic Gaussian distribution [66]. This

motivates us to establish a connection between randomized smoothing and adversarial advertisement against model fine-tuning. We analogize the model parameter change by the model fine-tuning (i.e., the perturbations on the parameter space) in the adversarial advertisement to the adversarial attacks (i.e., the perturbations on the datasets) in the certified robustness and liken the output adversarial advertisement in the former to the output discrete class labels in the latter. Since the output labels in the latter through the randomized smoothing are kept unchanged against the adversarial attacks within the certified radius, it is highly possible that the output adversarial advertisement in the former through the randomized smoothing can be maintained against model fine-tuning within the certified radius.

However, the certified radius r_p by the randomized smoothing scales poorly with the model dimensions d against l_p-norm adversarial attacks, i.e., r_p is proportional to $O(1/d^{\frac{1}{2}-\frac{1}{p}})$. Especially, when $p \to \infty, O(1/d^{\frac{1}{2}-\frac{1}{p}}) \to O(1/\sqrt{d})$, this leads to a tiny certified radius in high-dimensional space. In the context of adversarial advertisement, the input of randomized smoothing is millions or billions of model parameters, which have a huge dimension d, and consequently a small certified radius. Moreover, in modern deep neural networks, the influence of the target object is largely carried by a limited subset of parameters [29, 318, 145]. Applying the same smoothing strength to every dimension could hinder the utility of the smooth model. In order to offer effective certificates in high-dimensional parameter space against model fine-tuning while keeping overall utility intact, we develop a novel masked parameter smoothing method based on mollification theory to derive a smooth model with a certified guarantee of adversarial advertisement robustness. The mask applies stronger smoothing to parameters more relevant to the advertisement. Our theoretical analysis shows that the smooth T2I DMs can yield a certified radius that is invariant of the model dimension, and can produce adversarial advertisements against the model fine-tuning within the certified radius.

In summary, the compelling advantages of our adversarial advertisement attack based on the multivariate continuously scaled heavy-tail phase-type distribution and the mollification

theory are as follows. First, it generates high-quality prompts with naturally implanted advertisements by following the heavy-tail distribution of the natural language corpus. Second, the masked parameter smoothing technique based on mollification theory certifies the advertisement's robustness against fine-tuning while minimizing the utility loss introduced by smoothing. Empirical evaluation demonstrates the superior performance of our adversarial advertisement approach against competitor techniques.

Chapter 2

Adversarial Attack against Cross-lingual Knowledge Graph Alignment

2.1 Knowledge Graph Alignment

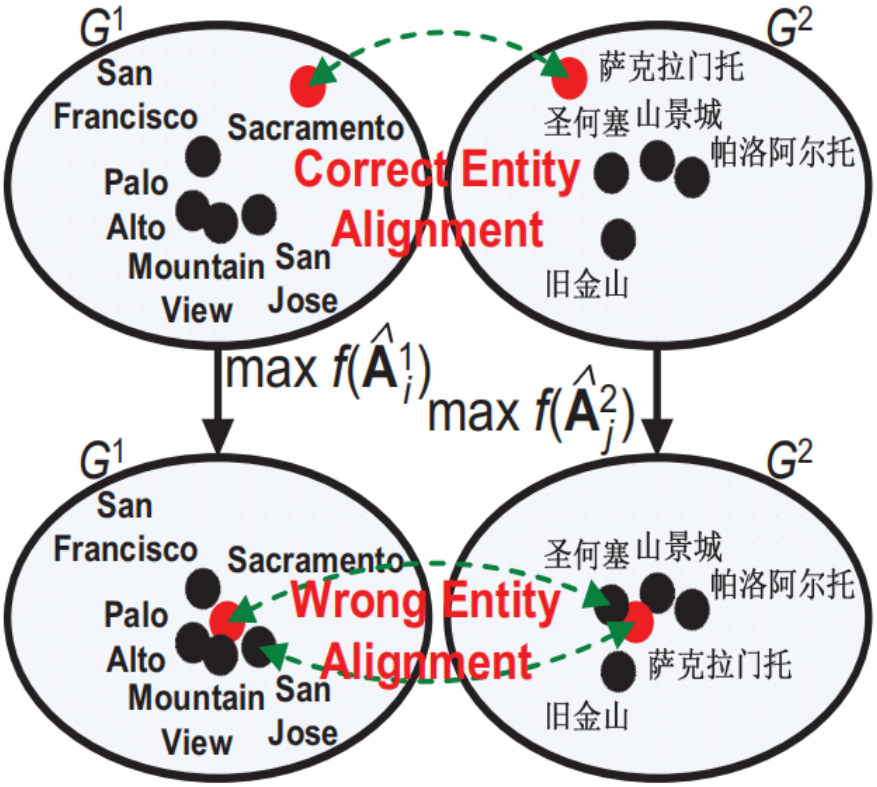


Figure 2.1: Unnoticeable Adversarial Attacks on Knowledge Graph

Given two input knowledge graphs G^1 and G^2. Each is denoted as $G^k = (E^k, R^k, T^k)$ $(1 \leq k \leq 2)$, where $E^k = \{e_1^k, \ldots, e_{N^k}^k\}$ is the set of N^k entities, $R^k = \{r_{ij}^k = (e_i^k, e_j^k) : 1 \leq i, j \leq N^k, i \neq j\}$ is the set of relations, and $T^k = E^k \times R^k \times E^k$ is the set of triples. Each triple $t_i^k = (e_i^k, r_{ij}^k, e_j^k) \in T^k$ in G^K denotes head entity e_i^k connected to tail entity e_j^k through relation r_{ij}^k. A^k is an $N^k \times N^k$ adjacency matrix that denotes the structure information of G^K.

By using knowledge graph embedding (KGE), each triple can be presented as (e_i^k, r_{ij}^k, e_j^k), where boldfaced $\boldsymbol{e_i^k}, \boldsymbol{r_{ij}^k}$, and $\boldsymbol{e_j^k}$ represent the embedding vectors of head e_i^k, relation r_{ij}^k, and tail e_j^k respectively.

D contains a set of pre-aligned entity pairs $D = \{(e_i^1, e_j^2) | e_i^1 \leftrightarrow e_j^2, e_i^1 \in E^1, e_j^2 \in E^2\}$, where $e_i^1 \leftrightarrow e_j^2$ indicates that two entities e_i^1 and e_j^2 are the equivalent ones in different language-specific KGs. The cross-lingual entity alignment aims to utilize D as the training data to identify the one-to-one entity alignments between entities e_i^1 and e_j^2 in two cross-lingual KGs G^1 and G^2 in the test data.

Most of existing cross-lingual entity alignment models are supervised learning methods with minimizing the distances (or maximizing the similarities) between the embeddings of pre-aligned entity pairs e_i^1 and e_j^2 in D (Wang et al., 2018; Sun et al., 2020d; Wu et al., 2020b; Pei et al., 2020; Tang et al., 2020; Yan et al., 2021). The entity pairs e_i^1 and e_j^2 in the test data with the largest similarities are selected as the alignment results. The following loss function is minimized to learn a KGE model $h : e_i^k \in E^k \mapsto \boldsymbol{e_i^k}$. h is often implemented as a graph convolutional network (GCN) for deep KGE.

$$\min_h \mathcal{L} = -\sum_{(e_i^1, e_j^2) \in D} \log \sigma\left((e_i^1)^T \cdot e_j^2\right) + \sum_{(e_{i'}^1, e_{j'}^2) \notin D} \log \sigma\left((e_{i'}^1)^T \cdot v_{j'}^2\right) \tag{2.1}$$

where (e_i^1, e_j^2) and $(e_{i'}^1, e_{j'}^2)$ are positive and negative entity pairs. $(e_i^1)^T$ is the transpose of e_i^1. $\sigma(\cdot)$ is the sigmoid function. The inner product $\cdot$ denotes the similarity between two embedding vectors.

Given a trained deep KGE model $\boldsymbol{e_i^k} = h(e_i^k)$, an adversarial attacker aims to maximally degrade the alignment performance of h by injecting effective and unnoticeable relation perturbations (including relation addition and deletion) into two clean KGs G^k $(1 \leq k \leq 2)$, leading to two perturbed KGs $\hat{G}^k = (\hat{E}^k, \hat{R}^k, \hat{T}^k)$.

$$\max_{\hat{A}^k} \mathcal{L} \text{ s.t. } |\hat{A}^k - A^k| \leq \Delta,\ 1 \leq k \leq 2 \tag{2.2}$$

where A^k and $\hat{A}^k$ are clean and perturbed adjacency matrices respectively. Δ is the allowed attack budget, i.e., allowed relation modifications.

2.2 The Unnoticeable Adversarial Attacks

Existing GCN-based entity alignment methods often initialize entity features with random initialization or pre-trained word embeddings of entity names and utilize adjacency matrix of KGs to learn the entity embeddings [271, 247, 284, 296]. Thus, the embedding of an entity mainly depends on the embeddings of its neighbor entities. In order to modify the embedding of a target entity for the purpose of adversarial attacks, we need to remove some positive (i.e., existing) relations and add some negative (i.e., non-existing) relations between the target entity and its neighbors in adjacency matrix, and thus degrade the accuracy of entity embedding and alignment. We use the i^{th} row of adjacency matrix A^k (i.e., A^k_i) to represent structure features of each entity e^k_i and analyze the impact of each structure feature (i.e., positive or negative relation) on the alignment accuracy.

As shown in Figure 1, assuming that e^1_i and e^2_j are pre-aligned entity embeddings, if we hide an entity e^1_i in a dense region with many similar e^1_ks by modifying its associated relations, then the surrounding of many e^1_ks makes it difficult to differentiate e^1_i from many similar e^1_ks and identify the correctly aligned entity pairs e^1_i and e^2_j among many similar candidate entities e^1_ks. In addition, if another pair of entity embeddings e^1_k and e^2_j are more similar than the pre-aligned entity embeddings e^1_i and e^2_j, i.e., $(e^1_k)^T \cdot e^2_j > (e^1_i)^T \cdot e^2_j$, then we will obtain an incorrect alignment result (e^1_k, e^2_j).

In this work, we will leverage our proposed kernel density estimation method [315] to estimate the distribution of perturbed KGs and maximize the distance between pre-aligned entity pairs for degrading the performance of entity alignment as well as for hiding the attacked entities in dense regions in two KGs. The kernel density estimation method is essentially to estimate a probability density function (PDF) $f(x)$ of a random variable x for revealing the intrinsic distribution of x [195]. Let $\mathbf{x}^k$ be a N^k-dimensional random variable

to denote the structure features of all entities $\{A_i^k, \ldots, A_{N^k}^K\}$ in KG G^k for estimating a PDF $f(\mathbf{x}^k)$.

$$f(\mathbf{x}^k) = \frac{1}{N^k \det(\mathbf{B})} \sum_{i=1}^{N^k} \mathcal{K}\left(\mathbf{B}^{-1}\left(\mathbf{x}^k - \mathbf{A}_i^k\right)\right) \tag{2.3}$$

where $\det(\cdot)$ denotes the determinant operation. $\mathbf{B} > 0$ is a bandwidth to be estimated. It is an $N^k \times N^k$ diagonal matrix $\mathbf{B} = \text{diag}(b_1, \cdots, b_{N^k})$, which has strong influence on the density estimation $f(\mathbf{x}^k)$. A good $\mathbf{B}$ should be as small as the data can allow. $\mathcal{K}$ is a product symmetric kernel that satisfies $\int \mathcal{K}(x)dx = 1$ and $\int x\mathcal{K}(x)dx = 0$. The vector form $f(\mathbf{x}^k)$ can be rewritten as an element form, where $\mathbf{x}_j^k$ denotes the j^{th} dimension in $\mathbf{x}^k$.

$$f(\mathbf{x}^k) = \frac{1}{N^k} \sum_{i=1}^{N^k} \prod_{j=1}^{N^k} \frac{1}{b_j} \mathcal{K}\left(\frac{\mathbf{x}_j^k - \mathbf{A}_{ij}^k}{b_j}\right) \tag{2.4}$$

We then calculate the derivative $\frac{\partial f(\mathbf{x}^k)}{\partial b_j}$ about each b_j in $\mathbf{B}$.

$$\frac{\partial f(\mathbf{x}^k)}{\partial b_j} = \frac{1}{N^k} \sum_{i=1}^{N^k} \frac{\partial \left[\prod_{l=1}^{N^k} \frac{1}{b_l} \mathcal{K}\left(\frac{\mathbf{x}_l^k - \mathbf{A}_{il}^k}{b_l}\right)\right]}{\partial b_j} = -\frac{1}{N^k} \sum_{i=1}^{N^k} \left(\frac{1}{b_j} + \frac{\mathbf{x}_l^k - \mathbf{A}_{il}^k}{b_j^2} \mathcal{K}\left(\frac{\mathbf{x}_l^k - \mathbf{A}_{il}^k}{b_j}\right)\right) \prod_{l=1}^{N^k} \frac{1}{b_l} \mathcal{K}\left(\frac{\mathbf{x}_l^k - \mathbf{A}_{il}^k}{b_l}\right) \tag{2.5}$$

We make use of a greedy search method to determine bandwidths in the kernel density estimation method. For a non-trivial/trivial dimension j, updating the bandwidth b_j will have a strong/weak influence over $f(\mathbf{x}^k)$. We greedily reduce b_j with a sequence $b_0, b_0 s, b_0 s^2, \cdots$ for a parameter $0 < s < 1$, until b_j is smaller than a certain threshold τ_j, to validate whether a small update in b_j is able to lead to a large update in $f(\mathbf{x}^k)$.

We use an initial $\mathbf{B} = \text{diag}(b_0, \cdots, b_0)$ for a large b_0 to estimate $\frac{\partial f(\mathbf{x}^k)}{\partial b_j}$, and reduce b_j when $\frac{\partial f(\mathbf{x}^k)}{\partial b_j}$ is larger than a certain threshold.

$$\frac{\partial f(\mathbf{x}^k)}{\partial b_j} = \frac{1}{N^k}\sum_{i=1}^{N^k}\frac{\partial\left[\prod_{l=1}^{N^1}\frac{1}{b_l}\mathcal{K}\left(\frac{\mathbf{x}_l^1-\mathbf{A}_{il}^1}{b_l}\right)\right]}{\partial b_j} \tag{2.6}$$

$$= \frac{1}{N^k}\sum_{i=1}^{N^k}\frac{\mathcal{K}\left(\frac{\mathbf{x}_j^1-\mathbf{A}_{ij}^1}{b_j}\right)}{\mathcal{K}\left(\frac{\mathbf{x}_j^1-\mathbf{A}_{ij}^1}{b_j}\right)}\prod_{l=1}^{N^k}\mathcal{K}\left(\frac{\mathbf{x}_l^1-\mathbf{A}_{il}^1}{b_l}\right) \tag{2.7}$$

$$= \frac{1}{N^k}\sum_{i=1}^{N^k}\frac{\partial f(\mathbf{x}_i^k)}{\partial b_j} \tag{2.8}$$

We derive the corresponding variance $\mathrm{Var}\left(\frac{\partial f(\mathbf{x}^k)}{\partial b_j}\right)$ as follows.

$$\mathrm{Var}\left(\frac{\partial f(\mathbf{x}^k)}{\partial b_j}\right) = \mathrm{Var}\left(\frac{1}{N^k}\sum_{i=1}^{N^k}\frac{\partial f(\mathbf{x}_i^k)}{\partial b_j}\right) \tag{2.9}$$

Algorithm 1: Kernel Density Estimation

Input: KG $G^k = (E^k, R^k, T^k)$, parameter $0 < s < 1$, initial bandwidth b_0, and parameter c.
Output: Bandwidth matrix $\mathbf{B}$.
Initialize all $b_1, \cdots, b_{N^k}$ with b_0;
for each $j = 1$ to N^k **do**
  do
  Estimate derivative $\frac{\partial f(\mathbf{x}^k)}{\partial b_j}$ and variance $\mathrm{Var}(\frac{\partial f(\mathbf{x}^k)}{\partial b_j})$;
  Compute $\tau_j = \sqrt{2 \cdot \mathrm{Var}(\frac{\partial f(\mathbf{x}^k)}{\partial b_j}) \cdot \log(cN^k)}$;
  if $|\frac{\partial f(\mathbf{x}^k)}{\partial b_j}| > \tau_j$, **then** Update $b_j = b_j s$;
  while $|\frac{\partial f(\mathbf{x}^k)}{\partial b_j}| > \tau_j$
end for
Return B.

According to the estimated bandwidth $\mathbf{B}$ by Algorithm 1, we can calculate density $f(\mathbf{x}^k)$ of $\mathbf{x}^k$ in Eq. (3). The perturbation process is to maximize the following attack loss $\mathcal{L}_A$ for producing unnoticeable perturbations, in terms of the estimations $f(\mathbf{x}^1)$ and $f(\mathbf{x}^2)$ in two KGs G^1 and G^2.

$$\max_{\hat{A}^k} \mathcal{L}_A = \left[\sum_{(e_i^1, e_j^2) \in D} -\log \sigma\left((\hat{e}_i^1)^T \cdot \hat{e}_j^2\right) + f(\hat{A}_i^1) + f(\hat{A}_j^2) \right] + \sum_{(e_{i'}^1, e_{j'}^2) \notin D} \log \sigma\left((e_{i'}^1)^T \cdot \mathbf{v}_{j'}^2\right) \tag{2.10}$$

$$\text{s.t. } |\hat{A}_i^k - A_i^k| \leq \Delta,\ 1 \leq k \leq 2$$

where $\hat{A}_i^1 = A_i^1 + \delta_i^1$ (and $\hat{A}_j^2 = A_j^2 + \delta_j^2$) denote perturbations of clean structure features A_i^1 (and A_j^2) in G^1 (and G^2) by adding a small amount of relation perturbations δ_i^1 (and δ_j^2), such that $\hat{e}_i^1$ is far away from $\hat{e}_j^2$ and thus the alignment accuracy is decreased. In addition, we push e_i^1 and e_j^2 to dense regions to generate $\hat{e}_i^1$ and $\hat{e}_j^2$, by maximizing $f(\hat{A}_i^1)$ and $f(\hat{A}_j^2)$, such that $\hat{e}_i^1$ and $\hat{e}_j^2$ are indistinguishable from their neighbors in perturbed KGs. This reduces the possibility of perturbation detection by humans or defender programs.

We leverage the Projected Gradient Descent (PGD) technique [166] to produce perturbed adjacency matrices $\hat{A}^1$ and $\hat{A}^2$ of two KGs G^1 and G^2.

$$\begin{aligned} (A_i^1)^{(t+1)} &= \Pi_{\Delta_1} \text{sgn}[\text{ReLU}(\nabla_{(A_i^1)_t} \mathcal{L}_A)] \\ (A_j^2)^{(t+1)} &= \Pi_{\Delta_2} \text{sgn}[\text{ReLU}(\nabla_{(A_j^2)_t} \mathcal{L}_A)] \end{aligned} \quad t = 1, \cdots, T \tag{2.11}$$

where $(A_i^1)^{(t+1)}$ and $(A_j^2)^{(t+1)}$ denote the perturbations of A_i^1 and A_j^2 derived at step t. ϵ specifies the budget of allowed perturbed relations for each attacked entity. $\Delta^k = \{(\delta^k)^t | \mathbf{1}^T (\delta^k)^t \leq \epsilon, (\delta^k)^t \in \{0,1\}^{N^k}\}$, where $(\delta^k)^t = \|(A_i^1)^t - A_i^1\|_2$, represents the constraint set of the projection operator Π, i.e., it encodes whether a relation in A_i^1 is modified or not. The composition of the ReLU and sign operators guarantees $(A_i^1)^t \in \{0,1\}^{N^1}$ and $(A_j^2)^t \in \{0,1\}^{N^2}$, as it adds (or removes) a relation or keeps it unchanged when a derivative in the gradient is positive (or negative). The outputs $(A_i^1)^T$ and $(A_j^2)^T$ at final step T are used as the perturbed adjacency matrices $\hat{A}_i^1$ and $\hat{A}_j^2$.

2.3 Effective Adversarial Attacks

Unfortunately, the above PGD-based unnoticeable attack method needs to iteratively calculate the gradient $\nabla_{(A_i^1)}\mathcal{L}_A$, which mainly depends on $\frac{\partial\left(\log\sigma((\mathbf{e}_i^1)^T\cdot\mathbf{e}_j^2)\right)}{\partial\mathbf{A}_i^1}$ in the GCN-based entity alignment models.

Given an alignment signal $\phi((\mathbf{e}_i^1)^T,\mathbf{e}_j^2)=\frac{\partial\left(\log\sigma((\mathbf{e}_i^1)^T\cdot\mathbf{e}_j^2)\right)}{\partial(\mathbf{e}_i^1)^T}$ and a Jacobian matrix $\mathbf{J}_i=\frac{\partial(\mathbf{e}_i^1)^T}{\partial\mathbf{A}_i^1}$, the gradient of $\log\sigma((\mathbf{e}_i^1)^T\cdot\mathbf{e}_j^2)$ is calculated as follows.

$$\frac{\partial\left(\log\sigma((\mathbf{e}_i^1)^T\cdot\mathbf{e}_j^2)\right)}{\partial\mathbf{A}_i^1}=\frac{\partial\left(\log\sigma((\mathbf{e}_i^1)^T\cdot\mathbf{e}_j^2)\right)}{\partial(\mathbf{e}_i^1)^T}\frac{\partial(\mathbf{e}_i^1)^T}{\partial\mathbf{A}_i^1} \tag{10}$$

$$=\phi((\mathbf{e}_i^1)^T,\mathbf{e}_j^2)\mathbf{J}_i \tag{2.12}$$

It is obvious that the gradient is determined with both the signal and the Jacobian together. The situation that either the signal has saturating gradient or the Jacobian is insignificant is able to result in vanishing gradients in $\frac{\partial\left(\log\sigma((\mathbf{e}_i^1)^T\cdot\mathbf{e}_j^2)\right)}{\partial\mathbf{A}_i^1}$ and thus the attack failures.

All singular values of a neural network's input-output Jacobian matrix concentrate near 1 is a property known as dynamical isometry (Pennington et al., 2017). Ensuring the mean squared singular value of a network's input-output Jacobian is $O(1)$ is essential for avoiding the exponential vanishing or explosion of gradients. We leverage the dynamical isometry theory for improving the effectiveness of the PGD adversarial attacks. Concretely, a neural network is dynamical isometry if all singular values λ_{ir} of the Jacobian $\mathbf{J}_i$ are close to 1, i.e., $1-\lambda_{ir}\leq\xi$ for $\forall r, r\in\{1,\cdots,\min\{N^1,N^2\}\}$ and a small positive number $\xi\approx 0$. In our problem, when the Jacobian matrix $\mathbf{J}_i$ is dynamical isometry, the signal $\phi((\mathbf{e}_i^1)^T,\mathbf{e}_j^2)$ backpropagates isometrically over the neural network and maintains the norm and all angles between vectors.

Intuitively, if we select a good attack signal amplification factor α to amplify $\mathbf{e}_i^1$ and $\mathbf{e}_j^2$ as follows, then this can improve the diffusion of attack signals. In addition, a good α should

guarantee the relative order of the network's output logits invariant, to ensure the decision boundary of entity alignment unchanged.

$$\tilde{\mathbf{e}}_i^1 = \alpha \mathbf{e}_i^1, \tilde{\mathbf{e}}_j^2 = \alpha \mathbf{e}_j^2 \tag{11}$$

We rewrite the gradients with α as follows.

$$\begin{aligned} \frac{\partial \left(\log \sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2)\right)}{\partial \mathbf{A}_i^1} &= \frac{\partial \left(\log \sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2)\right)}{\partial (\tilde{\mathbf{e}}_i^1)^T} \frac{\partial (\tilde{\mathbf{e}}_i^1)^T}{\partial (\mathbf{e}_i^1)^T} \frac{\partial (\mathbf{e}_i^1)^T}{\partial \mathbf{A}_i^1} \qquad (12) \\ &= \phi((\tilde{\mathbf{e}}_i^1)^T, \tilde{\mathbf{e}}_j^2) \alpha \mathbf{J}_i \qquad (2.13) \end{aligned}$$

Notice that $\phi((\tilde{\mathbf{e}}_i^1)^T, \tilde{\mathbf{e}}_j^2) = \frac{\sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2)(1-\sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2))\tilde{\mathbf{e}}_j^2}{\sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2)} = (1 - \sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2))\tilde{\mathbf{e}}_j^2$. When α is close to ∞, the alignment signal $\phi((\tilde{\mathbf{e}}_i^1)^T, \tilde{\mathbf{e}}_j^2)$ approaches zero and thus the vanishing gradient problem is encountered in adversarial attacks. In addition, all singular values of $\alpha \mathbf{J}_i$ are equal to zeros if $\alpha = 0$. $\frac{\partial \left(\log \sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2)\right)}{\partial \mathbf{A}_i^1}$ is equal to zero, which leads to the vanishing gradient problem too.

Therefore, a desired α for avoiding the exponential vanishing of gradients should stand in between 0 and ∞, in order to guarantee the signal $\phi((\tilde{\mathbf{e}}_i^1)^T, \tilde{\mathbf{e}}_j^2)$ large enough, i.e., $\|\phi((\tilde{\mathbf{e}}_i^1)^T, \tilde{\mathbf{e}}_j^2)\|_2 > \eta$ for a positive threshold η, as well as make all singular values of $\alpha \mathbf{J}_i$ close to 1, such that the signal $\phi((\tilde{\mathbf{e}}_i^1)^T, \tilde{\mathbf{e}}_j^2)$ can be well backpropagated from the output layer to the input layer.

In order to make the mean of singular values of $\alpha \mathbf{J}_i$ close to 1, the first option of α is the inverse of the mean of singular values of $\mathbf{J}_i$.

$$\alpha = \frac{|D|N}{\sum_{i=1}^{|D|} \sum_{r=1}^{N} \lambda_{ir}} \tag{13}$$

where λ_{ir} is the r^{th} singular value of $\mathbf{J}_i$. $|D|$ is the size of the set D of pre-aligned entity pairs and $N = \min\{N^1, N^2\}$.

For the purpose of ensuring $\|\phi((\tilde{\mathbf{e}}_i^1)^T, \tilde{\mathbf{e}}_j^2)\|_2 > \eta$, the second option of α should be satisfied with $1 - \sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2) > \eta / \|\tilde{\mathbf{e}}_j^2\|_2$. The feasible α can be obtained through the following theorem.

Theorem 2.1. *Let entity embedding vectors $\tilde{\mathbf{e}}_k^2$ and $\tilde{\mathbf{e}}_l^2$ be the most similar and least similar to $(\tilde{\mathbf{e}}_i^1)^T$ $(1 \leq k, l \leq N^2)$, i.e., $\tilde{\mathbf{e}}_k^2 = \arg\max_{\tilde{\mathbf{e}}_k^2} (\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_k^2$ and $\tilde{\mathbf{e}}_l^2 = \arg\min_{\tilde{\mathbf{e}}_l^2} (\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_l^2$, and $c = (\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_k^2$. Also, suppose that d is the minimal norm of entity embedding vectors in G^2, i.e., $d = \min_{\tilde{\mathbf{e}}_m^2} \|\tilde{\mathbf{e}}_m^2\|_2$ for $\forall e_m^2 \in E^2$. For a given $0 < \eta < d/2$, if $\alpha < \sqrt{\frac{1}{c} \log \frac{d-\eta}{\eta}}$, then $1 - \sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2) > \eta / \|\tilde{\mathbf{e}}_j^2\|_2$ for $\forall e_j^2 \in E^2$.*

Proof. $1 - \sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2) > \eta / \|\tilde{\mathbf{e}}_j^2\|_2$ is equivalent to $\sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2) < 1 - \eta / \|\tilde{\mathbf{e}}_j^2\|_2$. We convert it to $\frac{1}{1+\exp\left(-(\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2\right)} < 1 - \eta / \|\tilde{\mathbf{e}}_j^2\|_2$. As $(\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2 \leq c$, we have $\frac{1}{1+\exp\left(-\alpha^2 (\mathbf{e}_i^1)^T \cdot \mathbf{e}_j^2\right)} \leq \frac{1}{1+\exp(-\alpha^2 c)}$. If we can prove $\frac{1}{1+\exp(-\alpha^2 c)} < 1 - \eta / \|\tilde{\mathbf{e}}_j^2\|_2$, then we can testify $\frac{1}{1+\exp\left(-\alpha^2 (\mathbf{e}_i^1)^T \cdot \mathbf{e}_j^2\right)} < 1 - \eta / \|\tilde{\mathbf{e}}_j^2\|_2$. Thus, we need to solve $\exp(\alpha^2 c) < \frac{\|\tilde{\mathbf{e}}_j^2\|_2 - \eta}{\eta}$.

As $\|\tilde{\mathbf{e}}_j^2\|_2 \geq d$, feasible α for $\exp(\alpha^2 c) < \frac{d-\eta}{\eta}$ is also feasible for $\exp(\alpha^2 c) < \frac{\|\tilde{\mathbf{e}}_j^2\|_2 - \eta}{\eta}$. Since exp is a monotonic increasing function, by solving the above inequality, we have feasible $\alpha < \sqrt{\frac{1}{c} \log \frac{d-\eta}{\eta}}$.

Notice that $0 < \eta < d/2$. This makes $\frac{d-\eta}{\eta} > 1$ and the upper bound of α be positive. Therefore, for any $\alpha < \sqrt{\frac{1}{c} \log \frac{d-\eta}{\eta}}$, $1 - \sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2) > \eta / \|\tilde{\mathbf{e}}_j^2\|_2$ is satisfied. □

Algorithm 2 combines the above two kinds of α to produce effective adversarial attacks with attack signal amplification. The perturbed entity embeddings $\hat{\mathbf{e}}_i^1$ and $\hat{\mathbf{e}}_j^2$ are initialized with clean ones $\mathbf{e}_i^1$ and $\mathbf{e}_j^2$ in step 2. The first amplification factor α_1 is calculated in step 3. The second factor α_2 is computed in steps 5-7. α_1 and α_2 are integrated together for enhancing the attack signal propagation of neural networks in steps 8-9. The PGD attack method with attack signal amplification is utilized to perturb the KGs. The algorithm repeats the above iterative procedure until convergence.

Algorithm 2: Effective Adversarial Attacks

Input: KG $G^k = (E^k, R^k, T^k)$, set of pre-aligned entity pairs $D = \{(e_i^1, e_j^2)|e_i^1 \leftrightarrow e_j^2\}$, trained entity embedding model h, noise budget ϵ, and signal threshold η.
Output: Perturbed adjacency matrices $\{\hat{\mathbf{A}}_i^1, \hat{\mathbf{A}}_j^2|(e_i^1, e_j^2) \in D\}$.
for each pair (e_i^1, e_j^2) in D **do**
  Set $\hat{\mathbf{e}}_i^1 = \mathbf{e}_i^1 = h(e_i^1)$, $\hat{\mathbf{e}}_j^2 = \mathbf{e}_j^2 = h(e_j^2)$;
  Compute $\alpha_1 = \frac{|D|N}{\sum_{i=1}^{|D|}\sum_{r=1}^{N}\lambda_{ir}}$ in Eq.(13);
  for $t = 1, \cdots, T$ **do**
    Initialize $\alpha_2 = 1.0$;
    if $1 - \sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2) \leq \eta / \|\tilde{\mathbf{e}}_j^2\|_2$ **then**
      Update $\alpha_2 = \sqrt{\frac{1}{c}\log\frac{d-\eta}{\eta}}$ in Theorem 1;
    end if
    Amplify $\tilde{\mathbf{e}}_i^1 = \alpha_1\alpha_2\mathbf{e}_i^1$, $\tilde{\mathbf{e}}_j^2 = \alpha_1\alpha_2\mathbf{e}_j^2$;
    Calculate $\frac{\partial\left(\log\sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2)\right)}{\partial \mathbf{A}_i^1}$ and $\frac{\partial\left(\log\sigma((\tilde{\mathbf{e}}_i^1)^T \cdot \tilde{\mathbf{e}}_j^2)\right)}{\partial \mathbf{A}_j^2}$;
    Use the PGD to update $\hat{\mathbf{A}}_i^1$, $\hat{\mathbf{A}}_j^2$ in Eq.(9);
  end for
end for
Return $\{\hat{\mathbf{A}}_i^1, \hat{\mathbf{A}}_j^2|(e_i^1, e_j^2) \in D\}$.

2.4 Experiments

2.4.1 Experimental Setup

Dataset	#Entities	#Relations	#Triples	#Alignments
ZH-EN	ZH: 66,469 EN: 98,125	ZH: 2,830 EN: 2,317	ZH: 153,929 EN: 237,674	15,000
JA-EN	JA: 65,744 EN: 95,680	JA: 2,043 EN: 2,096	JA: 164,373 EN: 233,319	15,000
FR-EN	FA: 66,858 EN: 105,889	FA: 1,379 EN: 2,209	FA: 192,191 EN: 278,590	15,000

Table 2.1: Statistics of Datasets

Table 2.1 presents the statistics of the DBP15K datasets [250]. They consist of three different cross-lingual datasets, which are $DBP15K_{ZH-EN}$, $DBP15K_{JA-EN}$, and $DBP15K_{FR-EN}$. Each cross-lingual dataset contains two monolingual KGs in different languages and 15,000 pre-aligned entity pairs between the two KGs. In the experiment, 30% of the pre-aligned entity pairs are used for training data, and the remaining pairs are used for test data.

We compare the EAA model with seven state-of-the-art attack models. **Sememe-based Word Substitution (SWS)** incorporates the sememe-based word substitution and swarm optimization-based search to conduct word-level attacks [305]. **Inflection Word Swap (IWS)** perturbs the inflectional morphology of words to craft plausible and semantically similar adversarial examples [255, 179]. We utilize the above two word-level attack models to replace associated entities of a relation based on semantics. **GF-Attack** attacks graph embedding methods by devising new loss and approximating the spectrum [46]. **LowBlow** is a general low-rank adversarial attack model which is able to affect the performance of various graph learning tasks [80]. We use the above two graph attack models to directly add/remove relations in terms of graph topology. **CRIAGE** aims to add/remove the facts to/from the KG that degrades the performance of link prediction [202]. **DPA** contains a collection of data poisoning attack strategies against knowledge graph embedding [309]. **RL-RR** uses reinforcement learning policy to produce deceptively perturbed KGs while keeping the downstream quality of the original KG [212]. To our best knowledge, this work is the first to study adversarial attacks on cross-lingual entity alignment.

We evaluate four versions of EAA to show the strengths of different components. **EAA-P** uses the basic PGD [166] to produce adversarial attacks. **EAA-D** only utilizes the KDE and density maximization to generate effective and unnoticeable attacks. **EAA-A** employs only our attack signal amplification strategy to improve the performance of the basic PGD attack. **EAA** operates with the full support of both KDE and signal amplification components.

We validate the effectiveness of the above attack models with three representative cross-lingual entity alignment algorithms. **AttrGNN** integrates both attribute and relation triples for better performance of cross-lingual entity alignment [162]. **RNM** is a novel relation-aware neighborhood matching model for entity alignment [365]. To our best knowledge, **REA** is the only robust cross-lingual entity alignment solution against adversarial attacks by detecting noise in the perturbed inter-KG entity links [197].

We use two popular metrics in entity alignment to verify the attack effectiveness: *Hits@k* (i.e., the ratio of correctly aligned entities ranked in the top k candidates) and *MRR* (i.e., mean reciprocal rank). A smaller *Hits@k* or *MRR* indicates worse entity alignment but a better attack. K is fixed to 1 in all tests.

2.4.2 Results

Attacks	**AttrGNN**		**RNM**		**REA**	
	Hits@1	**MRR**	**Hits@1**	**MRR**	**Hits@1**	**MRR**
Clean	0.796	0.845	0.841	0.875	0.792	0.818
SWS	0.726	0.839	0.745	0.862	0.764	0.848
IWS	0.708	0.761	0.729	0.823	0.759	0.804
GF-Attack	0.709	0.815	0.724	0.833	0.733	0.844
LowBlow	0.677	0.773	0.678	0.776	0.697	0.797
CRIAGE	0.646	0.704	0.655	0.719	0.662	0.715
DPA	0.603	0.712	0.636	0.751	0.635	0.733
RL-RR	0.562	0.684	0.628	0.713	0.637	0.722
EAA	**0.497**	**0.538**	**0.525**	**0.636**	**0.538**	**0.641**

Table 2.2: Results on DBP15K$_{ZH-EN}$ with 5% perturbed relations.

Attacks	**AttrGNN**		**RNM**		**REA**	
	Hits@1	**MRR**	**Hits@1**	**MRR**	**Hits@1**	**MRR**
Clean	0.783	0.834	0.872	0.899	0.799	0.823
SWS	0.724	0.839	0.774	0.854	0.788	0.843
IWS	0.718	0.787	0.755	0.804	0.745	0.796
GF-Attack	0.715	0.824	0.747	0.826	0.767	0.845
LowBlow	0.737	0.783	0.728	0.800	0.723	0.821
CRIAGE	0.705	0.756	0.699	0.769	0.707	0.769
DPA	0.643	0.725	0.723	0.753	0.669	0.766
RL-RR	0.689	0.716	0.691	0.765	0.706	0.768
EAA	**0.579**	**0.612**	**0.618**	**0.642**	**0.621**	**0.652**

Table 2.3: Results on DBP15K$_{JA-EN}$ with 5% perturbed relations.

Attacks	AttrGNN		RNM		REA	
	Hits@1	**MRR**	**Hits@1**	**MRR**	**Hits@1**	**MRR**
Clean	0.919	0.910	0.938	0.954	0.812	0.855
SWS	0.782	0.873	0.814	0.886	0.807	0.846
IWS	0.755	0.801	0.803	0.836	0.802	0.806
GF-Attack	0.715	0.828	0.779	0.848	0.792	0.848
LowBlow	0.792	0.841	0.799	0.826	0.793	0.852
CRIAGE	0.733	0.864	0.744	0.873	0.781	0.831
DPA	0.704	0.757	0.796	0.817	0.695	0.791
RL-RR	0.754	0.792	0.745	0.823	0.754	0.784
EAA	**0.643**	**0.697**	**0.644**	**0.709**	**0.681**	**0.696**

Table 2.4: Results on DBP15K$_{FR-EN}$ with 5% perturbed relations.

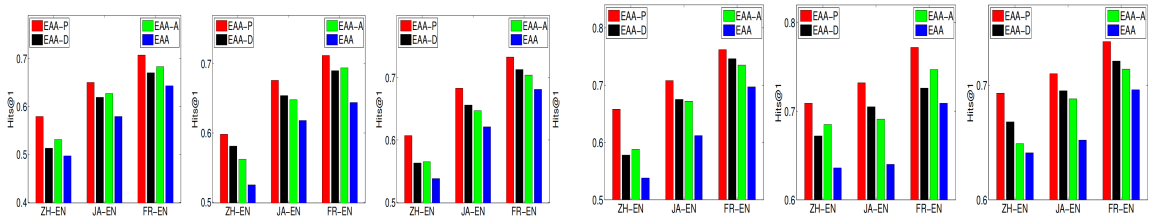
Attack Performance on Various Datasets with Different Entity Alignment Algorithms

Table 2.2-2.4 exhibit the *Hits@1* and *MRR* scores of three GCN-based entity alignment algorithms on test data by nine attack models over three groups of cross-lingual datasets. *Clean* represents that the experiments run on the original KGs without any perturbations. For all other attack models, the number of perturbed relations is fixed to 5% in these experiments. It is observed that among nine attack methods, no matter how strong the attacks are, the **EAA** method achieves the lowest *Hits@1* and *MRR* scores on perturbed KGs in most experiments, showing the effectiveness of **EAA** for the adversarial attacks.

Compared to the entity alignment results under other attack models, **EAA**, on average, achieves 17.7%, 12.8%, and 12.8% improvement of *Hits@1* and 17.6%, 16.9%, and 13.7% boost of *MRR* on *DBP15K*$_{ZH-EN}$, *DBP15K*$_{JA-EN}$, and *DBP15K*$_{FR-EN}$ respectively. In addition, the promising performance of **EAA** with all three entity alignment models implies that **EAA** has great potential as a general attack solution to other entity alignment methods, which is desirable in practice.

Ablation Study

Figures 2.2(a) and 2.2(b) present the *Hits@1* and *MRR* scores achieved by three entity alignment methods under adversarial attacks with four variants of our **EAA** attack model.



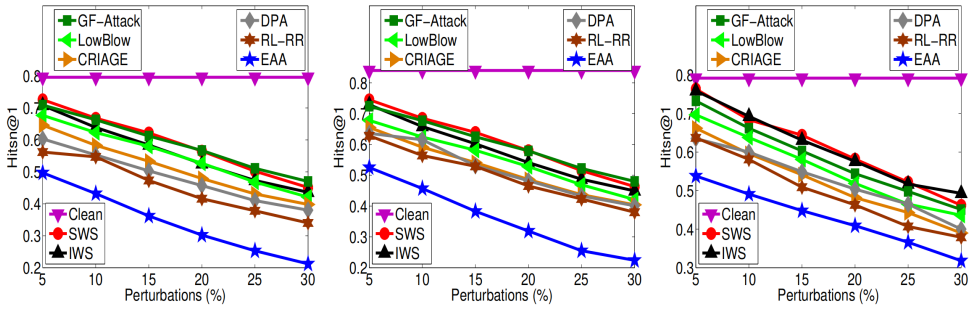
(a) Hit@1 (b) MRR

Figure 2.2: Hit@1 and MRR of EAA Variants

We have observed the complete **EAA** achieves the lowest *Hits@1* (< 0.681) and the smallest *MRR* scores (< 0.709) respectively, which are obviously better than other versions. Notice that **EAA-A** achieves the better attack performance than **EAA-P** in most tests. A reasonable explanation is that our attack signal amplification technique is able to alleviate the vanishing gradient issue, which effectively helps maintain the utility of adversarial attacks in GCN-based entity alignment models.

In addition, **EAA-D** also performs well in most experiments, compared with **EAA-P**. A rational guess is that it is difficult to correctly match the entities in two KGs when they lie in dense regions with many similar entities. These results illustrate both KDE and signal amplification methods are important in producing effective and unnoticeable attacks in entity alignment.

Attack Performance with Varying Perturbed Relations

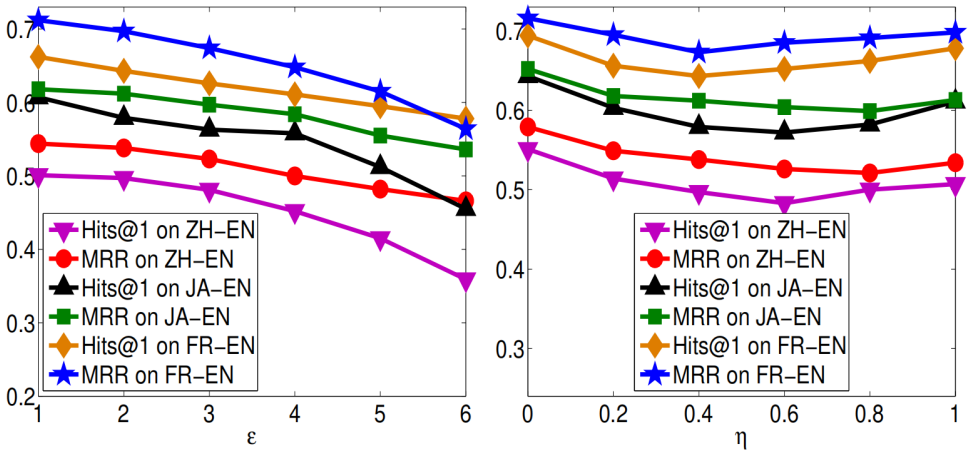


(a) AttrGNN on ZH-EN (b) RNM on ZH-EN (c) REA on ZH-EN

Figure 2.3: Hits@1 with varying perturbed relations

Figure 2.3 presents the performance of entity alignment under nine attack models by varying the ratios of perturbed edges from 5% to 30%. It is obvious that the attacking performance improves for each attacker with an increase in the number of perturbed edges. This phenomenon indicates that current GCN-based entity alignment methods are very sensitive to adversarial attacks. **EAA** achieves the lowest *Hits@1* values (< 0.538), which are still better than the other eight methods in most tests. Especially, when the perturbation ratio is large than 10%, the *Hits@1* values drop quickly.

Attack Performance with Varying Perturbed Relations



(a) Perturbation budget ϵ (b) Signal threshold η

Figure 2.4: Results with varying parameters

Figure 2.4(a) measures the performance effect of ϵ in the **EAA** model for the entity alignment by varying ϵ from 1 to 6. It is observed that when increasing ϵ, both *Hits@1* and *MRR* scores of the **EAA** model decrease substantially. This demonstrates it is difficult to train a robust entity alignment model under large ϵ constraint. However, a large ϵ can be easily detected by humans or by defender programs. Notice that the average number of associated relations of each entity in three datasets is between 2.3 and 2.9. Thus, we suggest generating both effective and unnoticeable attacks for the entity alignment task under ϵ between 2 and 3, such that ϵ is smaller than the average number of associated relations.

2.5 Conclusion

We have studied the problem of adversarial attacks against cross-lingual entity alignment. First, we proposed to utilize kernel density estimation technique to estimate and maximize the densities of attacked entities and generate effective and unnoticeable perturbations, by pushing attacked entities to dense regions in two KGs. Second, we analyze how gradient vanishing causes failures of gradient-based adversarial attacks. We design an attack signal amplification method to ensure informative signal propagation. The EAA model achieves superior performance against representative attack models.

Chapter 3

Robust Network Alignment via Attack Signal Scaling and Adversarial Perturbation Elimination

3.1 Graph Alignment

In this work, we aim to learn a preemptive protection model for existing network alignment algorithms, enhanced with the guidance of effective adversarial attacks. Given one

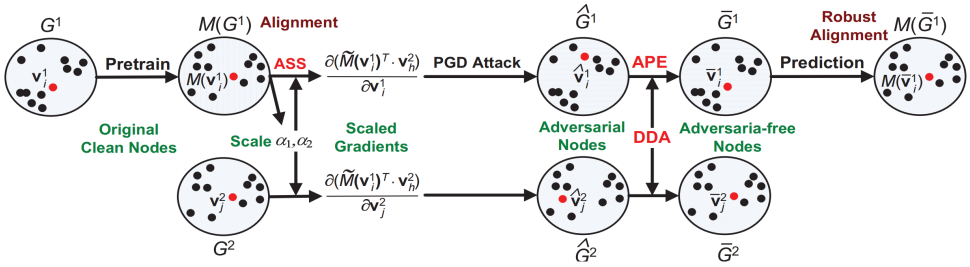


Figure 3.1: Adversarial Attacks with Signal Scaling and Adversarial Perturbation Elimination for Robust Network Alignment

source network $G^1 = (V^1, E^1)$ and one target network $G^2 = (V^2, E^2)$ to be aligned, each network is denoted as $G^t = (V^t, E^t)$ ($t = 1$ or 2), where $V^t = \{v_1^t, \cdots, v_{N^t}^t\}$ is the set of N^t nodes and $E^t = \{(v_i^t, v_k^t) : 1 \leq i, k \leq N^t\}$ is the set of edges. A node $v_i^t \in V^t$ ($1 \leq i \leq N^t$) represents an entity in G^t. An edge $(v_i^t, v_k^t) \in E^t$ is associated with two nodes $v_i^t \in V^t$ and $v_k^t \in V^t$ and denotes the relationship between two corresponding entities.

Each G^t has an $N^t \times N^t$ binary adjacency matrix A^t, where each entry $A_{ik}^t = 1$ if there exists an edge $(v_i^t, v_k^t) \in E^t$; otherwise $A_{ik}^t = 0$. A_i^t specifies the ith row vector of A^t. In this paper, if there are no specific descriptions, we use v_i^t to denote a node v_i^t itself and its representation A_i^t, i.e., $v_i^t = A_i^t$, and we utilize v_{ik}^t to specify the kth dimension of v_i^t, i.e., $v_{ik}^t = A_{ik}^t$.

The dataset is divided into two disjoint sets D and D'. The former denotes a set of known aligned node pairs $D = \{(v_i^1, v_j^2)|v_i^1 \leftrightarrow v_j^2, v_i^1 \in V^1, v_j^2 \in V^2\}$, where $v_i^1 \leftrightarrow v_j^2$ indicates that two nodes v_i^1 and v_j^2 belong to the same entity. The latter, denoted by $D' = \{(v_i^1, v_j^2)|v_i^1 \leftrightarrow v_j^2, v_i^1 \in V^1, v_j^2 \in V^2\}$, is used to evaluate the network alignment performance, where the nodes (but not their alignments) are also observed during training. The goal of supervised network alignment is to use D as the training data to identify the one-to-one matching relationships between nodes v_i^1 and v_j^2 belonging to the same entities in the test data D'.

Many supervised learning methods learn effective network alignment algorithms by maximizing the similarities (or minimizing the distances) between projected source anchor nodes $M(v_i^1) \in D$ and target ones $v_j^2 \in D$ [142, 143, 299, 325]. The node pairs $(v_i^1, v_j^2) \in D'$ with the largest similarities are selected as the alignment results. The following loss function is minimized to learn an injective one-to-one matching function $M : v_i^1 \in V^1 \mapsto v_j^2 \in V^2$. In deep network alignment approaches [65, 87, 143, 325], M is often implemented as a neural network.

$$\mathcal{L}(v_i^1, v_j^2) = -\log\sigma\left(M(v_i^1)^T \cdot v_j^2\right) + \sum_{k=1}^{K} \mathbb{E}_{v_k^2 \sim p(v_k^2)} \log\sigma\left(M(v_i^1)^T \cdot v_k^2\right) \tag{3.1}$$

$$\min_M \mathcal{L} = \mathbb{E}_{(v_i^1, v_j^2) \in D} \mathcal{L}(v_i^1, v_j^2) \tag{3.2}$$

where $M(v_i^1)^T$ is the transpose of $M(v_i^1)$. $p(v_k^2)$ denotes the distribution for sampling K negative nodes $v_k^2 \neq v_j^2$ through the negative sampling method [173]. $\sigma(\cdot)$ is the sigmoid function. The inner product $\cdot$ represents the similarity degree between two node vectors. The above loss is equivalent to a cross-entropy loss with $(v_i^1, v_j^2) \in D$ as positive samples and $(v_i^1, v_k^2) \notin D$ as negative ones.

Given a trained network alignment method $v_j^2 = M(v_i^1)$, an adversarial attacker aims to maximally degrade the alignment performance of M on the test data D' by injecting

edge perturbations (including edge insertion and deletion) into $G^t = (V^t, E^t)$ ($t = 1$ or 2), leading to two adversarial networks $\hat{G}^t = (\hat{V}^t, \hat{E}^t)$. In order to generate effective adversarial perturbations for better training adversarial perturbation elimination, we assume that the attacker can access the prediction result and gradient information of M.

In contrast, with the generated adversarial perturbations as the supervision, an adversarial defender is trained to be able to eliminate the perturbations before feeding it into M for providing the preemptive protection. A desired perturbation elimination result should ensure that the model achieves the high utility of network alignment on the newly perturbed networks $\hat{G}^1$ and $\hat{G}^2$.

This work proposes a robust network alignment solution that contains two analytics components: (1) Adversarial attacks with attack signal scaling and (2) Adversarial perturbation elimination via Dirac delta approximation, as shown in Figure 1. Here, given two networks G^1 and G^2, two nodes within each network are close/distant if they have similar/dissimilar structural features. Two red dots denote a pair of aligned nodes $\mathbf{v}_i^1$ and $\mathbf{v}_j^2$ in G^1 and G^2. The red dots are target nodes to be attacked.

(1) **Adversarial attacks with attack signal scaling** model attempts to generate the effective adversarial nodes in G^1 and G^2 that can easily fool the network alignment algorithm M trained on clean G^1 and G^2, and thus output wrong alignment results: (a) An attack signal scaling (ASS) method based on the dynamical isometry theory is proposed to compute signal scales α_1 and α_2 by using Eq.(6) and Theorem 5.1, in order to ensure informative attack signal with both well-conditioned Jacobian and meaningful signal propagation from the alignment loss in Eq.(1); (b) By integrating scaled signal $\tilde{M}(\mathbf{v}_i^1) = \alpha_1\alpha_2 M(\mathbf{v}_i^1)$ and gradients $\frac{\partial\left(\tilde{M}(\hat{\mathbf{v}}_i^1)^T \cdot \hat{\mathbf{v}}_h^2\right)}{\partial \hat{\mathbf{v}}_i^1}$ and $\frac{\partial\left(\tilde{M}(\hat{\mathbf{v}}_i^1)^T \cdot \hat{\mathbf{v}}_h^2\right)}{\partial \hat{\mathbf{v}}_j^2}$, the Projected Gradient Descent (PGD) is utilized to add and remove noisy edges to $\mathbf{v}_i^1$ and $\mathbf{v}_j^2$ in terms of Eq.(2) and Algorithm 1. The attacker moves $\mathbf{v}_i^1$ and $\mathbf{v}_j^2$ and derives adversarial nodes $\hat{\mathbf{v}}_i^1$ and $\hat{\mathbf{v}}_j^2$, such that the similarity $\log\sigma\left(M(\hat{\mathbf{v}}_i^1)^T \cdot \mathbf{v}_k^2\right) > \log\sigma\left(M(\hat{\mathbf{v}}_i^1)^T \cdot \mathbf{v}_j^2\right)$, where $\mathbf{v}_k^2 \neq \mathbf{v}_j^2$ is a negative node, and thus a wrong alignment $\mathbf{v}_k^2 = M(\hat{\mathbf{v}}_i^1)$ is produced.

(2) **Adversarial perturbation elimination via Dirac delta approximation** tries to rule out the negative effects of adversarial nodes and improve the robustness on the perturbed networks: (a) An adversarial perturbation elimination (APE) model with the LSTM models is proposed to neutralize adversarial nodes $\hat{\mathbf{v}}_i^1$ (or $\hat{\mathbf{v}}_j^2$) in vulnerable space to adversarial-free nodes $\bar{\mathbf{v}}_i^1$ (or $\bar{\mathbf{v}}_j^2$) in safe area in Eq.(8); (b) A Dirac delta approximation (DDA) technique is designed to make $\bar{\mathbf{v}}_i^1$ (or $\bar{\mathbf{v}}_j^2$) be close to $\mathbf{v}_i^1$ (or $\mathbf{v}_j^2$) as much as possible, as well as cause the defense loss $\mathcal{L}_{\mathcal{D}}$ on adversarial-free nodes in Eq.(7) to be identical to the original network alignment loss $\mathcal{L}$ on clean nodes in Eq.(1); (c) The adversarial-free $\bar{\mathbf{v}}_i^1$ and $\bar{\mathbf{v}}_j^2$ are fed into the trained M to output the network alignment results.

3.2 Adversarial Attacks With Attack Signal Scaling

In this section, we will analyze how poor signal propagation can cause vanishing gradients in iterative gradient-based adversarial attacks, and then propose an attack signal scaling (ASS) method based on the dynamical isometry theory to scale attack signal in back-propagation, to ensure informative attack signal with both well-conditioned Jacobian and meaningful signal propagation from the alignment loss.

3.3 PGD-based Adversarial Attacks

Based on the alignment loss in Eq. (1), we propose to utilize the Projected Gradient Descent (PGD) method to produce adversarial nodes towards network alignment.

$$\begin{aligned}(v_i^1)^{(s+1)} &= \Pi_{\Delta_i^1}\mathrm{sgn}\left[\mathrm{ReLU}\left(\nabla_{(v_i^1)^s}\mathcal{L}((v_i^1)^s,(v_j^2)^s)\right)\right],\\ (v_j^2)^{(s+1)} &= \Pi_{\Delta_j^2}\mathrm{sgn}\left[\mathrm{ReLU}\left(\nabla_{(v_j^2)^s}\mathcal{L}((v_i^1)^s,(v_j^2)^s)\right)\right],\end{aligned} \tag{3.3}$$

where $s = 1, \cdots, S$.

where $(v_i^1)^s$ and $(v_j^2)^s$ denote the adversarial nodes of v_i^1 and v_j^2 derived at step s. ϵ specifies the budget of allowed perturbed edges for each attacked node. $\Delta_i^1 = \{(\delta_i^1)^s | (\delta_i^1)^s = (v_i^1)^s - v_i^1, \|(\delta_i^1)^s\|_1 \leq \epsilon\}$ represents the constraint set of the projection operator Π, i.e., it

encodes whether an edge of v_i^1 is modified or not. Δ_j^2 has the similar definition for v_j^2. The composition of the ReLU and sign operators guarantees $(v_i^1)^s \in \{0,1\}^{N^1}$ and $(v_j^2)^s \in \{0,1\}^{N^2}$, as it adds (or removes) an edge or keeps it unchanged when a derivative in the gradient is positive (or negative). The outputs $(v_i^1)^S$ and $(v_j^2)^S$ at final step S are used as the adversarial nodes $\hat{v}_i^1 = (v_i^1)^S$ and $\hat{v}_j^2 = (v_j^2)^S$. Recall the alignment loss in Eq. (1), the gradient $\nabla_{v_i^1}\mathcal{L}(v_i^1, v_j^2)$ in the PGD is mainly determined with

$$\frac{\partial \log \sigma(M(v_i^1)^T \cdot v_h^2)}{\partial v_i^1},$$

where v_h^2 can be either a positive sample v_j^2 or a negative one v_k^2 in Eq. (1). $\log \sigma(M(v_i^1)^T \cdot v_h^2)$ is a function of $M(v_i^1)$, and $M(v_i^1)$ is a function of v_i^1. We employ the chain rule for composite functions to compute the gradient of $\log \sigma(M(v_i^1)^T \cdot v_h^2)$.

$$\frac{\partial \log \sigma(M(v_i^1)^T \cdot v_h^2)}{\partial v_i^1} = \frac{\partial \log \sigma(M(v_i^1)^T \cdot v_h^2)}{\partial M(v_i^1)^T} \frac{\partial M(v_i^1)^T}{\partial v_i^1} = \phi(M(v_i^1)^T, v_h^2) J_i, \tag{3.4}$$

where

$$\phi(M(v_i^1)^T, v_h^2) = \frac{\partial \log \sigma(M(v_i^1)^T \cdot v_h^2)}{\partial M(v_i^1)^T}$$

is a signal from the alignment loss and

$$J_i = \frac{\partial M(v_i^1)^T}{\partial v_i^1} \in \mathbb{R}^{N^2 \times N^1}$$

is the input-output Jacobian matrix of the neural network.

Both the signal $\phi(M(v_i^1)^T, v_h^2)$ and the Jacobian matrix J_i influence the gradient

$$\frac{\partial \log \sigma(M(v_i^1)^T \cdot v_h^2)}{\partial v_i^1}$$

together. The attack would fail if either the signal has saturating gradient or the Jacobian is poorly conditioned. Either of them will lead to vanishing gradients in

$$\frac{\partial \log \sigma(M(v_i^1)^T \cdot v_h^2)}{\partial v_i^1}.$$

3.4 Signal Scaling via Dynamical Isometry

A recent study describes that a neural network is dynamical isometry if all singular values λ_{ir} of the Jacobian J_i are close to 1, i.e.,

$$1 - \lambda_{ir} \leq \xi \quad \text{for } \forall r, r \in \{1, \cdots, \min\{N^1, N^2\}\}$$

and a small positive number $\xi \approx 0$ [200]. In this case, the loss signal $\phi(M(v_i^1)^T, v_h^2)$ backpropagates isometrically through the neural network, and thus maintains the norm and all angles between vectors. They utilize the dynamical isometry to speed up the training of neural networks by improving the signal propagation. Motivated by this, we explore to employ the dynamical isometry to improve the iterative gradient-based adversarial attack.

In order to keep the decision boundary of network alignment invariant, we propose to discover a well-chosen scalar $\alpha \geq 0$ to improve signal propagation as well as maintain the relative order of the logits at the output layer of the neural network.

$$\tilde{M}(v_i^1) = \alpha M(v_i^1) \tag{4}$$

where α is the attack signal scale.

By enhancing with α, the gradients can be reformulated below.

$$\frac{\partial \log \sigma(\tilde{M}(v_i^1)^T \cdot v_h^2)}{\partial v_i^1} = \frac{\partial \log \sigma(M(v_i^1)^T \cdot v_h^2)}{\partial \tilde{M}(v_i^1)^T} \frac{\partial \tilde{M}(v_i^1)^T}{\partial M(v_i^1)^T} \frac{\partial M(v_i^1)^T}{\partial v_i^1} = \phi(\tilde{M}(v_i^1)^T, v_h^2) \alpha J_i \tag{5}$$

where α linearly influences the Jacobian matrix J_i while nonlinearly affecting the loss signal $\phi(M(v_i^1)^T, v_h^2)$.

As

$$\phi(\tilde{M}(v_i^1)^T, v_h^2) = \sigma(\tilde{M}(v_i^1)^T \cdot v_h^2) \left(1 - \sigma(\tilde{M}(v_i^1)^T \cdot v_h^2)\right) v_h^2$$

it is clear that the loss signal is equal to zero when $\alpha = \infty$ in Eq. (5). This leads to the vanishing gradient issue in adversarial attacks. On the other hand, when $\alpha = 0$, the norm of the loss signal is maximal. However, the singular values of αJ_i are all zeros. Thus,

$$\frac{\partial \log \sigma(\tilde{M}(v_i^1)^T \cdot v_h^2)}{\partial v_i^1}$$

is equal to zero, which results in the vanishing gradient issue too.

Based on the above analysis, in order to alleviate the vanishing gradients, a desired α should (1) guarantee αJ_i is well conditioned, i.e., all singular values of αJ_i are close to 1, such that the loss signal $\phi(\tilde{M}(v_i^1)^T, v_h^2)$ can be well backpropagated from the output layer to the input layer and (2) ensure the loss signal meaningful, i.e.,

$$\|\phi(\tilde{M}(v_i^1)^T, v_h^2)\|_2 > \eta$$

for a certain positive signal threshold η.

First, we choose α as the inverse of the mean of singular values of J_i, so as to scale the mean of singular values of αJ_i to closer to 1.

$$\alpha = \frac{|D|N}{\sum_{i=1}^{|D|} \sum_{r=1}^{N} \lambda_{ir}} \tag{6}$$

where $|D|$ is the size of the set D of aligned node pairs and $N = \min\{N^1, N^2\}$. λ_{ir} denotes the rth singular value of J_i.

Second, in order to guarantee

$$\|\phi(\tilde{M}(v_i^1)^T, v_h^2)\|_2 > \eta,$$

we need to select α to ensure

$$\left(1 - \sigma(\tilde{M}(v_i^1)^T \cdot v_h^2)\right) > \frac{\eta}{\|v_h^2\|_2}.$$

The following theorem demonstrates the upper bound of a qualified α.

Theorem 4.1. Let nodes v_j^2 and v_k^2 be the most similar and least similar to $M(v_i^1)^T$ $(1 \leq j, k \leq N^2)$, i.e., $v_j^2 = \arg\max_{v_j^2}(M(v_i^1)^T \cdot v_j^2)$ and $v_k^2 = \arg\min_{v_k^2}(M(v_i^1)^T \cdot v_k^2)$, and $c = (M(v_i^1)^T \cdot v_j^2)$. Also, suppose that d is the minimal norm of node representation vectors in G^2, i.e., $d = \min_{v_g^2} \|v_g^2\|_2$ for $\forall v_g^2 \in V^2$. For a given $0 < \eta < d/2$, if

$$\alpha < \frac{1}{c} \log \frac{d - \eta}{\eta},$$

then

$$1 - \sigma(\alpha M(v_i^1)^T \cdot v_h^2) > \frac{\eta}{\|v_h^2\|_2}$$

for $\forall v_h^2 \in V^2$.

The above two types of α are integrated together to make the Jacobian well-conditioned as well as the loss signal meaningful.

By assembling PGD-based adversarial attacks and attack signal scaling together, Algorithm 1 presents the pseudo code of our adversarial attack model. Line 2 initializes adversarial nodes $\hat{\mathbf{v}}_i^1$ and $\hat{\mathbf{v}}_j^2$ with real aligned nodes $\mathbf{v}_i^1$ in G^1 and $\mathbf{v}_j^2$ in G^2. Line 3 computes the first signal scale α_1 to make the Jacobian $\alpha\mathbf{J}_i$ well-conditioned. Lines 5-8 calculate the second signal scale α_2 to make the loss signal meaningful. Lines 9-10 scale and improve the attack signal propagation of neural networks by integrating two signal scales. Line 11 utilizes the

PGD method to add and remove edges. The loop repeats the above iterative procedure until achieving the maximum iterations of the PGD.

Algorithm 3: PGD Attacks with Attack Signal Scaling

Input: Source network $G^1 = (V^1, E^1)$, target network $G^2 = (V^2, E^2)$, set of known aligned node pairs $D = \{(\mathbf{v}_i^1, \mathbf{v}_j^2) | \mathbf{v}_i^1 \leftrightarrow \mathbf{v}_j^2\}$, trained network alignment model M, noise budget ϵ, and signal threshold η.
Output: Adversarial node pairs $\{(\hat{\mathbf{v}}_i^1, \hat{\mathbf{v}}_j^2) | (\mathbf{v}_i^1, \mathbf{v}_j^2) \in D\}$.
for each aligned node pair $(\mathbf{v}_i^1, \mathbf{v}_j^2)$ in D **do**
  Initialize $\hat{\mathbf{v}}_i^1 = \mathbf{v}_i^1$ and $\hat{\mathbf{v}}_j^2 = \mathbf{v}_j^2$;
  Compute signal scale $\alpha_1 = \frac{|D|N}{\sum_{i=1}^{|D|}\sum_{r=1}^{N}\lambda_{ir}}$ in Eq.(6);
  for $s = 1, \cdots, S$ **do**
    Initialize signal scale $\alpha_2 = 1.0$;
    for each $\hat{\mathbf{v}}_h^2$ of positive sample $\hat{\mathbf{v}}_j^2$ and K negative ones $\mathbf{v}_k^2$ **do**
      if $1 - \sigma\left(\alpha_1 M(\hat{\mathbf{v}}_i^1)^T \cdot \hat{\mathbf{v}}_h^2\right) \leq \eta / \|\hat{\mathbf{v}}_h^2\|_2$ **then**
        Update $\alpha_2 = \frac{1}{c}\log\frac{d-\eta}{\eta}$ based on Theorem 5.1;
      end if
      Scale attack signal $\tilde{M}(\hat{\mathbf{v}}_i^1) = \alpha_1\alpha_2 M(\hat{\mathbf{v}}_i^1)$;
      Calculate $\frac{\partial\left(\tilde{M}(\hat{\mathbf{v}}_i^1)^T \cdot \hat{\mathbf{v}}_h^2\right)}{\partial \hat{\mathbf{v}}_i^1}$ and $\frac{\partial\left(\tilde{M}(\hat{\mathbf{v}}_i^1)^T \cdot \hat{\mathbf{v}}_h^2\right)}{\partial \hat{\mathbf{v}}_j^2}$;
      Use the PGD to update $\hat{\mathbf{v}}_i^1$ and $\hat{\mathbf{v}}_j^2$ in terms of Eq.(2);
    end for
  end for
end for
Return $\{(\hat{\mathbf{v}}_i^1, \hat{\mathbf{v}}_j^2) | (\mathbf{v}_i^1, \mathbf{v}_j^2) \in D\}$.

Theorem 5.1. Let nodes v_j^2 and v_k^2 be the most similar and least similar to $M(v_i^1)^T$ $(1 \leq j, k \leq N^2)$, i.e., $v_j^2 = \arg\max_{v_j^2}(M(v_i^1)^T \cdot v_j^2)$ and $v_k^2 = \arg\min_{v_k^2}(M(v_i^1)^T \cdot v_k^2)$, and $c = (M(v_i^1)^T \cdot v_j^2)$. Also, suppose that d is the minimal norm of node representation vectors in G^2, i.e., $d = \min_{v_g^2} \|v_g^2\|_2$ for $\forall v_g^2 \in V^2$. For a given $0 < \eta < d/2$, if

$$\alpha < \frac{1}{c}\log\frac{d-\eta}{\eta},$$

then

$$1 - \sigma(\alpha M(v_i^1)^T \cdot v_h^2) > \frac{\eta}{\|v_h^2\|_2}$$

for $\forall v_h^2 \in V^2$.

Proof. $1 - \sigma(\alpha M(v_i^1)^T \cdot v_h^2) > \eta / \|v_h^2\|_2$ is equivalent to $\sigma(\alpha M(v_i^1)^T \cdot v_h^2) < 1 - \eta / \|v_h^2\|_2$. We convert it to

$$\frac{1}{1 + \exp(-\alpha M(v_i^1)^T \cdot v_h^2)} < 1 - \frac{\eta}{\|v_h^2\|_2}.$$

As $(M(v_i^1)^T \cdot v_h^2) \leq c$, we have

$$\frac{1}{1 + \exp(-\alpha c)} \leq \frac{1}{1 + \exp(-\alpha M(v_i^1)^T \cdot v_h^2)}.$$

If we can prove

$$\frac{1}{1 + \exp(-\alpha c)} < 1 - \frac{\eta}{\|v_h^2\|_2},$$

then we can testify

$$\frac{1}{1 + \exp(-\alpha M(v_i^1)^T \cdot v_h^2)} < 1 - \frac{\eta}{\|v_h^2\|_2}.$$

Thus, we need to solve

$$\exp(\alpha c) < \frac{\|v_h^2\|_2 - \eta}{\eta}.$$

Since exp is a monotonic increasing function, by solving the above inequality, we have

$$\alpha < \frac{1}{c} \log \frac{\|v_h^2\|_2 - \eta}{\eta}.$$

As $\|v_h^2\|_2 \geq d$ and

$$\alpha < \frac{1}{c} \log \frac{d - \eta}{\eta},$$

the above statement is proved.

Notice that $0 < \eta < d/2$. This makes $\frac{d-\eta}{\eta} > 1$ and the upper bound of α be positive. Therefore, for any $\alpha < \frac{1}{c} \log \frac{d-\eta}{\eta}$, $1 - \sigma(\alpha M(v_i^1)^T \cdot v_h^2) > \eta / \|v_h^2\|_2$ is satisfied.

Theorem 6.2. If $P(\hat{v}_i^1 | v_i^1) = \tau(\hat{v}_i^1 - v_i^1)$ and $Q(\hat{v}_j^2 | v_j^2) = \tau(\hat{v}_j^2 - v_j^2)$ for $\forall (v_i^1, v_j^2) \in D$, where $\tau(\cdot)$ is the Dirac delta function, then $\mathcal{L}_D$ in Eq. (7) is equivalent to $\mathcal{L}$ in Eq. (1).

Proof.

$$\mathcal{L}_D = \mathbb{E}_{(v_i^1, v_j^2)\in D} \int_{\Delta_i^1} \int_{\Delta_j^2} \mathbb{E}_{\hat{v}_i^1 \sim P(\hat{v}_i^1|v_i^1)} \mathbb{E}_{\hat{v}_j^2 \sim Q(\hat{v}_j^2|v_j^2)} \mathcal{L}(\hat{v}_i^1, \hat{v}_j^2) q(\delta_i^1) p(\delta_j^2) d\delta_i^1 d\delta_j^2$$

$$= \mathbb{E}_{(v_i^1, v_j^2)\in D} \int_{\Delta_i^1} p(\delta_i^1) d\delta_i^1 \int_{\Delta_j^2} q(\delta_j^2) d\delta_j^2 \int_{v_i^1} P(\hat{v}_i^1|v_i^1) dv_i^1 \int_{v_j^2} Q(\hat{v}_j^2|v_j^2) \mathcal{L}(\hat{v}_i^1, \hat{v}_j^2) dv_j^2$$

$$= \mathbb{E}_{(v_i^1, v_j^2)\in D} \int_{\Delta_i^1} p(\delta_i^1) d\delta_i^1 \int_{\Delta_j^2} q(\delta_j^2) d\delta_j^2 \int_{v_i^1} \tau(\hat{v}_i^1 - v_i^1) dv_i^1 \int_{v_j^2} \tau(\hat{v}_j^2 - v_j^2) \mathcal{L}(v_i^1, v_j^2) dv_j^2 \tag{11}$$

According to the sifting property of the integral of Dirac delta function, we apply the sifting operation twice and have:

$$\mathcal{L}_D = \mathbb{E}_{(v_i^1, v_j^2)\in D} \int_{\Delta_i^1} p(\delta_i^1) d\delta_i^1 \int_{\Delta_j^2} q(\delta_j^2) d\delta_j^2 \int_{v_i^1} \tau(\hat{v}_i^1 - v_i^1) dv_i^1 \mathcal{L}(v_i^1, v_j^2)$$

$$= \mathbb{E}_{(v_i^1, v_j^2)\in D} \int_{\Delta_i^1} p(\delta_i^1) d\delta_i^1 \int_{\Delta_j^2} q(\delta_j^2) d\delta_j^2 \mathcal{L}(v_i^1, v_j^2)$$

$$= \mathbb{E}_{(v_i^1, v_j^2)\in D} \mathcal{L}(v_i^1, v_j^2) = \mathcal{L}. \tag{12}$$

Theorem 6.3. Assuming that $(\delta_i^{1*}, \delta_j^{2*}) = \arg\min_{\delta_i^1 \in \Delta_i^1, \delta_j^2 \in \Delta_j^2} \mathcal{L}(v_i^1 + \delta_i^1, v_j^2 + \delta_j^2)$, if $P(\hat{v}_i^1|v_i^1) = \tau(\hat{v}_i^1 - v_i^1 - \delta_i^{1*})$ and $Q(\hat{v}_j^2|v_j^2) = \tau(\hat{v}_j^2 - v_j^2 - \delta_j^{2*})$ for $\forall (v_i^1, v_j^2) \in D$, then $\mathcal{L}_D$ in Eq. (7) achieves its lower bound.

Proof.

$$\mathcal{L}_D = \mathbb{E}_{(v_i^1, v_j^2)\in D} \int_{\Delta_i^1} \int_{\Delta_j^2} \mathbb{E}_{\hat{v}_i^1 \sim P(\hat{v}_i^1|v_i^1)} \mathbb{E}_{\hat{v}_j^2 \sim Q(\hat{v}_j^2|v_j^2)} \mathcal{L}(\hat{v}_i^1, \hat{v}_j^2) q(\delta_i^1) p(\delta_j^2) d\delta_i^1 d\delta_j^2$$

$$= \mathbb{E}_{(v_i^1, v_j^2)\in D} \int_{\Delta_i^1} p(\delta_i^1) d\delta_i^1 \int_{\Delta_j^2} q(\delta_j^2) d\delta_j^2 \int_{v_i^1} P(\hat{v}_i^1 | v_i^1) dv_i^1 \int_{v_j^2} Q(\hat{v}_j^2 | v_j^2) \mathcal{L}(\hat{v}_i^1, \hat{v}_j^2) dv_j^2$$

$$\geq \mathbb{E}_{(v_i^1, v_j^2)\in D} \int_{\Delta_i^1} p(\delta_i^1) d\delta_i^1 \int_{\Delta_j^2} q(\delta_j^2) d\delta_j^2 \int_{v_i^1} P(\hat{v}_i^1 | v_i^1) dv_i^1 \int_{v_j^2} Q(\hat{v}_j^2 | v_j^2) [\min \mathcal{L}(\hat{v}_i^1, \hat{v}_j^2)] dv_j^2$$

$$= \mathbb{E}_{(v_i^1, v_j^2)\in D} \int_{\Delta_i^1} p(\delta_i^1) d\delta_i^1 \int_{\Delta_j^2} q(\delta_j^2) d\delta_j^2 \int_{v_j^2} Q(\hat{v}_j^2 | v_j^2) dv_j^2 \mathcal{L}(v_i^1 + \delta_i^{1*}, v_j^2 + \delta_j^{2*})$$

$$= \mathbb{E}_{(v_i^1, v_j^2)\in D} \int_{\Delta_i^1} p(\delta_i^1) d\delta_i^1 \int_{\Delta_j^2} q(\delta_j^2) d\delta_j^2 \mathcal{L}(v_i^1 + \delta_i^{1*}, v_j^2 + \delta_j^{2*})$$

$$= \mathbb{E}_{(v_i^1, v_j^2)\in D} \mathcal{L}(v_i^1 + \delta_i^{1*}, v_j^2 + \delta_j^{2*})$$

The equality is satisfied when $P(\hat{v}_i^1 | v_i^1) = \tau(\hat{v}_i^1 - v_i^1 - \delta_i^{1*})$ and $Q(\hat{v}_j^2 | v_j^2) = \tau(\hat{v}_j^2 - v_j^2 - \delta_j^{2*})$.

The above two types of α are integrated together to make the Jacobian well-conditioned as well as the loss signal meaningful.

By assembling PGD-based adversarial attacks and attack signal scaling together, Algorithm 1 presents the pseudo code of our adversarial attack model. Line 2 initializes adversarial nodes $\hat{v}_i^1$ and $\hat{v}_j^2$ with real aligned nodes v_i^1 in G^1 and v_j^2 in G^2. Line 3 computes the first signal scale α_1 to make the Jacobian αJ_i well-conditioned.

3.5 Adversarial Perturbation Elimination

In this section, we develop an adversarial perturbation elimination (APE) model by integrating Dirac delta approximation (DDA) techniques and the LSTM models to offer preemptive protection to trained network alignment models.

3.5.1 Perturbation Elimination

Based on a trained network alignment model M and any network alignment loss, e.g., the loss $\mathcal{L}$ defined in Eq. (1), the defender aims to learn an APE model $P(\hat{v}_i^1|v_i^1)$ (or $Q(\hat{v}_j^2|v_j^2)$) to neutralize adversarial nodes $\hat{v}_i^1$ (or $\hat{v}_j^2$) in vulnerable space Δ_i^1 (or Δ_j^2) to adversarial-free nodes $\tilde{v}_i^1$ (or $\tilde{v}_j^2$) in safe area, such that $\tilde{v}_i^1$ (or $\tilde{v}_j^2$) are close to original clean nodes v_i^1 (or v_j^2). The defense loss function $\mathcal{L}_D$ is defined to minimize the following marginalized expectation.

$$\mathcal{L}_D = \mathbb{E}_{(v_i^1, v_j^2)\in D} \int_{\Delta_i^1} \int_{\Delta_j^2} \mathbb{E}_{\tilde{v}_i^1 \sim P(\tilde{v}_i^1|v_i^1)} \mathbb{E}_{\tilde{v}_j^2 \sim Q(\tilde{v}_j^2|v_j^2)} \mathcal{L}(\tilde{v}_i^1, \tilde{v}_j^2) q(\delta_j^2) p(\delta_i^1) d\delta_j^2 d\delta_i^1 \tag{7}$$

where Δ_i^1 and Δ_j^2 have the same definitions as the symbols in Eq. (6). $p(\delta_i^1)$ and $q(\delta_j^2)$ denote the distribution of edge perturbations in Δ_i^1 and Δ_j^2, respectively.

As discussed earlier, each dimension v_{ik}^1 in the representation vector v_i^1 denotes the existence of edge $(v_i^1, v_k^1) \in E^1$ in G^1. We treat each v_i^1 as a one-dimensional sequence $v_{i1}^1, \cdots, v_{iN^1}^1$ and use one LSTM as the probabilistic model to learn a joint probability $P(\tilde{v}_i^1|\hat{v}_i^1)$ among all dimensions, which is factorized into a product of conditional distributions.

$$P(\tilde{v}_i^1|\hat{v}_i^1) = \prod_{k=1}^{N^1} p(\tilde{v}_{ik}^1|[\tilde{v}_{i1}^1, \cdots, \tilde{v}_{i(k-1)}^1], \hat{v}_i^1). \tag{8}$$

Based on adversarial nodes $\hat{v}_i^1$, the LSTM takes the hidden state w.r.t. learnt adversarial-free dimensions $[\tilde{v}_{i1}^1, \cdots, \tilde{v}_{i(k-1)}^1]$ as input to estimate current adversarial-free dimensions $\tilde{v}_{ik}^1$ with the conditional probability $p(\tilde{v}_{ik}^1|[\tilde{v}_{i1}^1, \cdots, \tilde{v}_{i(k-1)}^1], \hat{v}_i^1)$. Similarly, we employ another LSTM to learn a joint probability $Q(\tilde{v}_j^2|\hat{v}_j^2)$ and neutralize adversarial nodes $\hat{v}_j^2$ in G^2.

The following theorems exhibit the defense loss $\mathcal{L}_D$ on adversarial-free nodes in Eq. (7) is equivalent to the original alignment loss $\mathcal{L}$ on clean nodes in Eq. (1), if satisfied with some conditions. In addition, they exhibit the existence of an optimal distribution for the APE model to reach a lower bound.

Theorem 5.1. If $P(\tilde{v}_i^1|v_i^1) = \tau(\tilde{v}_i^1 - v_i^1)$ and $Q(\tilde{v}_j^2|v_j^2) = \tau(\tilde{v}_j^2 - v_j^2)$ for $\forall(v_i^1, v_j^2) \in D$, where $\tau(\cdot)$ is the Dirac delta function, then $\mathcal{L}_D$ in Eq. (7) is equivalent to $\mathcal{L}$ in Eq. (1).

Theorem 5.2. Assuming that $(\delta_i^{1*}, \delta_j^{2*}) = \arg\min_{\delta_i^1 \in \Delta_i^1, \delta_j^2 \in \Delta_j^2} \mathcal{L}(v_i^1 + \delta_i^1, v_j^2 + \delta_j^2)$, if $P(\tilde{v}_i^1|v_i^1) = \tau(\tilde{v}_i^1 - v_i^1 - \delta_i^{1*})$ and $Q(\tilde{v}_j^2|v_j^2) = \tau(\tilde{v}_j^2 - v_j^2 - \delta_j^{2*})$ for $\forall(v_i^1, v_j^2) \in D$, then $\mathcal{L}_D$ in Eq. (7) achieves its lower bound.

Lemma 5.3. If $\mathcal{L}_D$ in Eq. (7) achieves its lower bound, adversarial perturbation exists only if $\delta_i^1 \notin \Delta_i^1$ and $\delta_j^2 \notin \Delta_j^2$.

3.5.2 Dirac Delta Approximation

The above theoretical analysis offers a great opportunity to integrate DDA techniques into the LSTM models to make $\tilde{v}_i^1$ (or $\tilde{v}_j^2$) be close to v_i^1 (or v_j^2) as much as possible, and cause $\mathcal{L}_D$ on adversarial-free nodes to be identical to $\mathcal{L}$ on clean nodes.

In mathematics, the Dirac delta function is a distribution to model the density of an idealized point mass or point charge as a function equal to zero everywhere except for zero and whose integral over the entire real line is equal to one [73]. In practice, a Dirac delta function can be approximated using a commonly used method developed by Zahedi and Tornberg [304].

$$\tau(x) = \frac{y \exp(-yx)}{(1 + \exp(-yx))^2} \tag{9}$$

where x is a scalar variable and y is a scalar constant. The larger y is, the more accurate the approximation will be. It can be extended to a vector $\mathbf{x} = [x_1, \cdots, x_n]$ with the form of $\tau(x) = \prod_{i=1}^n \tau(x_i)$.

In this paper, we utilize the DDA method to approximate the generative probability of adversarial-free nodes. Here, we set $y = 4$ to make the maximum of $\tau(x)$ equal to 1, i.e., $\tau(x) = 1$ when $x = 0$.

3.6 Robust Graph Alignment

3.6.1 Experimental Setup

Dataset	AS		SNS		DBLP
Graph	v1	v2	Last.fm	LiveJournal	2013
#Nodes	10,900	11,113	5,682	17,828	28,478
#Edges	31,180	31,434	23,393	244,496	128,073
#Matched Nodes	6,462		2,138		4,000

Table 3.1: Statistics of the Datasets

In this section, we perform a set of extensive experiments to evaluate the robustness of our RNA model for network alignment on three groups of datasets: autonomous systems (**AS**) [3], social networks (**SNS**) [13], and DBLP coauthor networks [2], as shown in Table 3.1.

Autonomous system is a collection of connected Internet Protocol networks and routers under the control of network operators. **Last.fm** is a music-oriented online social network that provides a radio streaming service. **LiveJournal** is an online social platform where users can keep a blog, journal, or diary. **DBLP** is a computer science bibliography website that offers an online search and retrieval service for the academic community. We collect two DBLP versions with highly prolific authors from all research areas, each with the data in 2013 and 2014 respectively.

Attack baselines. We compare our **SSPGD** model with six state-of-the-art graph attack models. **Random Attack** randomly adds and removes edges to generate perturbed graphs. **Meta-Self** [369] is a poisoning attack model for node classification by using meta-gradients to solve the bilevel optimization problem. **GF-Attack** [47] attacks general learning methods by devising new loss and approximating the spectrum. **CD-ATTACK** [146] hides nodes in the community by attacking the graph autoencoder model. **LowBlow** [80] is a general low-rank adversarial attack model which is able to affect the performance of various graph learning tasks. **GMA** [319] is the first to conduct adversarial attacks on graph matching (i.e., network alignment) by estimating and maximizing the densities of nodes to

be attacked, for pushing them to dense regions in two graphs to generate imperceptible and effective attacks.

Network Alignment AlgorithmsSNNA [143] is an adversarial learning model to solve the weakly-supervised identity alignment problem by incorporating available annotations as the learning guidance. **CrossMNA** [65] is a cross-network embedding-based supervised network alignment method by learning inter/intra-embedding vectors for each node and by computing pairwise node similarity scores across networks. **Deep Graph Matching Consensus (DGMC)** [87] is a supervised graph matching method which aims to reach a data-driven neighborhood consensus between matched node pairs.

Defense Baselines To our best knowledge, this work is the first to deal with the robustness analysis of network alignment against adversarial attacks. We compare our RNA model with three state-of-the-art graph perturbation elimination models: **GCN-Jaccard** [277] eliminates edges that connect nodes with Jaccard similarity of features smaller than a threshold τ. Here we use structural features of nodes to calculate the Jaccard similarity. **GCN-SVD** [80] learns a low-rank approximation of the graph to resist high-rank perturbations. Both **GCN-Jaccard** and **GCN-SVD** are general perturbation elimination models irrelevant to specific graph learning tasks and architectures. **Pro-GNN** [119] jointly learns a clean graph and a robust graph neural network model from the perturbed graph guided by the low rank and sparsity properties. Notice that it originally targets at defending node classification. In order to make a fair comparison, we change the classification loss $\mathcal{L}_{GNN}$ in the original paper as the alignment loss. We will verify the robustness of these perturbation elimination models by feeding neutralized graphs into the above three network alignment algorithms.

Variants of Our Model We evaluate five variants of our method to show the strengths of different components. **Attack variants**: **PGD** only utilizes the basic PGD model [167] to produce adversarial attacks; **SSPGD** uses our proposed attack signal scaling (ASS) plus the PGD model to generate effective attacks. **Defense variants**: **RNA-A** only employs

the basic adversarial perturbation elimination (APE) model (without the ASS and Dirac delta approximation (DDA) models) to neutralize adversarial attacks; **RNA-AD** uses the APE model plus the DDA model (without adversarial attacks as the supervision); **RNA** well eliminates adversarial perturbations with the full support of the APE, DDA, and ASS components.

Evaluation Metrics We use two popular measures in network alignment to evaluate the attack and defense quality: *Accuracy* [60, 311, 313] and *Precision@K* [65, 325]. A larger Mismatching Rate (i.e., $1 -$ Accuracy on test data) or a smaller *Precision@K* indicates a better attack, while a higher *Accuracy* or *Precision@K* presents a better defense. K is fixed to 30 in all tests.

Experimental Settings Our experiments were conducted on a compute server running on Red Hat Enterprise Linux 7.2 with 2 CPUs of Intel Xeon E5-2650 v4 (at 2.66 GHz) and 8 GPUs of NVIDIA GeForce GTX 2080 Ti (with 11 GB of GDDR6 on a 352-bit memory bus and memory bandwidth in the neighborhood of 620 GB/s), 256 GB of RAM, and 1 TB of HDD. Overall, our experiments took about 8 days in a shared resource setting. We expect that a consumer-grade single-GPU machine (e.g., with a 1080 Ti GPU) could complete our full set of experiments in around 12-15 days, if its full resources were dedicated.

In our current implementation, the LSTM models in Eq. (10) are implemented as three-layer perceptrons (input-hidden-output). The number of neurons in the hidden layer is set to 500. The model uses a mini-batch of size 500. The learning rate is equal to 0.001.

The noise budget ϵ in Algorithm 1 specifies the budget of allowed perturbed edges for each attacked node. Thus, ϵ should be a positive integer. In addition, most of the real-world graph datasets (e.g., all datasets used in this paper) are extremely sparse, i.e., the average node degree of most datasets in this paper is 2 to 5. Thus, even if ϵ is very small, say 1 or 2, a large number of edges will be modified, which results in noticeable perturbations. For example, AS v1 contains 10,900 nodes and 31,180 edges. Therefore, we combine ϵ and the number limit of perturbed edges in entire graphs for the actual perturbation budget. For

instance, when $\epsilon = 1$, we randomly select one target node to add or remove one edge at a time and repeat the same process to attack other nodes until the overall edge perturbations in entire graphs are beyond the number limit of perturbed edges, say 5%. After that, we stop attacking the rest of the nodes.

Unless otherwise explicitly stated, we used the following default parameter settings in the experiments. The noise budget ϵ in adversarial attacks is set to 2. The number limit of perturbed edges in entire graphs is set to 5%. The attack signal threshold η in Algorithm 1 is set to 0.4. The scalar constant y in the Dirac delta approximation in Eq. (9) is fixed to 4. The number K of sampled negative nodes in Eq. (1) is set to 20.

For the three network alignment algorithms of **SNNA**, **CrossMNA**, and **DGMC**, we used the open-source implementation and default parameter settings by the original authors for our experiments. All models were trained for 500 iterations, with a batch size of 500 and a learning rate of 0.001. The training data ratio of the above three network alignment methods is fixed to 10%.

For the seven state-of-the-art graph attack/defense models of **Meta-Self**, **GF-Attack**, **CD-ATTACK**, **LowBlow**, **GMA**, **GCN-SVD**, and **Pro-GNN**, we also utilized the same model architecture as the official implementation provided by the original authors and used the same perturbation budgets to attack the deep graph learning models in all experiments.

3.6.2 Results

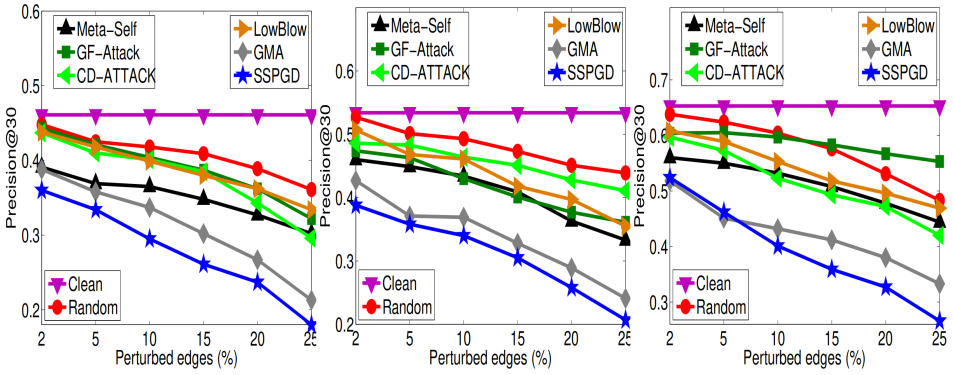
Attack Performance

Table 3.2 exhibits the mismatching rates of three deep network alignment algorithms on clean data and perturbed data by seven attack models over three groups of datasets. We randomly sample 10% of known matched node pairs as training data and the rest as test data. We repeat the selection process of matched node pairs five times and report the average scores. For all attack models, the number of perturbed edges is fixed to 5% in these experiments. It is observed that among eight attack methods, no matter how strong

Attack Model	AS			SNS			DBLP		
	SNNA	CrossMNA	DGMC	SNNA	CrossMNA	DGMC	SNNA	CrossMNA	DGMC
Clean	53.9	46.6	34.7	45.2	50.4	41.6	56.1	51.9	63.2
Random	57.5	49.9	37.6	48.8	52.0	46.8	59.8	54.0	68.8
Meta-Self	63.1	55.1	45.0	55.1	64.8	51.3	65.7	63.7	73.3
GF-Attack	57.9	53.7	39.5	52.9	59.6	47.9	64.9	61.1	69.1
CD-ATTACK	59.0	51.7	42.7	54.0	59.8	50.2	64.0	61.8	72.0
LowBlow	58.2	53.2	41.1	50.7	55.6	52.0	62.9	60.1	67.8
GMA	64.2	62.9	**54.9**	61.2	69.6	55.7	74.2	74.3	80.7
SSPGD	**66.6**	**64.2**	53.8	**65.3**	**71.0**	**58.7**	**77.4**	**78.2**	**82.4**

Table 3.2: Attack performance: Mismatching rate (%) with 5% perturbed edges.

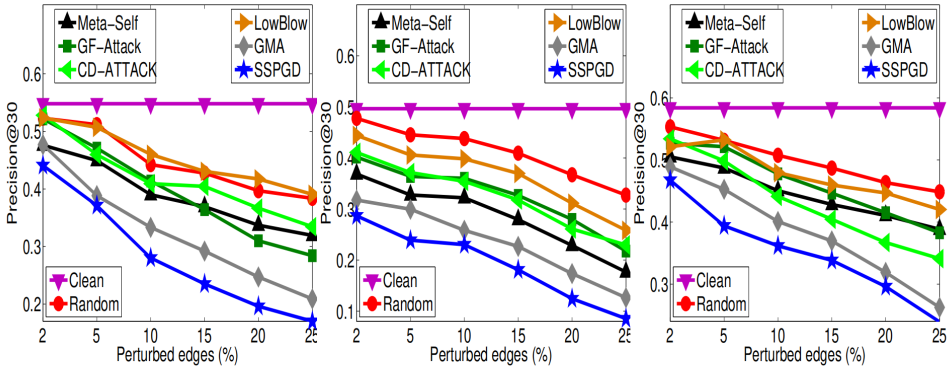
the attacks are, our proposed **SSPGD** attack method achieves the highest mismatching rates on perturbed graphs in most experiments, showing the effectiveness of SSPGD to the adversarial attacks. Compared to the network alignment results under other attack models, **SSPGD**, on average, achieves 17.9%, 21.2%, and 28.8% improvement of mismatching rates on **AS**, **SNS**, and **DBLP**, respectively. The promising performance of **SSPGD** with all three network alignment models implies that SSPGD has great potential as a general attack solution to other network alignment methods, which is desirable in practice.



(a) SNNA (b) CrossMNA (c) DGMC

Figure 3.2: Precision on AS with varying perturbed edges

Figures 3.2 and 3.3 show the network alignment quality under eight attack models by varying the ratios of perturbed edges from 2% to 25%. It is obvious that the attacking performance improves for each attacker with an increase in the number of perturbed edges. This phenomenon indicates that current deep network alignment algorithms are very sensitive

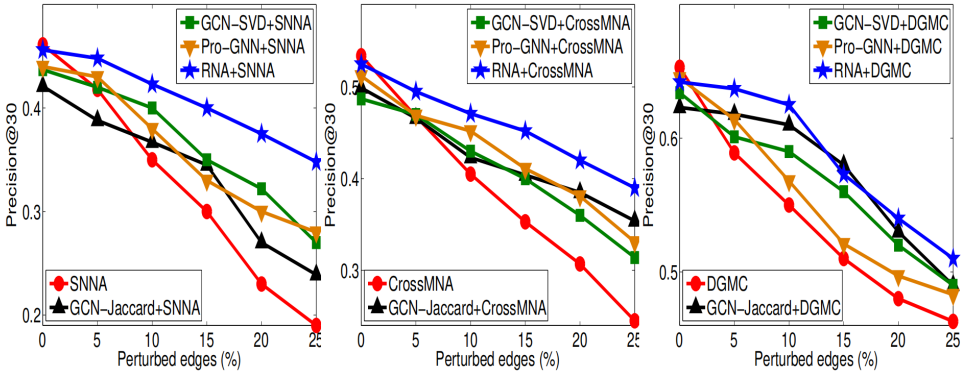


(a) SNNA (b) CrossMNA (c) DGMC

Figure 3.3: Precision on SNS with varying perturbed edges

to adversarial attacks. **SSPGD** achieves the lowest *Precision* values (< 0.524), which are much better than other seven methods in most tests. Especially, when the perturbation ratio is larger than 10%, the precision values drop quickly.

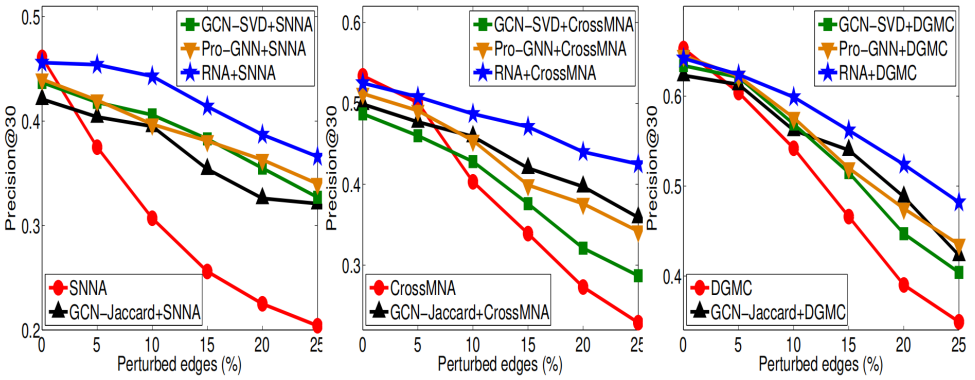
Defense Performance



(a) SNNA (b) CrossMNA (c) DGMC

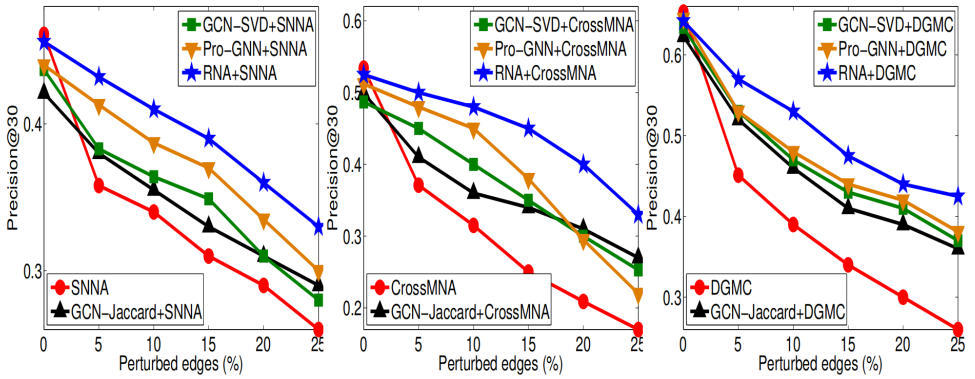
Figure 3.4: Defense on AS under Random Attack

Figures 3.4 to 3.9 present the network alignment results with the protection of four graph perturbation elimination models on the datasets **AS** and **SNS**. In order to verify the robustness to the adversarial attacks, we evaluate the performance of different alignment methods under three adversarial attack models. For **Random Attack**, we evaluate on both



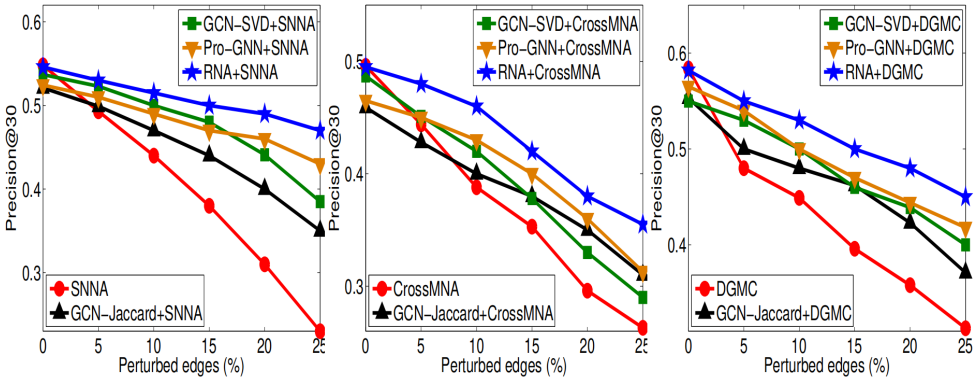
(a) SNNA (b) CrossMNA (c) DGMC

Figure 3.5: Defense on AS under LowBlow Attack



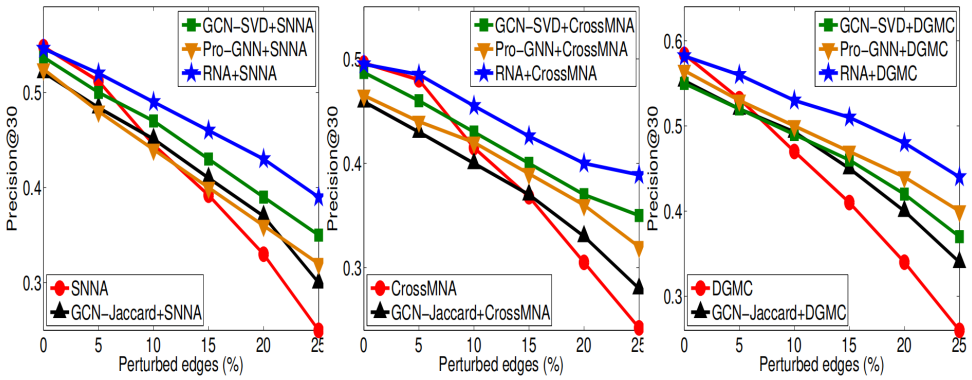
(a) SNNA (b) CrossMNA (c) DGMC

Figure 3.6: Defense on AS under GMA Attack



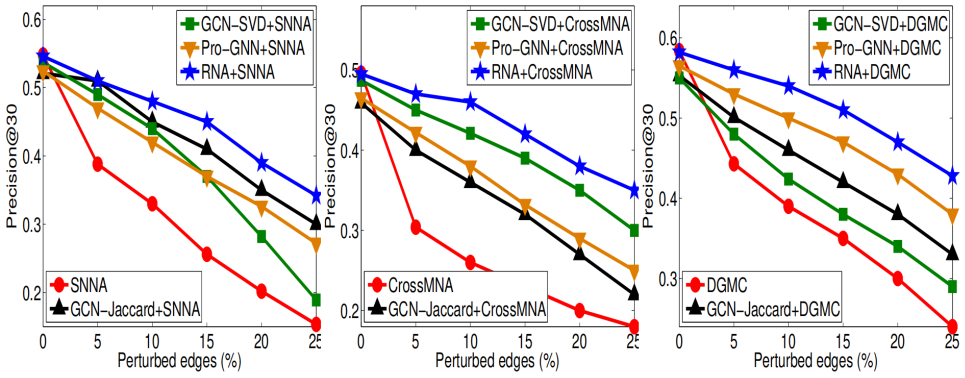
(a) SNNA (b) CrossMNA (c) DGMC

Figure 3.7: Defense on SNS under Random Attack



(a) SNNA (b) CrossMNA (c) DGMC

Figure 3.8: Defense on SNS under LowBlow Attack



(a) SNNA (b) CrossMNA (c) DGMC

Figure 3.9: Defense on SNS under GMA Attack

original and noisy networks to achieve the best quality in most experiments, demonstrating the effectiveness and robustness of **RNA**.

Compared to the network alignment results by all other algorithms, **RNA** achieves 15.2%, 20.3%, and 8.9% Precision boost on average by using **SNNA**, **CrossMNA**, and **DGMC** as network alignment methods, respectively. These results illustrate that both attack signal scaling and adversarial perturbation elimination are important in solving the robust network alignment problem. The former ensures informative attack signal propagation in adversarial attacks and thus helps identify the weak features in the graphs for better training adversarial perturbation elimination. On the other hand, with the help of effective attacks, the latter can neutralize adversarial nodes in vulnerable spaces to adversarial-free nodes in safe areas, alleviating the negative effect of adversarial attacks.

Ablation Study

Alignment Method	**AS**			**SNS**			**DBLP**		
	SNNA	**CrossMNA**	**DGMC**	**SNNA**	**CrossMNA**	**DGMC**	**SNNA**	**CrossMNA**	**DGMC**
PGD	60.5	57.1	48.4	53.5	60.8	51.7	66.5	61.6	71.2
SSPGD	**66.6**	**64.2**	**53.8**	**65.3**	**71.0**	**58.7**	**77.4**	**78.2**	**82.4**

Table 3.3: Attack: Mismatching rate (%) of SSPGD variants with 5% perturbed edges.

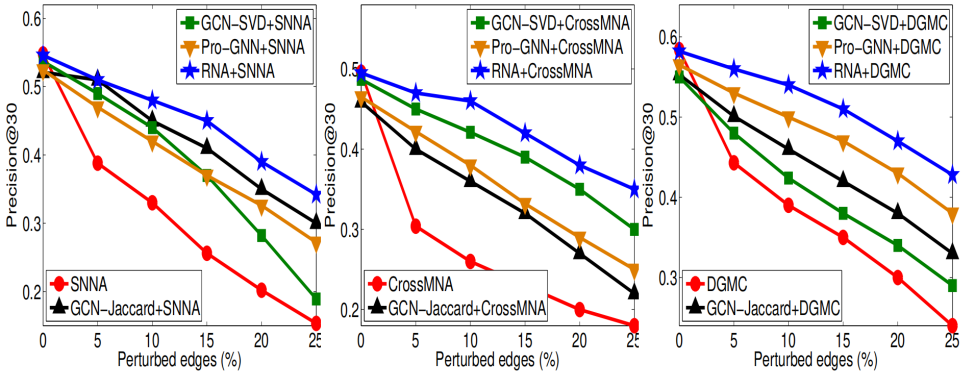
Alignment Method	**AS**			**SNS**			**DBLP**		
	SNNA	**CrossMNA**	**DGMC**	**SNNA**	**CrossMNA**	**DGMC**	**SNNA**	**CrossMNA**	**DGMC**
RNA-A	35.5	40.8	54.1	47.4	43.6	51.2	25.3	23.8	25.4
RNA-AD	41.7	45.6	58.2	49.5	46.9	52.8	29.4	26.9	27.3
RNA	**45.4**	**50.8**	**62.4**	**52.3**	**48.5**	**56.2**	**31.6**	**30.9**	**28.7**

Table 3.4: Defense: Precision (%) of RNA variants with 5% perturbed edges.

Table 3.3 presents the mismatching rates of network alignment on three groups of datasets with two variants of our attack model. We have observed that our **SSPGD** achieves the highest mismatching rates ($> 53.8\%$) on **AS**, ($> 58.7\%$) over **SNS**, and ($> 77.4\%$) on **DBLP**, which are obviously better than **PGD**. **PGD** does not utilize our attack signal scaling (**ASS**) method to solve the gradient vanishing issues and shows a fake sense of attack effectiveness.

Table 3.4 reports the precision scores of network alignment with three variants of our adversarial perturbation elimination (**APE**) model. **RNA** achieves the best performance in all experiments, while **RNA-AD** outperforms **RNA-A** in most experiments. A reasonable explanation is that the **ASS** model can help generate effective adversarial attacks, which assists our **APE** model to know what real attacks look like and learn how to combat them. The Dirac delta approximation (**DDA**) method ensures the **APE** model is able to make adversarial-free networks as close to original clean ones as much as possible.

Parameter Analysis



(a) SNNA (b) CrossMNA (c) DGMC

Figure 3.10: Precision of DGMC with varying parameters

Figure 3.10(a) measures the performance effect of different noise budgets ϵ for robust network alignment by varying ϵ from 1 to 6. We have observed that when increasing ϵ, the performance scores under three attack models initially keep stable or slightly decrease but decrease substantially after a certain threshold. This shows it is possible to train a robust perturbation elimination model by utilizing the guidance of adversarial attacks under an appropriate ϵ. However, too many adversarial perturbations can completely destroy the learning process of perturbation elimination. The noise with large ϵ is easily noticeable by users, and thus we suggest generating a robust perturbation elimination solution for network alignment with $\epsilon = 2$ or 3. Also, it seems that the denser the network is, the

larger the optimal ϵ should be. A reasonable explanation is denser networks need more edge perturbations for each node to change the relative order of the similarity scores between nodes and thus modify alignment results.

Figure 3.10(b) shows the influence of the signal threshold η in adversarial attacks over two groups of datasets. It is observed that the performance curves initially rise when η increases. Intuitively, this can help alleviate the vanishing gradient issue in gradient-based adversarial attacks. The effective adversarial attacks can help the training of the adversarial perturbation elimination model focus on the elimination of the perturbations on the weak edges, and thus improve the alignment robustness. Later on, the performance curves keep decreasing when η continuously increases. A rational guess is that a large η will make many negative nodes $v_k^2 \neq v_j^2$ have large gradients and thus influence the PGD computation in Eq. (2). However, the success of attacks mainly depends on the single v_k^2 that is most similar to v_j^2, instead of all negative nodes.

Figure 3.10(c) exhibits the impact of the scalar constant y in Dirac delta approximation. The experimental results are consistent with the discussion in Eq. (9), i.e., the larger y is, the more accurate the approximation will be. This approximation is a commonly used practice for the delta function in some existing works. We set $y = 4$ to approximate the generative probability of adversarial-free nodes. The performance curves keep relatively stable when y continuously increases to 5.

3.7 Conclusion

We have presented a robust network alignment solution. First, we analyze how gradient vanishing causes failures of gradient-based adversarial attacks. Second, we design an attack signal scaling method to ensure informative signal propagation. Finally, we develop an adversarial perturbation elimination model to neutralize adversarial nodes in vulnerable space to adversarial-free nodes in safe area.

Chapter 4

Adversarial Advertisement in Text-to-Image Generative Models

4.1 Digital Advertisement

This section presents formal definitions and notations regarding text-to-image diffusion models (T2I DMs), the attack scenario, the adversary's O_{tar}, and the adversary's capabilities.

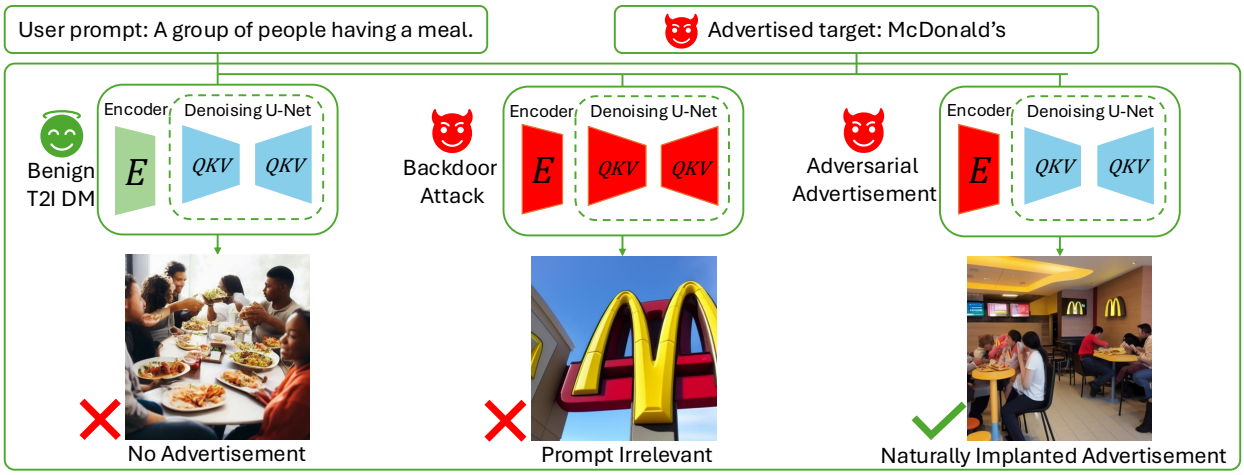


Figure 4.1: Illustration of the adversarial advertisement setting in T2I DMs. *(Left)* **Clean model**: the user prompt is processed faithfully, producing an image without McDonald's. *(Middle)* **Backdoor attack**: the attacked model produces the implanted pattern while ignoring the original prompt semantics upon detection of a trigger. *(Right)* **Adversarial advertisement** (ours): our attack implants advertisement naturally into the generated image while preserving the original semantics.

Text-to-image diffusion model A Text-to-image diffusion model (T2I DM) is a generative model that maps a textual caption (prompt) s to an image I. The model operates in two main steps. First, a text encoder $E(\cdot)$ processes the input prompt s to produce a latent representation $\mathbf{z}_s$. Next, a denoising network $\mathcal{G}(\cdot)$ takes the latent representation $\mathbf{z}_s$ and generates the final image I. Formally, the pipeline can be expressed as $I = \mathcal{G}(E(s))$, where $E : \mathcal{S} \to \mathcal{Z}$ maps the prompt space $\mathcal{S}$ to the latent space $\mathcal{Z}$, and $\mathcal{G} : \mathcal{Z} \to \mathcal{I}$ maps the latent space $\mathcal{Z}$ to the image space $\mathcal{I}$.

Advertised Target We denote the brand to be advertised as O_{tar}. Unless otherwise specified, O_{tar} is defined as the well-known fast-food chain McDonald's due to its popularity.

Attack scenario: We define an 'adversary' as an advertiser aiming to maximize the exposure of O_{tar} through image generation on the attacked T2I DM. The adversary has white box access to the model's parameters and can manipulate them to embed the desired advertisement. After completing the attack, the adversary releases the manipulated model on an open-source platform or community [112], where it is publicly available for users to download and use. This scenario is common in open-source machine learning communities, where personalized checkpoints are frequently shared and fine-tuned by users [276]. Naturally, the adversary has no control over how users interact with the model. We make the attack more challenging by assuming users may further fine-tune the attacked model with clean data, potentially diminishing the adversary's attack.

Adversary's goal: The adversary manipulates the T2I DM so that the generated images include O_{tar} as much as possible. Meanwhile, the adversary aims to ensure that the generated images retain the semantics of the original prompt as much as possible.

Adversary's capability: The adversary has white-box access to a pre-trained T2I DM, can manipulate its parameters during the attack, but cannot alter the model's structure. After completing the attack, the adversary can upload the modified model to an open-source platform or community [112] for users to access. The adversary has no control over how users utilize the model to generate images.

4.2 Adversarial Advertisement with Heavy-tail Phase-type Distribution

Although backdoor attack techniques can achieve adversarial advertisement implantation, a key challenge remains unsolved: how to incorporate the heavy-tailed characteristic of natural language corpora into perturbed prompts [115, 303, 110]. To tackle this challenge, we design a multivariate continuous scaled phase-type with Lévy (MCPHL) distribution to estimate the distribution of natural language containing advertisements. The high-density

regions of this distribution correspond to natural sentence embeddings with a higher likelihood of containing advertisements. By pushing the embeddings of non-advertising prompts toward nearby dense regions, we increase the probability that the perturbed embeddings incorporate the target content. Meanwhile, the heavy-tailed nature of our MCPHL ensures that the perturbed embeddings retain the characteristics of natural sentence embeddings. Consequently, the perturbed prompts become indistinguishable from natural prompts with the advertisements, resulting in generated images that not only contain the advertisement but also appear more natural. Theoretical analysis further validates that the estimation of our MCPHL converges to the empirical distribution from real data. Figure 4.2 shows

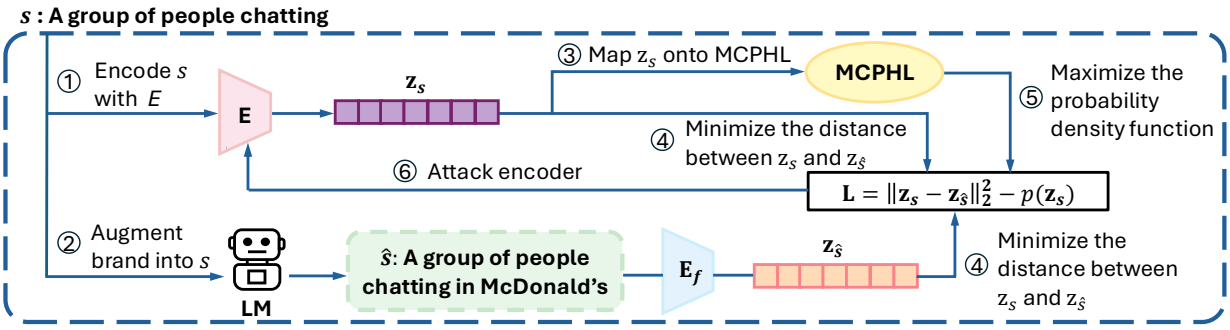


Figure 4.2: Adversarial advertisement implantation.

a high-level illustration of our advertisement implantation by attacking encoder E. Given a non-advertising prompt s (e.g., "A group of people chatting"), the text encoder E first converts it into a sentence embedding $\mathbf{z}_s$ (①). Simultaneously, a language model augments s with the target brand O_{tar}, generating a modified prompt $\hat{s}$ (e.g., "A group of people chatting at McDonald's"), which is then encoded by a fixed pre-trained encoder E_f into its corresponding embedding $\mathbf{z}_{\hat{s}}$ (②). Note that E_f's parameters are frozen during the attack. Next, the non-advertising embedding $\mathbf{z}_s$ is mapped onto the multivariate continuously scaled heavy-tail phase-type distribution space (MCPHL)(③). To guide $\mathbf{z}_s$ toward a nearby high-density region, we maximize its probability density $p(\mathbf{z}_s)$ within the estimated distribution. The loss function that minimizing the distance between $\mathbf{z}_s$ and $\mathbf{z}_{\hat{s}}$ (④) while maximizing the probability density $p(\mathbf{z}_s)$ (⑤) is used to update the victim encoder E (⑥). The attack makes

the output of the attacked encoder E indistinguishable from natural sentence embeddings that contain the target brand O_{tar}, ensuring brand exposure while preserving naturalness.

A phase-type distribution, formed by the convolution of exponential distributions, is dense among all positive-valued distributions, allowing it to approximate any positive-valued distribution [12, 189]. Despite its flexibility, it exhibits a light-tailed behavior, which makes it less effective for modeling heavy-tailed data like natural language distributions [115, 303, 110]. Continuously scaled phase-type distribution [7] provides a more expressive framework for capturing the heavy-tailed nature.

Definition A random variable X is said to follow a continuous scaled phase-type distribution with parameters (α, T, Θ) if its distribution function is given by

$$F_X(x) = 1 - \alpha \mathcal{L}_\Theta(-Tx)\mathbf{1}, \quad x > 0, \tag{4.1}$$

where X is a non-negative random variable, $\alpha \in \mathbb{R}^m$ represents the initial probabilities., $T \in \mathbb{R}^{m\times m}$ is a sub-intensity matrix [103], $\mathbf{1} \in \mathbb{R}^m$ is an all-one column vector, and $\mathcal{L}_\Theta(\lambda)$ is the Laplace transform of a positive real-valued random variable Θ, defined as $\mathbb{E}[e^{-\lambda\Theta}]$, $\lambda > 0$.

We choose Θ to follow a Lévy distribution with location parameter $\mu = 0$ and scale parameter $\eta > 0$.

Definition [81] Let $\mu \in \mathbb{R}$ be the location parameter and $\eta > 0$ the scale parameter. A random variable Θ follows a Lévy distribution, denoted as $\Theta \sim L(\mu, \frac{\eta^2}{2})$, where $\Theta \in (\mu, +\infty)$. The probability density function of the Lévy distribution is given by

$$f_\Theta(\theta; \mu, \eta) = \sqrt{\frac{\eta^2}{4\pi}} \frac{1}{(\theta-\mu)^{3/2}} \exp\left(-\frac{\eta^2}{4(\theta-\mu)}\right), \tag{4.2}$$

The Lévy distribution is a special case of the positive stable distribution with a stability parameter of $\frac{1}{2}$ and a skewness parameter of 1.

Definition Let X be a random variable following a continuous scaled phase-type with a Lévy (CPHL) distribution, where $\Theta \sim L(0, \frac{\eta^2}{2})$ is a Lévy-distributed random variable and $B = -\sqrt{-T}$ is a sub-intensity matrix. For $x > 0$, the survival function of X is defined as:

$$\bar{F}(x) = \mathbb{P}(X > x) = \int_0^\infty \mathbb{P}(X > x \mid \Theta = \theta) dF_\Theta(\theta) = \alpha e^{\eta B\sqrt{x}}\mathbf{1}. \tag{4.3}$$

As the set of prompt embeddings $\mathcal{E} = \{\mathbf{z} \mid \mathbf{z} = E(\hat{s}), \hat{s} \in \mathcal{S}\}$ lie in d-dimensional space, we use the multivariate continuous scaled phase-type with Lévy distribution to estimate their distribution. Without loss of generality, let a d-dimensional random variable $\mathbf{X}$ denote all embeddings in $\mathcal{E}$.

Definition For a d-dimensional random variable $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_d]$ and $0 \leq x_1 \leq \ldots \leq x_d$, assume $\mathbf{X}$ has the same boundary on all d dimension, i.e., $0 \leq x_1 = \ldots = x_d = x$, and let θ follow Lévy distribution with location parameter $\mu = 0$ and scale parameter $\eta > 0$. Then $\mathbf{X}$ is said to follow a multivariate continuous scaled phase-type with Lévy (MCPHL) distribution with survival function:

$$\begin{aligned}\bar{F}(x_1, x_2, \ldots, x_d) &= \int_0^\infty \alpha e^{\mathbf{T}Ax_d}\mathbf{D}_d e^{\mathbf{T}A(x_d - x_{d-1})}\mathbf{D}_{d-1}\ldots e^{\mathbf{T}A(x_2-x_1)} \quad \mathbf{D}_1 \frac{\eta}{2\sqrt{\pi\theta^3}} e^{-\frac{\eta^2}{4\theta}}\mathbf{1}d\theta \\ &= \alpha e^{\eta\mathbf{B}\sqrt{x}}\mathbf{D}\mathbf{1},\end{aligned} \tag{4.4}$$

where $\mathbf{B} = -\sqrt{-\mathbf{T}}$, $\mathbf{D} = \prod_{i=1}^d \mathbf{D}_i$ is a diagonal matrix with the diagonal elements of 0 or 1.

Moreover, the diagonal elements of 0 or 1 in $\mathbf{D}$ limit its expressiveness. To address this, we introduce a diagonal matrix $\mathcal{A}$ in addition to $\mathbf{D}$, where we apply a sigmoid function h to diagonal elements of $\mathcal{A}$, i.e., $\mathcal{A} = \text{diag}(h(d_1), \cdots, h(d_m))$. Based on newly introduced expressive factor $\mathcal{A}$, we have corresponding survival function $\bar{F}_\mathcal{A}(x_1, \ldots, x_d)$, distribution function $F_\mathcal{A}(x_1, \ldots, x_d) = 1 - \bar{F}_\mathcal{A}(x_1, \ldots, x_d) = 1 - \alpha e^{\eta\mathbf{B}\sqrt{x}}\mathbf{D}\mathcal{A}\mathbf{1}$ (i.e., $Q_\mathcal{A}(x)$), and probability density function

$$p_\mathcal{A}(x) = -\frac{\alpha\eta\mathbf{B}}{2\sqrt{x}} e^{\eta\mathbf{B}\sqrt{x}}\mathbf{D}\mathcal{A}\mathbf{1}, \tag{4.5}$$

and likelihood function $L(\alpha, \eta, \mathbf{B}, \mathbf{D}, \mathcal{A}|x)$. We use maximum likelihood estimation (MLE) [11, 44] to estimate α, η $\mathbf{B}$, $\mathbf{D}$, and $\mathcal{A}$:

$$L(\alpha, \eta, \mathbf{B}, \mathbf{D}, \mathcal{A}|x) = P_{\mathcal{A}}(x) \log Q_{\mathcal{A}}(x) + (1 - P_{\mathcal{A}}(x)) \log(1 - Q_{\mathcal{A}}(x)), \tag{4.6}$$

where $P_{\mathcal{A}}(x)$ is the observation and $Q_{\mathcal{A}}(x) = 1 - \bar{F}_{\mathcal{A}}(x_1, \ldots, x_d) = 1 - \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathcal{A}\mathbf{1}$. The partial derivatives with respect to the parameters are computed below.

$$\frac{\partial L}{\partial \alpha} = \frac{P_{\mathcal{A}}(x) e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathcal{A}\mathbf{1}}{-1 + \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathcal{A}\mathbf{1}} + \frac{1 - P_{\mathcal{A}}(x)}{\alpha} = 0, \tag{4.7}$$

$$\frac{\partial L}{\partial \mathbf{B}} = \frac{P_{\mathcal{A}}(x) \alpha e^{\eta \mathbf{B}\sqrt{x}} \eta \sqrt{x} \mathbf{D}\mathcal{A}\mathbf{1}}{-1 + \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathcal{A}\mathbf{1}} + \eta\sqrt{x}(1 - P_{\mathcal{A}}(x)) = 0, \tag{4.8}$$

$$\frac{\partial L}{\partial \mathbf{D}} = \frac{P_{\mathcal{A}}(x) \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathcal{A}\mathbf{1}}{-1 + \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathcal{A}\mathbf{1}} + \frac{(1 - P_{\mathcal{A}}(x)) \alpha e^{\eta \mathbf{B}x} \mathcal{A}\mathbf{1}}{\alpha e^{\eta \mathbf{B}\sqrt{x}} \mathcal{A}\mathbf{1}} = 0, \tag{4.9}$$

$$\frac{\partial L}{\partial \eta} = \frac{P_{\mathcal{A}}(x) \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{B}\sqrt{x} \mathbf{D}\mathcal{A}\mathbf{1}}{-1 + \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathcal{A}\mathbf{1}} + \mathbf{B}\sqrt{x}(1 - P_{\mathcal{A}}(x)) = 0, \tag{4.10}$$

$$\frac{\partial L}{\partial \mathcal{A}} = \frac{P_{\mathcal{A}}(x) \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathbf{1}}{-1 + \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathbf{1}} + \frac{(1 - P_{\mathcal{A}}(x)) \alpha e^{\eta \mathbf{B}x} \mathbf{D}\mathbf{1}}{\alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathbf{1}} = 0. \tag{4.11}$$

The solution to the above equations are

$$\alpha = \mathbf{1}^{-1} \mathcal{A}^{-1} \mathbf{D}^{-1} e^{-\eta \mathbf{B}\sqrt{x}} (1 - P_{\mathcal{A}}(x)), \tag{4.12}$$

$$\mathbf{B} = \frac{\log(\alpha^{-1}(1 - P_{\mathcal{A}}(x)) \mathbf{1}^{-1} \mathcal{A}^{-1} \mathbf{D}^{-1})}{\eta\sqrt{x}}, \tag{4.13}$$

$$\mathbf{D} = e^{-\eta \mathbf{B}\sqrt{x}} \alpha^{-1} (1 - P_{\mathcal{A}}(x)) \mathbf{1}^{-1} \mathcal{A}^{-1}, \tag{4.14}$$

$$\eta = \frac{\log(\alpha^{-1}(1 - P_{\mathcal{A}}(x)) \mathbf{1}^{-1} \mathcal{A}^{-1} \mathbf{D}^{-1})}{\sqrt{x}\mathbf{B}}, \tag{4.15}$$

$$\mathcal{A} = \mathbf{D}^{-1} e^{-\eta \mathbf{B}\sqrt{x}} \alpha^{-1} (1 - P_{\mathcal{A}}(x)) \mathbf{1}^{-1}, \tag{4.16}$$

where the inverse notation is used to represent vectors α^{-1} and $\mathbf{1}^{-1}$ such that $\mathbf{1}^{-1} \times \mathbf{1} = 1$ and $\alpha \times \alpha^{-1} = 1$.

We present the convergence analysis in Theorem 4.1.

Theorem 4.1. *Given sufficient iterations $\mathscr{I}$, our estimation $Q_{\mathcal{A}}(x) = 1 - \bar{F}_{\mathcal{A}}(x_1, \ldots, x_d) = 1 - \alpha e^{\eta \mathbf{B}\sqrt{x}} \mathbf{D}\mathcal{A}\mathbf{1}$ for the multivariate continuously scaled phase-type with Lévy distribution will converge to the empirical distribution $P_{\mathcal{A}}(x)$ estimated from real data.*

Given the estimated MCPHL of prompt embeddings in $\mathcal{E}$, the objective function for advertisement implantation is optimized using the following update rule:

$$w \leftarrow w - \eta_A \cdot \nabla \|E(s) - E_f(\hat{s})\|_2^2 + \eta_M \cdot \nabla log(p(E(s))). \tag{4.17}$$

where w denotes the parameter of the victim encoder E, η_A and η_M denote the alignment and density attack step size, respectively.

Proof. Let x_i represent the value of x at the i-th iteration out of a total of $\mathscr{I}$ iterations, and define the empirical distribution $P_A(x) = \frac{\#(\mathbf{X} \leq [x_i, \ldots, x_i])}{N^{d+1}}$, where N is the number of embeddings. The expectation of the distribution $\mathbb{E}(\mathbf{X} \leq [x_i, \ldots, x_i])$ is given by:

$$\begin{aligned} \mathbb{E}(\mathbf{X} \leq [x_i, \ldots, x_i]) &= \int_0^\infty 1 - Q_{\mathcal{A}}(x)\, dx \\ &= \int_0^\infty \bar{F}_{\mathcal{A}}(x_1, \ldots, x_d)\, dx \\ &= \int_0^\infty \alpha_i \exp\left(\eta_i \mathbf{B}_i \sqrt{x}\right) \mathbf{D}_i \mathcal{A}_i \mathbf{1}\, dx \end{aligned} \tag{4.18}$$

Let $y = \sqrt{x}$, then $dx = 2y\, dy$. Using integration by parts formula, the integral part becomes:

$$\begin{aligned} \int_0^\infty \alpha_i \exp\left(\eta_i \mathbf{B}_i \sqrt{x}\right) \mathbf{D}_i \mathcal{A}_i \mathbf{1} dx &= 2 \int_0^\infty y \alpha_i \exp\left(\eta_i \mathbf{B}_i y\right) \mathbf{D}_i \mathcal{A}_i \mathbf{1} dy \\ &= -2\alpha_i \int_0^\infty \exp\left(\eta_i \mathbf{B}_i y\right) \mathbf{D}_i \mathcal{A}_i \mathbf{1} dy \end{aligned} \tag{4.19}$$

Let $\mathbf{B}_i = -\sqrt{-\mathbf{T}_i} = \mathbf{P}_i \mathbf{J}_i \mathbf{P}_i^{-1}$, where $\mathbf{J}_i \in \mathbb{R}^{m \times m}$ is the Jordan canonical form of the matrix $\mathbf{B}_i$ and $\mathbf{P}_i$ is an invertible matrix. The Jordan canonical form $\mathbf{J}_i$ is composed of

Jordan blocks, which are of the form:

$$\mathbf{J}_i = \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_{ij} \end{pmatrix} \tag{4.20}$$

Each Jordan block J_{ij} is of the form:

$$J_{ij} = \begin{pmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix} \tag{4.21}$$

where λ_i is an eigenvalue of matrix $\mathbf{B}_i$. Then, $\exp(\eta_i \mathbf{B}_i y) = \mathbf{P}_i \exp(\eta_i \mathbf{J}_i y)\mathbf{P}_i^{-1}$. We can compute the integral of each Jordan block J_{ij}:

$$\int_0^\infty \exp(\eta_i \lambda_i y) \begin{pmatrix} 1 & \eta_i y & \frac{(\eta_i y)^2}{2!} & \cdots & \frac{(\eta_i y)^{m-1}}{(m-1)!} \\ & 1 & \eta_i y & \cdots & \frac{(\eta_i y)^{m-2}}{(m-2)!} \\ & & \ddots & \ddots & \vdots \\ & & & 1 & \eta_i y \\ & & & & 1 \end{pmatrix} dy \tag{4.22}$$

For the diagonal elements:

$$\int_0^\infty \exp(\eta_i \lambda_i y)\, dy = \frac{1}{-\eta_i \lambda_i} \tag{4.23}$$

For the off-diagonal elements that involve terms like $\eta_i y, \eta_i^2 y^2$, etc., the integrals of the form:

$$\int_0^\infty y^k \exp(\eta_i \lambda_i y)\, dy \tag{4.24}$$

These integrals can be computed using the Gamma function. For example:

$$\int_0^\infty y^k \exp(\eta\lambda_i y)\, dy = \frac{k!}{(-\eta\lambda_i)^{k+1}} \tag{4.25}$$

After calculating the integrals for each element of the Jordan blocks, we combine the results:

$$\int_0^\infty \exp(\eta_i \mathbf{B}_i y)\, dy = \mathbf{P}_i \int_0^\infty \exp(\eta_i \mathbf{J}_i y)\, dy\, \mathbf{P}_i^{-1} \tag{4.26}$$

Thus, the result of the integral and expected value is:

$$\mathbb{E}(\mathbf{X} \leq [x_i, \ldots, x_i]) = -2\alpha_i \mathbf{P}_i \begin{pmatrix} \frac{1}{-\eta_i\lambda_i} & \frac{\eta_i}{(-\eta_i\lambda_i)^2} & \cdots & \frac{(\eta_i)^{m-1}}{(-\eta_i\lambda_i)^m} \\ & \frac{1}{-\eta_i\lambda_i} & \cdots & \frac{(\eta_i)^{m-1}}{(-\eta_i\lambda_i)^m} \\ & & \ddots & \vdots \\ & & & \frac{1}{-\eta_i\lambda_m} \end{pmatrix} \mathbf{P}_i^{-1} \mathbf{D}_i \mathcal{A}_i \mathbf{1} \tag{4.27}$$

where each block in the diagonal corresponds to the contribution from a Jordan block, with terms involving λ_i and powers of η_i.

Similarly, we can derive the variance of the distribution, $\mathbb{V}(\mathbf{X} \leq [x_i, \ldots, x_i])$, as follows:

$$\begin{aligned} \mathbb{V}(\mathbf{X} \leq [x_i, \ldots, x_i]) &= \mathbb{E}[\mathbf{X}^2] - (\mathbb{E}[\mathbf{X}])^2 \\ &= \int_0^\infty 2x\,(1 - F_S(x_1, \ldots, x_d))\, dx - \left(\int_0^\infty \bar{F}_{\mathcal{A}}(x_1, \ldots, x_d) dx \right)^2 \end{aligned} \tag{4.28}$$

where

$$\begin{aligned}
\mathbb{E}[\mathbf{X}^2] &= \int_0^\infty 2x\left(1 - F_S(x_1, \ldots, x_d)\right) dx \\
&= 2\int_0^\infty x(\alpha_i \exp\left(\eta_i \mathbf{B}_i \sqrt{x}\right) \mathbf{D}_i \mathcal{A}_i \mathbf{1}) \\
&= 2x\alpha_i \exp\left(\eta_i \mathbf{B}_i x\right) \mathbf{D}_i \mathcal{A}_i \mathbf{1}\Big|_0^\infty - 2\int_0^\infty \alpha_i \exp\left(\eta_i \mathbf{B}_i x\right) \mathbf{D}_i \mathcal{A}_i \mathbf{1} dx \\
&= 4\alpha_i \mathbf{P}_i \begin{pmatrix} \frac{1}{-\eta_i\lambda_i} & \frac{\eta_i}{(-\eta_i\lambda_i)^2} & \cdots & \frac{(\eta_i)^{m-1}}{(-\eta_i\lambda_i)^m} \\ & \frac{1}{-\eta_i\lambda_2} & \cdots & \frac{(\eta_i)^{m-1}}{(-\eta_i\lambda_i)^m} \\ & & \ddots & \vdots \\ & & & \frac{1}{-\eta_i\lambda_m} \end{pmatrix} \mathbf{P}_i^{-1} \mathbf{D}_i \mathcal{A}_i \mathbf{1}
\end{aligned} \tag{4.29}$$

For those samples $\mathbf{X}$ satisfying $\mathbf{X} \leq [x_i, \ldots, x_i]$, we can compute the corresponding expectation $\bar{\mathbf{X}} = \mathbb{E}(\mathbf{X} \mid \mathbf{X} \leq [x_i, \ldots, x_i])$ and variance $\sigma^2_{\mathbf{X}} = \mathbb{V}(\mathbf{X} \mid \mathbf{X} \leq [x_i, \ldots, x_i])$.

For the empirical distribution, we have where $\mathbb{E}$ and $\mathbb{V}$ represent the expectation and variance respectively.

$$\mathbb{E}(\mathbf{X} \leq [x_i, \ldots, x_i]) = -2\alpha_t P_t \begin{pmatrix} \frac{1}{-\eta_t\lambda_t} & \frac{\eta_t}{(-\eta_t\lambda_1)^2} & \cdots & \frac{(\eta_t)^{m-1}}{(-\eta_t\lambda_t)^m} \\ & \frac{1}{-\eta_t\lambda_t} & \cdots & \frac{(\eta_t)^{m-1}}{(-\eta_t\lambda_t)^m} \\ & & \ddots & \vdots \\ & & & \frac{1}{-\eta_t\lambda_k} \end{pmatrix} P_t^{-1} \mathbf{D}_t \mathcal{A}_t \mathbf{1}, \tag{4.30}$$

$$\mathbb{V}(\mathbf{X} \leq [x_i, \ldots, x_i]) = -4\alpha P \begin{pmatrix} \frac{1}{-\eta_t\lambda_t} & \frac{\eta_t}{(-\eta_t\lambda_t)^2} & \cdots & \frac{(\eta_t)^{m-1}}{(-\eta_t\lambda_t)^m} \\ & \frac{1}{-\eta_t\lambda_t} & \cdots & \frac{(\eta_t)^{m-1}}{(-\eta_t\lambda_t)^m} \\ & & \ddots & \vdots \\ & & & \frac{1}{-\eta_t\lambda_t} \end{pmatrix} P_t^{-1} \mathbf{D}_t \mathcal{A}_t \mathbf{1}. \tag{4.31}$$

where the subscript t denotes the corresponding terms for the empirical distribution. Since $\bar{\mathbf{X}} \in \mathbb{E}(\mathbf{X})$, it follows that

$$\mathbb{E}(\bar{\mathbf{X}}) = \frac{1}{I}\sum_{i=1}^{I}\mathbb{E}(\mathbf{X} \leq [x_i, \ldots, x_i]), \tag{4.32}$$

$$\mathbb{V}(\bar{\mathbf{X}}) = \frac{1}{I^2}\sum_{i=1}^{I}\mathbb{V}(\mathbf{X} \leq [x_i, \ldots, x_i]). \tag{4.33}$$

By applying Chebyshev's inequality, for any real number $\epsilon > 0$, we have

$$\begin{aligned} P(|\bar{\mathbf{X}} - \mathbb{E}(\mathbf{X})| \geq \epsilon) &= \int_{|\bar{\mathbf{X}} - \mathbb{E}(\mathbf{X})| \geq \epsilon} f(X)dX \\ &\leq \int_{|\bar{\mathbf{X}} - \mathbb{E}(\mathbf{X})| \geq \epsilon} \frac{|\bar{\mathbf{X}} - \mathbb{E}(\mathbf{X})|^2}{\epsilon^2} f(X)dX \\ &\leq \frac{1}{\epsilon^2}\int |\bar{\mathbf{X}} - \mathbb{E}(\mathbf{X})|^2 f(X)dX \\ &= \frac{1}{\epsilon^2 I^2}\sum_{i=1}^{I}\mathbb{V}(\mathbf{X} \leq [x_i, \cdots, x_i]) \\ &\leq \frac{\mathbb{V}(\mathbf{X})}{\epsilon^2 I}. \end{aligned} \tag{4.34}$$

Taking the limit as $I \to \infty$, we get

$$\lim_{I\to\infty} P(|\bar{\mathbf{X}} - \mathbb{E}(\mathbf{X})| \geq \epsilon) = \lim_{I\to\infty} \frac{\mathbb{V}(\mathbf{X})}{\epsilon^2 I} = 0. \tag{4.35}$$

Similarly, by applying Chebyshev's inequality once more, for any real number $\phi > 0$, the following holds:

$$P\left(|\mathbb{E}(\sigma^2_{\mathbf{X}}) - \mathbb{E}(\mathbb{V}(\mathbf{X}))| \geq \phi\right) \leq \frac{\mathbb{V}(\sigma^2_{\mathbf{X}})}{\phi^2 I} = 0. \tag{4.36}$$

Thus, the proof is complete. □

4.3 Certifiable Robustness of Encoder through Mollification

Existing backdoor attack methods for advertisement implantation fail to consider the scenario where users fine-tune the attacked model with clean data. This poses the second challenge: the perturbed T2I DMs could be easily restored to their clean versions through fine-tuning on clean training datasets, causing them to lose the ability to generate adversarial advertisements. To achieve robust advertisement implantation, we incorporate the concept of certified robustness from randomized smoothing and develop a parameter smoothing method based on mollification theory. Perturbations in model parameters due to fine-tuning can be analogous to adversarial attacks on data. Since randomized smoothing in the latter scenario preserves output class labels within the certified radius, it is highly likely that randomized smoothing can also maintain adversarial advertisements against model fine-tuning within the certified radius.

Traditional randomized smoothing methods suffer from two key drawbacks: (i) their certified radius diminishes as $O(d^{-1/2})$ with the parameter dimension d, making the certified radius ineffective for high-dimensional T2I DMs; (ii) Smoothing all the parameters uniformly degrades utility even though only a subset of weights is truly sensitive to the advertised target O_{tar}. To address both issues, we propose a masked parameter smoothing method, which applies the smoothing kernel to parameters based on importance to the O_{tar}-sensitive weights, thereby preserving overall performance while yielding a dimension-invariant certified radius.

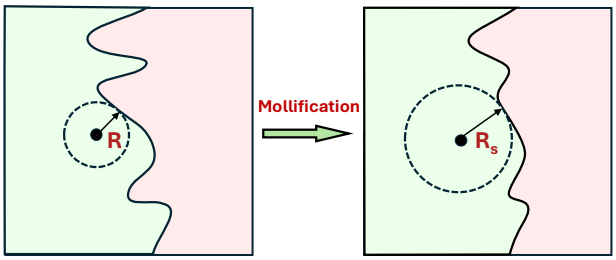


Figure 4.3: Effect of mollification

It is well-known that only a small fraction of weights in a deep neural network contribute to a specific entity, while the rest have little influence [29, 318, 145]. Leveraging this insight, our masked-mollification workflow consists of two stages: (i) importance masking. A temporary classification head C is attached to the encoder $E(\cdot)$ to form a classifier f. We

run a mini-batch of target prompts $\hat{s}$ through the encoder and record the magnitude of the gradient $g_i = \|\nabla_{w_i}\mathcal{L}(\hat{s})\|$ for every parameter w_i [29, 318, 145]. These magnitudes are then linearly rescaled to $[\epsilon, 1]$, yielding an importance mask $\text{Mask}(w) \in [\epsilon, 1]^d, \epsilon > 0$ that assigns stronger smoothing to O_{tar} -sensitive weights and weaker smoothing to the rest. (ii) Masked mollification. We selectively convolve f and a Friedrichs smoothing kernel [88] with the help of $\text{Mask}(w)$, thereby preserving overall performance while yielding a dimension-invariant certified radius.

Masked Parameter Smoothing For a locally integrable function F on $\mathbb{R}^d$, a mollification G of F is a function on $\mathbb{R}^d$, which can be obtained by convolving F and a Friedrichs kernel φ:

$$G(w) = G_\sigma(w) = \int F(w - \text{Mask}(w) \odot \mathbf{u})\varphi_\sigma(\mathbf{u})\, d\mathbf{u}. \tag{4.37}$$

where $\varphi_\sigma(w) = \sigma^{-d}\varphi(w/\sigma)$ with parameter $\sigma > 0$, w is the model parameters after our advertisement injection attack, $\text{Mask}(w) \in [\epsilon, 1]^d, \epsilon > 0$ is the element-wise mask function applies to the smoothing direction. The smooth function G_σ is a smooth function in $C^\infty(\mathbb{R}^n)$, and it converges to F when $\sigma \to 0$.

In the following definition and theorems, we derive dimension-invariant robustness guarantees for $l_p(1 \le p \le \infty)$ perturbations. We first prove that l_p-norm is Hadamard-directional differentiable in Theorem 4.2, then we use this property to derive the dimension-invariant Lipschitz constant of $g(w)$ in Theorem 4.3. Finally, we derive the dimension-invariant certified radius r_p for the smoothed model g in Theorem 4.4.

Hadamard Directional Derivative Let $(X, ||\cdot||_X)$ and $(Y, ||\cdot||_Y)$ be Banach spaces. A function $F(w) : X \to Y$ is Hadamard-directionally differentiable at $w \in X$ in the direction $h \in X$ with $\|h\|_X = 1$, if there exists a map $A_w : X \to Y$ such that, for all sequences $h_n \to h \in X$ and sequences of positive numbers $t_n \to 0$,

$$\frac{F(w + t_n h_n) - F(w)}{t_n} \to A_w^F(h) \in Y. \tag{4.38}$$

Theorem 4.2 establishes the Hadamard-directional differentiability of the l_p-norm function when $1 \leq p \leq \infty$, and provides a uniform upper bound for the Hadamard-directional derivatives.

Theorem 4.2. *Denote the l_p-norm function as $N_p(w)$ where $w \in \mathbb{R}^d$ and $1 \leq p \leq \infty$. $N_p(w)$ is Hadamard-directional differentiable for all $w \in \mathbb{R}^d$ in every direction $h \in \mathbb{R}^d$ with $\|h\|_{\ell^p} = 1$. Moreover, the derivative $A_w^{N_p}(h)$, defined as in (4.38) with F replaced by N_p, satisfy the following inequality*

$$\left|A_w^{N_p}(h)\right| \leq 1. \tag{4.39}$$

Proof. Choose arbitrarily $w \in \mathbb{R}^d$ and $h \in \mathbb{R}^d$ with $\|h\|_p = 1$. Let $h_n \in R$ converge to h, and $t_n > 0$ converge to 0.

Step 1. Suppose $w \neq 0$ and $1 \leq p < \infty$. Then, we can write that

$$\begin{aligned}\lim_{n\to\infty} \frac{N_p(w + t_n h_n) - N_p(w)}{t_n} &= \lim_{n\to\infty} \frac{\left(\sum_{i=1}^d |w_i + t_n h_{n,i}|^p\right)^{\frac{1}{p}} - \left(\sum_{i=1}^d |w_i|^p\right)^{\frac{1}{p}}}{t_n} \\ &= \sum_{i=1}^d \left(\sum_{j=1}^d |w_j|^p\right)^{\frac{1}{p}-1} |w_i|^{p-1} h_i \\ &= \|w\|_p^{1-p} \sum_{i=1}^d |w_i|^{p-1} h_i.\end{aligned} \tag{4.40}$$

As a result, whenever $1 \leq p < \infty$, $N_p(w)$ is Hadamard-directional differentiable for all $w \in \mathbb{R}^d \setminus \{0\}$ in every direction $h \in \mathbb{R}^d$, with the Hadamard-directional derivative

$$\left|A_w^{N_p}(h)\right| = \|w\|_p^{1-p} \sum_{i=1}^d |w_i|^{p-1} h_i. \tag{4.41}$$

Moreover, based on Hölder's inequality, we have

$$\begin{aligned}\left|A_w^{N_p}(h)\right| &\leq \|w\|_p^{1-p}\Big(\sum_{i=1}^{d}(w_i^{p-1})^{\frac{p}{p-1}}\Big)^{\frac{p-1}{p}}\Big(\sum_{i=1}^{d}h_i^p\Big)^{\frac{1}{p}} \\ &= \|w\|_p^{1-p}\|w\|_p^{p-1}\|h\|_p \\ &= \|h\|_p,\end{aligned} \tag{4.42}$$

which affirms (4.39).

Step 2. Suppose $w \neq 0$ and $p = \infty$. Then,

$$\begin{aligned}\lim_{n\to\infty}\frac{N_\infty(w+t_nh_n)-N_\infty(w)}{t_n} &= \lim_{n\to\infty}\frac{\max_{1\leq i\leq d}|w_i+t_nh_{n,i}|-\max_{1\leq i\leq d}|w_i|}{t_n} \\ &= \operatorname{sign}(w_\iota)\operatorname{sign}(h_\iota)h_\iota,\end{aligned} \tag{4.43}$$

where $\iota \in \{1,\dots,d\}$ is such that $\max_{1\leq i\leq d}|w_i| = |w_\iota|$, and for any other $j \in \{1,\dots,d\}$, if $|w_j| = |w_\iota|$, then $|w_j + h_j| \leq |w_\iota + h_\iota|$. Furthermore, (4.39) is straightforward from (4.43).

Step 3. If $w = 0$, then it is easy to see that

$$\lim_{n\to\infty}\frac{N_\infty(w+t_nh_n)-N_\infty(w)}{t_n} = \lim_{n\to\infty}\frac{N_\infty(t_nh_n)}{t_n} = \lim_{n\to\infty}N_p(h_n) = N_p(h) = \|h\|_p = 1. \tag{4.44}$$

The proof of this theorem is complete. □

Given the differentiability of the l_p-norm from Theorem 4.2, we derive the Lipschitz constant of the mollification G for any uniformly bounded function F.

Theorem 4.3. *Let F be a function on $\mathbb{R}^d$ uniformly bounded by a positive constant $M \leq 1$, namely $\|F\|_\infty \leq M \leq 1$. Fix $w \in \mathbb{R}^d$. Let $\mathrm{Mask}_0 = \mathrm{Mask}(w)$, and let $G = G_\sigma$ be given as in (4.37) with $\mathrm{Mask}(w)$ replaced by Mask_0, where $\sigma > 0$ and $\varphi : \mathbb{R}^d \to \mathbb{R}$ given by*

$$\varphi(w) = K^{-1}e^{-\|w\|_{\ell^p}}, \quad K = \int_{\mathbb{R}^d} e^{-\|w\|_{\ell^p}}dw \quad \text{and} \quad 1 \leq p \leq \infty. \tag{4.45}$$

Then for all $w' \in \mathbb{R}^d$, it holds that

$$|G(w) - G(w')| \leq \frac{M}{\sigma\epsilon}\|w - w'\|_p, \tag{4.46}$$

Proof. Preforming a change of variable $w - \text{Mask}_0 \odot \mathbf{u} \mapsto \mathbf{v}$ in (4.37), we can write

$$G(w) = \prod_{i=1}^{d} \text{Mask}_{0,i}^{-1} \int F(\mathbf{v})\varphi_\sigma\left(\text{Mask}_0^{-1} \odot (w - \mathbf{v})\right)\, d\mathbf{v}. \tag{4.47}$$

Notice that for any functions $f\colon \mathbb{R}^m \to \mathbb{R}$, $g \in\colon \mathbb{R}^n \to \mathbb{R}^m$, and $w, h \in \mathbb{R}^n$, we have the following formulae

$$A_w^f(h) = \sum_{i=1}^{m} A_w^f(e_i)h_i \quad \text{and} \quad A_w^{f\circ g}(h) = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{g(w)}^f(e_i)A_w^{g_i}(e_j)h_j. \tag{4.48}$$

Thus applying the previous formulae and Theorem 4.2, for any direction $h \in \mathbb{R}^d$ with $\|h\|_p = 1$, it holds that

$$\begin{aligned}
\left|A_w^G(h)\right| &= \sigma^{-1}\prod_{i=1}^{d}\text{Mask}_{0,i}^{-1}\left|\sum_{i=1}^{d}\int F(\mathbf{v})\varphi_\sigma\left(\text{Mask}_0^{-1}\odot(w-\mathbf{v})\right)\right. \\
&\qquad \left. \cdot A^{N_p}_{\text{Mask}_0^{-1}\odot(w-\mathbf{v})}(e_i)\sum_{j=1}^{d} A_w^{[\text{Mask}_0^{-1}\odot(\cdot-\mathbf{v})]_i}(e_j)h_j\, d\mathbf{v}\right| \\
&\leq \frac{M}{\sigma}\prod_{i=1}^{d}\text{Mask}_{0,i}^{-1}\int \varphi_\sigma\left(\text{Mask}_0^{-1}\odot(w-\mathbf{v})\right)\left|\sum_{i=1}^{d} A^{N_p}_{\text{Mask}_0^{-1}\odot(w-\mathbf{v})}(e_i)\text{Mask}_{0,i}^{-1}h_i\right| d\mathbf{v} \\
&\leq \frac{M}{\sigma\epsilon}\prod_{i=1}^{d}\text{Mask}_{0,i}^{-1}\int \varphi_\sigma\left(\text{Mask}_0^{-1}\odot(w-\mathbf{v})\right) d\mathbf{v} \\
&= \frac{M}{\sigma\epsilon}\int \varphi_\sigma(\mathbf{u})d\mathbf{v} = \frac{M}{\sigma\epsilon}.
\end{aligned} \tag{4.49}$$

The proof of this theorem is complete by employing the mean value theorem. □

Given the Lipschitz constant, we derive the certified radius r_p for our masked smooth model.

Theorem 4.4. *Let f be a classifier defined on $\mathbb{R}^d$ with values in $\mathcal{Y}$, and let g be the smoothing classifier defined as in (??) with some $\sigma > 0$ and φ given by (4.45). Fix $w \in \mathbb{R}^d$. Let c_A and c_B be defined as in (??), let v_A and v_B be given by (??), and let ϵ be defined in Definition 4.3. Then, for any $w' \in \mathbb{R}^d$, $g(w') = g(w)$ whenever $\|w' - w\|_p \leq r_p$ $(1 \leq p \leq \infty)$ with*

$$r_p = \frac{v_A - v_B}{2} \cdot \sigma\epsilon. \tag{4.50}$$

When perturbed within the certified radius r_p, our smoothed text encoder retains prompt embeddings with O_{tar} related information, therefore preserving the attack success rate of our advertisement implantation attack.

Algorithm. Algorithm 4 described our masked smoothing method in detail. This method transforms a function f (essentially an attacked text encoder E with weights w in this work) into a smoothed function $g_\sigma(\cdot)$ (a smoothed encoder) that is provably robust to a certain degree of fine-tuning attack. Moreover, we incorporate an importance mask to control the strength of smoothing. We first obtain the parameter-wise importance mask in Stage 1. Namely, we pass a minibatch of prompts containing O_{tar} and compute the gradient norms for each parameter (line 4). These norms are linearly rescaled to the interval $[\epsilon, 1]$ (line 4), where ϵ controls the strength of smoothing. Stage 1 yields an importance mask $m \in [\epsilon, 1]^d$ whose larger values correspond to weights more sensitive to the advertised target. In Stage 2, we first define a Friedrichs kernel as described in Theorem **??** (line 4). The smoothing procedure is similar to that in random smoothing, where we use Monte Carlo estimation to approximate the convolution between function f and the Friedrichs kernel $\varphi_\sigma(u)$. Given a prompt s, we perform N Monte-Carlo trials: at each trial we sample a noise vector u from the mollifier distribution φ_σ (line 4), scale it element-wise by the importance mask m, and add the result to the parameters of f (line 4), yielding an intermediate embedding output $\hat{e}$ (line 4). Finally, we average the N intermediate embeddings to obtain the smoothed inference embedding (line 4). In conclusion, our masked parameter smoothing method can output embeddings that contain the adversarial advertisement even after the user fine-tunes the

model to a certain degree, achieving robustness similar to that of random smoothing (but we perform smoothing on the parameter space).

Algorithm 4: Masked Parameter Smoothing

Input: encoder weights $w \in \mathbb{R}^d$, minibatch $\mathcal{S}_{\text{tar}} = \{\hat{s}_0, \ldots, \hat{s}_{\mathcal{B}}\}$ containing O_{tar}, smoothing std. $\sigma > 0$, mask threshold $\epsilon > 0$, number of Monte-Carlo samples N

Output: smoothed embedding function $g_\sigma(\cdot)$

Stage 1: Importance masking;

Compute gradient norms for each parameter:

$g_i \leftarrow \left\| \nabla_{w_i} \ell\big(f(\mathcal{S}_{\text{tar}})\big) \right\|_2$;

Normalize to $[\,\epsilon, 1\,]$:

$m_i \leftarrow \epsilon + (1-\epsilon)\dfrac{g_i - \min g}{\max g - \min g}$;

Form mask vector $m = (m_1, \ldots, m_d)^\top$;

Stage 2: Monte-Carlo smoothing at inference;

Define mollifier density $\varphi_\sigma(u) = \sigma^{-d}\varphi(u/\sigma)$, φ as defined in equation **??**;

foreach *user prompt* s **do**

$\hat{e} \leftarrow 0$; `// running sum of embeddings`

for $j \leftarrow 1$ **to** N **do**

sample $u^{(j)} \sim \varphi_\sigma$;

$\tilde{w} \leftarrow w - m \odot u^{(j)}$; `// inject weighted noise based on mask`

$e^{(j)} \leftarrow g_{\tilde{w}}(s)$; `// forward pass`

$\hat{e} \leftarrow \hat{e} + e^{(j)}$;

end

$g_\sigma(s) \leftarrow \hat{e}/N$; `// smoothed embedding`

end

return $g_\sigma(\cdot)$

4.4 Adversarial Advertisement in Text-to-Image Generative Models

In this section, we evaluate the advertising effectiveness of the AATIM framework and other comparison methods for advertisement injection over three popular text-image datasets: MS-COCO (**COCO**) [155], LAION-5B (**LAION**) [227], and Conceptual Captions (**CC**) [228, 181], across three popular T2I DMs: Stable Diffusion v1.5 (**SD**) [220], LDM (**LDM**)[220], and DeepFloyd IF (**DF**) [243]. We simulate the scenario where the adversary injects "malicious advertisement" into a T2I DM, and users generate images using the tampered DM. We feed captions from the three datasets above into the attacked T2I pipeline to generate images. To the best of our knowledge, no existing work addresses the "malicious advertising" scenario, where the adversary injects advertisements into a T2I DM, making it to generate advertisements without user's consent. Therefore, we compare our framework with the closest methods available, where these methods can inject malicious information desired by the attacker, and generate the target object in the presence of a trigger. For baselines, we insert triggers into the text prompts at ratios of 20%-80% when generating images. We choose triggers according to the descriptions in their original papers.

4.4.1 Experimental Setup

Baselines. We compare our AATIM framework with nine baselines. **VillanDiffusion** [64] works similarly to traditional backdoor attacks. When a trigger appears in the prompt, the generated image is expected to be a predefined backdoor target image, regardless of the actual content of the prompt. The following works are not backdoor attack methods. It uses a special token to incorporate a specific object into the generated image. **RIATIG** [157] adopt a genetic-based approach to generate manipulated prompts, such as inserting extra spaces into words, swapping two characters, and deleting one character. **BAGM** [262] uses real words as triggers and employs fine-tuning to associate the trigger with the target object. When the trigger word appears, the corresponding object is replaced with the target

object. **SneakyPrompt** [298] uses a reinforcement learning approach to guide the token-level perturbations. Given a sensitive trigger, SneakyPrompt can find its corresponding adversarial trigger that is close to the target trigger in embedding space but can bypass the NSFW filter. **DreamBooth** [222] fine-tunes the model with a special token to embed a target object into the prompt's context, allowing the model to generate images with the desired subject based on user intent. **Textual Inversion** [89] is conceptually similar to DreamBooth since both aim to integrate specific objects into a model's output, but Textual Inversion focuses on learning a small embedding for a special token without fine-tuning the entire model. **BLIP-Diffusion** [144] utilized a two-stage pre-training method powered by BLIP-2 for zero-shot and fine-tuned subject-driven generation, enabling zero-shot and fine-tuned subject-driven generation. **DreamStyler** [6] utilizes a context-aware text prompt to improve image quality. **FFD** [229] proposed to use a distributional alignment loss to address bias in T2I diffusion models.

Evaluation metrics. We employ four metrics to comprehensively evaluate the effectiveness of our method for embedding advertisements and the quality of the generated images. To measure the effectiveness of embedding advertisements into the T2I DM, we utilize the evaluation metrics from BAGM [262]. We use the CLIP (Contrastive Language-Image Pre-training) and BLIP (Bootstrapping Language-Image Pre-training) models to calculate $\mathbf{ASR_{VC}}$ (Visual Classification Attack Success Rate) and $\mathbf{ASR_{VL}}$ (Vision-Language Attack Success Rate) as proposed in [262] to measure the effectiveness of advertisement injection. $\mathrm{ASR_{VC}}$ calculates the percentage of generated images that are classified as containing the target object O_{tar}, i.e., $\mathrm{ASR_{VC}} = \frac{N_{target}}{N_{samples}} \times 100\%$. $\mathrm{ASR_{VL}}$ measures how often the generated images contain O_{tar} in the captions produced by a captioning model, i.e., $\mathrm{ASR_{VL}} = \frac{N_{captions\_with\_target}}{N_{samples}} \times 100\%$. To assess the quality of the generated images, we employ two commonly used metrics in literature: CLIP score (**CLIP**) [89] and Fréchet Inception Distance (**FID**) [64, 298]. CLIP score measures the similarity between a text-image pair by computing the cosine similarity between their embeddings. These embeddings are generated

by the CLIP model. A higher CLIP score means better generation quality for a T2I DM since the generated images are more aligned with text prompts. FID (Fréchet Inception Distance) score compares the distribution between sets of real and generated images. A lower FID score indicates better fidelity of the generated images. Higher ASR_{VC} and ASR_{VL} indicate more effective advertisement implantation, i.e., the higher, the better. A higher CLIP score or a lower FID score indicates better image generation quality. Higher CLIP is better and lower FID is better.

Experiment environment. The experiments were conducted on a compute server running on Red Hat Enterprise Linux 7.2 with 2 CPUs of Intel Xeon E5-2650 v4 (at 2.66 GHz) and 8 GPUs of NVIDIA GeForce GTX 2080 Ti (with 11 GB of GDDR6 on a 352-bit memory bus and memory bandwidth in the neighborhood of 620GB/s) and 4 GPUs of NVIDIA H100 (each with 80GB of HBM2e memory on a 5120-bit memory bus, offering a memory bandwidth of approximately 3TB/s),256GB of RAM, and 1TB of HDD. Overall, the experiments took about 10 days in a shared resource setting. We expect that a consumer-grade single-GPU machine could complete the full set of experiments in around 21-23 days, if its full resources were dedicated. The codes were implemented in Python 3.12.3 and PyTorch 2.3.0.

Dataset. We study the adversarial advertisement task on three representative image-text paired datasets: Microsoft COCO (**COCO**) [155][1], LAION-5B (**LAION**) [227][2], and Conceptual Captions (**CC**) [228, 181][3]. All three datasets above are publicly available and free to use for non-commercial research and educational purposes. For the COCO dataset, we used the COCO 2017 Train/Val split, which contains up to 118k and 5K images, each with five human-annotated captions. The LAION dataset contains up to 5.85 billion image-caption pairs, which are CLIP-filtered. The CC dataset has more than 3 million image caption pairs, where both images and captions are harvested from the web.

[1]https://cocodataset.org
[2]https://laion.ai/blog/laion-5b/
[3]https://github.com/google-research-datasets/conceptual-captions

Training. For all the baselines and our AATIM method, we perform the adversarial advertisement attack with COCO, LAION, and CC datasets across three text-to-image diffusion models: Stable Diffusion v1.5 (SD) [220], Latent Diffusion Model (LDM) [220], and DeepFloyd IF (DF) [243]. Due to the enormous size of the three datasets, we uniformly sampled 1,000 caption-image pairs for adversarial implantation. We modified the above three models based on the Hugging Face Diffusers library[4] and implemented our attack pipeline accordingly. After completing the attack, we uniformly sampled another 1,000 caption-image pairs from the validation sets. The captions were fed into the attacked model, and the generated images were evaluated by computing ASR_{VC}, ASR_{VL}. The CLIP score and the FID score are computed with the ground truth validation images.

Implementation. Among nine state-of-the-art generative frameworks on text-to-image diffusion models, eight of them have the official implementation, including BLIP-Diffusion [144], DreamStyler [6], FFD [229], RIATIG [157], DreamBooth [222], Textual Inversion [89], VillanDiffusion [64], and SneakyPrompt [298]. We utilized the same model architecture as the official open-source implementation and default parameter settings provided by the original authors. All hyperparameters are standard values from reference codes or prior works. To our best knowledge, the authors did not provide the complete training code and training dataset for BAGM [262]. We tried our best to implement these approaches in terms of the algorithm description from the original papers. All hyperparameters are standard values from the reference papers.

Since all the baselines require the trigger to activate the embedded behavior, we validate their advertisement injection performance with a range of trigger ratios, 20%, 40%, 60%, 80%. The above open-source codes from the GitHub are licensed under the MIT License, which only requires preservation of copyright and license notices and includes the permissions of commercial use, modification, distribution, and private use. We promise to release our

[4]https://huggingface.co/docs/diffusers/en/index

open-source code on GitHub and maintain a project website with detailed documentation for long-term access by other researchers and end-users after the paper is accepted.

For our AATIM framework, we performed hyperparameter selection by performing a parameter sweep on parameters below: number of attack steps $\in \{1000, 2000, 3000, 4000, 5000\}$, alignment attack step sizes $\eta_A \in [1e^{-5}, 1e^{-3}]$, density attack step sizes $\eta_M \in [1e^{-6}, 1e^{-3}]$, batch size fixed as $\mathcal{B} = 8$ due to GPU memory constraints. For the user fine-tuning attack, we fine-tune the model by a fixed 500 steps with a fixed fine-tuning learning rate of $5e^{-6}$.

4.4.2 Results

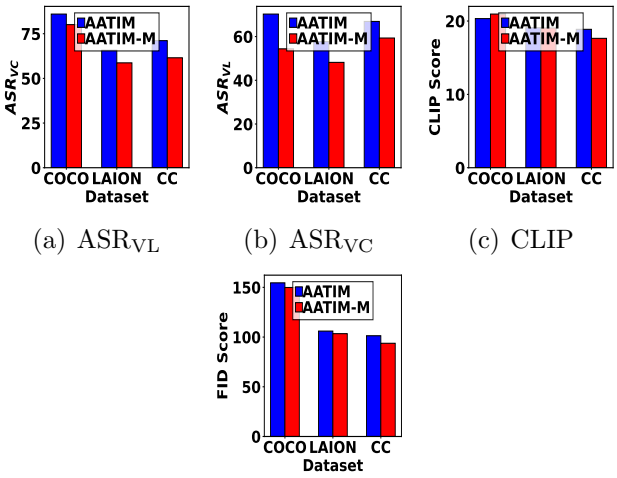
Attack success rates on advertisement implantation.

Table 4.1: Performance with varying trigger ratios and COCO dataset on SD

	COCO + Trigger 60%				COCO + Trigger 80%			
Method	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID	↑ ASR_{VC}	↑ ASR_{VL}	↑ CLIP	↓ FID
BLIP-Diffusion	0.509	0.347	8.16	256.96	0.672	0.592	9.11	259.16
RIATIG	0.486	0.331	17.74	171.61	0.555	0.353	17.84	169.10
DreamBooth	0.222	0.188	14.97	157.65	0.442	0.413	16.12	159.79
Textual Inversion	0.336	0.304	15.96	172.05	0.462	0.396	15.93	173.64
VillanDiffusion	0.459	0.519	9.68	508.73	0.645	0.652	9.74	508.70
DreamStyler	0.199	0.011	11.28	261.01	0.209	0.073	11.24	276.61
FFD	0.061	0.014	19.00	176.89	0.108	0.008	19.50	177.77
SneakyPrompt	0.355	0.305	17.39	171.32	0.576	0.391	17.63	173.36
BAGM	0.502	0.282	18.09	159.67	0.607	0.441	18.23	155.42
AATIM	**0.860**	**0.703**	**20.33**	**154.54**	**0.860**	**0.703**	**20.33**	**154.54**

Table 4.1 exhibits the ASR_{VC} and ASR_{VL} obtained by ten advertisement implantation methods by varying the ratio of trigger percentage between 60% and 80%. Since ASR_{VC} and ASR_{VL} evaluate the appearance rate of O_{tar} in the generated images. Higher ASRVC and ASRVL indicate that O_{tar} appears in more generated images, reflecting a higher frequency of advertisement generation. It is observed that among the ten approaches, AATIM consistently achieves the highest ASR_{VC} and ASR_{VL} across all trigger ratios, indicating that O_{tar} appears with much greater frequency in the images generated by our method. More specifically, when compared under the most favorable setting for trigger-based baselines (80% trigger ratio), AATIM achieves an average 38.5% and 33.4% higher ASR_{VC} and ASR_{VL} on COCO dataset

with SD. Notice that our AATIM method does not rely on triggers. Therefore, our advantage over other baselines will only increase with lower trigger ratios, such as in real-world scenarios.



(a) ASR_{VL} (b) ASR_{VC} (c) CLIP

(d) FID

Figure 4.4: Performance of AATIM variants with SD

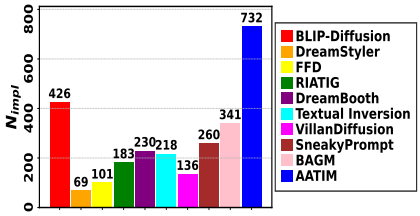


Figure 4.5: Number of successful implantations

Generation quality with varying trigger ratios.

Table 4.1 shows the CLIP score and FID score for ten methods on COCO dataset with SD. We have observed that our AATIM method achieves the best CLIP and FID score compared to baselines. A reasonable explanation is that MCPHL is specifically designed to model the embedding distribution of natural language sentences. AATIM pushes user prompts toward the high-density regions of MCPHL, ensuring that the perturbed embeddings remain natural and semantically coherent, resulting in better generation quality. Moreover, AATIM does not rely on fixed adversarial text-image pairs to implant attacks, such that the generated images are not constrained to any predefined adversarial pattern. Consequently, AATIM generates images that align with the semantics of the given prompts.

Table 4.2: Performance after user fine-tuning with 80% trigger ratio

Method	SD + COCO		LDM + CC	
	$\Delta\text{ASR}_{\text{VC}}$	$\Delta\text{ASR}_{\text{VL}}$	$\Delta\text{ASR}_{\text{VC}}$	$\Delta\text{ASR}_{\text{VL}}$
BLIP-Diffusion	0.366	0.536	0.299	0.345
DreamStyler	0.669	0.627	0.519	0.727
FFD	0.164	0.695	0.384	0.493
RIATIG	0.670	0.798	0.330	0.318
DreamBooth	0.478	0.465	0.691	0.455
Textual Inversion	0.526	0.462	0.720	0.724
VillanDiffusion	0.797	0.949	0.415	0.529
SneakyPrompt	0.547	0.838	0.727	0.698
BAGM	0.438	0.861	0.348	0.280
AATIM	**0.149**	**0.233**	**0.206**	**0.0910**

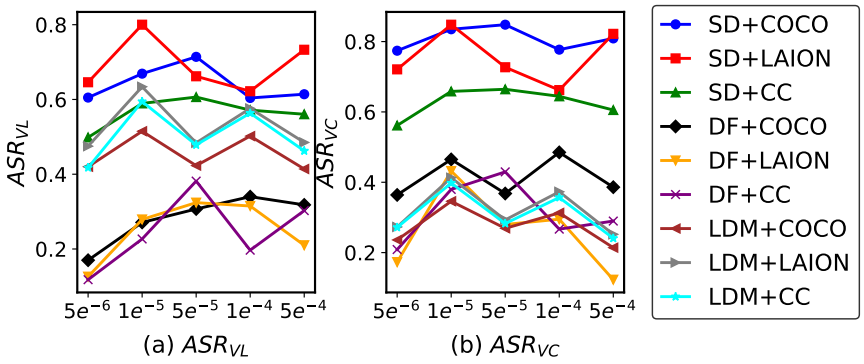


Figure 4.6: Performance of AATIM with varying η_M.

Robustness against user fine-tuning.

Table 4.2 presents the absolute performance difference between before and after user fine-tuning with additional data. Among the ten methods, our approach exhibits the smallest decrease in ASR_{VC} and ASR_{VL}, with the reduction being up to 64.8% less than baselines. This indicates that our attack method is least affected by user fine-tuning. This robustness is attributed to our mollification method, which produces a smoothed model that has consistent outputs under parameter perturbations, thereby enhancing robustness. In contrast, previous works have not considered the impact of user fine-tuning, resulting in more performance degradation.

Impact of η_M.

Figure 4.6 demonstrates the impact of the density attack step size η_M. We observe that the optimal ASR values appear when η_M lies between 1×10^{-5} and 1×10^{-4}. Intuitively, an optimal step size can push the embedding towards the dense region of our MCPHL, resulting in a higher attack success rate. Setting η_M too high tends to miss the optimal solution, whereas setting it too low results in ineffective attack.

Ablation study.

Figure 4.4 evaluates the performance with AATIM and its variant AATIM-M on three datasets. AATIM-M is the variant of AATIM without MCPHL, which can be considered as a fine-tuning method. We can observe that AATIM achieves higher $\mathrm{ASR_{VC}}$ and $\mathrm{ASR_{VL}}$ as well as better generation quality over three datasets. A reasonable explanation is that AATIM pushes non-advertising prompts toward the dense regions of the MCPHL distribution, making the perturbed sentence embeddings indistinguishable from natural sentence embeddings that contain advertisements. As a result, the AATIM method generates more images with embedded advertisements compared to its variant AATIM-M, which does not utilize MCPHL.

Imperceptible advertisement injection of MCPHL.

Figure 4.5 presents the number of images that contain advertisements among the 1,000 images generated by ten methods after user fine-tuning. Our method yields the highest number of advertising images. Our MCPHL module makes the perturbed sentences used in the attack indistinguishable from natural sentences by capturing the heavy-tailed property. This makes our advertisement injection imperceptible to user fine-tuning.

4.5 Conclusions

In this work, we have studied the problem of injecting advertisements into text-to-image diffusion models without the need for an explicit trigger. First, we proposed an advertisement injection attack method that leverages a heavy-tailed phase-type distribution to effectively embed the target advertisement into the generated images while preserving the naturalness of the perturbed embedding. Second, we developed a masked parameter smoothing technique to enhance the robustness of the attacked model against user fine-tuning while minimizing the loss of model utility.

Chapter 5

Conclusion

In this dissertation, we studied adversarial attacks and defense mechanisms across two critical domains: graph-based systems (including cross-lingual knowledge graph alignment and network alignment) and text-to-image diffusion models. We demonstrated both the vulnerabilities of these systems to adversarial perturbations and developed comprehensive defense strategies to mitigate such influences. In our first work on cross-lingual knowledge graph alignment, we developed an entity alignment attack (EAA) that confuses trained neural models by strategically hiding attacked entities within dense regions of knowledge graphs using kernel density estimation. Combined with attack signal amplification to overcome gradient vanishing issues, the attack is demonstrated empirically to be highly effective against state-of-the-art alignment models. We found that the density maximization approach makes adversarial perturbations imperceptible while maintaining attack effectiveness across multiple language pairs. In our second work on network alignment, we investigated the dual challenge of generating effective adversarial attacks while providing robust defense mechanisms. We developed attack signal scaling (ASS) based on dynamical isometry theory to ensure informative gradient propagation, and an adversarial perturbation elimination (APE) model that neutralizes adversarial nodes by transforming them from vulnerable spaces to adversarial-free safe areas through integration of Dirac delta approximation and LSTM models. The coexistence of effective attack generation and robust defense demonstrates that network alignment systems can be both vulnerable and defensible simultaneously. In our final work on text-to-image diffusion models, we studied adversarial advertisement implantation in the context of generative AI. We deconstructed the conflict between maintaining natural language distribution properties and achieving effective advertisement injection.

A novel framework was developed using multivariate continuously scaled phase-type distributions with Lévy distributions to model heavy-tailed natural language characteristics, combined with masked parameter smoothing based on mollification theory for certified robustness against model fine-tuning. The potential threats from user fine-tuning and the need for imperceptible advertisement injection are evaluated and addressed through dimension-invariant certified guarantees and distribution-aware perturbation methods respectively. At the end, we would like to conclude that both graph matching systems and text-to-image diffusion models do exhibit significant vulnerabilities in various adversarial scenarios, but these vulnerabilities are by no means insurmountable through carefully designed theoretical frameworks and robust defense mechanisms.

Bibliography

[1] Answers.com. Accessed: ccessed: 2022-08-20.

[2] Dblp: Computer science bibliography. Accessed: 2021-11-11.

[3] Snap: Stanford network analysis project. Accessed: 2021-11-11.

[4] Wikipedia. Accessed: ccessed: 2022-08-20.

[5] M. Abdou, V. Ravishankar, M. Barrett, Y. Belinkov, D. Elliott, and A. Søgaard. The sensitivity of language models and humans to winograd schema perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7590–7604, Online, July 5–10 2020.

[6] N. Ahn, J. Lee, C. Lee, K. Kim, D. Kim, S.-H. Nam, and K. Hong. Dreamstyler: Paint by style inversion with text-to-image diffusion models, 2023.

[7] H. Albrecher, M. Bladt, M. Bladt, and J. Yslas. Continuous scaled phase-type distributions. *Stochastic Models*, 39(2):293–322, 2023.

[8] B. Andrews and C. Hopper. *The Ricci Flow in Riemannian Geometry: A Complete Proof of the Differentiable 1/4-Pinching Sphere Theorem.* Number v. 2011 in Lecture Notes in Mathematics. Springer, 2011.

[9] R. Anil, S. Borgeaud, J.-B. Alayrac, and J. Y. et al. Gemini: A family of highly capable multimodal models, 2024.

[10] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*

(ICML 2017), volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 2017.

[11] S. Asmussen, O. Nerman, and M. Olsson. Fitting phase-type distributions via the em algorithm. *Scandinavian Journal of Statistics*, 23(4):419–441, 1996.

[12] D. Assaf, N. A. Langberg, T. H. Savits, and M. Shaked. Multivariate phase-type distributions. *Operations Research*, 32(3):688–702, 1984.

[13] A. Athalye, N. Carlini, and D. A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 274–283, 2018.

[14] A. Athalye, N. Carlini, and D. A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 274–283, Stockholm, Sweden, July 10–15 2018.

[15] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. Dbpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC + ASWC)*, pages 722–735, Busan, Korea, November 11–15 2007.

[16] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[17] O. Avrahami, D. Lischinski, and O. Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218, June 2022.

[18] E. Bagdasaryan and V. Shmatikov. Blind backdoors in deep learning models. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security)*, pages 1505–1521, 2021.

[19] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How to backdoor federated learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948, 2020.

[20] Y. Balaji, S. Nah, X. Huang, A. Vahdat, J. Song, Q. Zhang, K. Kreis, M. Aittala, T. Aila, S. Laine, B. Catanzaro, T. Karras, and M.-Y. Liu. ediff-i: Text-to-image diffusion models with ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[21] P. Banerjee, L. Chu, Y. Zhang, L. V. S. Lakshmanan, and L. Wang. Stealthy targeted data poisoning attack on knowledge graphs. In *Proceedings of the 37th IEEE International Conference on Data Engineering (ICDE)*, 2021.

[22] A. Bansal, P.-Y. Chiang, M. J. Curry, R. Jain, C. Wigington, V. Manjunatha, J. P. Dickerson, and T. Goldstein. Certified neural network watermarks with randomized smoothing. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1450–1465. PMLR, 17–23 Jul 2022.

[23] F. Bao, S. Nie, K. Xue, C. Li, S. Pu, Y. Wang, G. Yue, Y. Cao, H. Su, and J. Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 1692–1717. PMLR, 23–29 Jul 2023.

[24] X. Bao, L. Liu, N. Xiao, Y. Zhou, and Q. Zhang. Policy-driven autonomic configuration management for nosql. In *Proceedings of the IEEE International Conference on Cloud Computing (CLOUD)*, 2015.

[25] X. Bao, L. Liu, N. Xiao, Y. Zhou, and Q. Zhang. Policy-driven autonomic configuration management for nosql. In *Proceedings of the 2015 IEEE International Conference on Cloud Computing (CLOUD)*, pages 245–252, New York, NY, 2015.

[26] Y. Belinkov and Y. Bisk. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, April 30–May 3 2018.

[27] M. Berrendorf, E. Faerman, and V. Tresp. Active learning for entity alignment. In *Advances in Information Retrieval - 43rd European Conference on IR Research (ECIR)*, volume Part I, pages 48–62, Virtual Event, March 28–April 1 2021.

[28] M. Berrendorf, L. Wacker, and E. Faerman. A critical assessment of state-of-the-art in entity alignment. In *Advances in Information Retrieval - 43rd European Conference on IR Research (ECIR)*, volume Part II, pages 18–32, Virtual Event, March 28–April 1 2021.

[29] K. Bhardwaj, N. P. Pandey, S. Priyadarshi, V. Ganapathy, S. Kadambi, R. Esteves, S. Borse, P. Whatmough, R. Garrepalli, M. Van Baalen, H. Teague, and M. Nagel. Sparse high rank adapters. In *Advances in Neural Information Processing Systems*, volume 37, pages 13685–13715. Curran Associates, Inc., 2024.

[30] M. Bladt and B. F. Nielsen. *Matrix-Exponential Distributions in Applied Probability*. Probability Theory and Stochastic Modelling. Springer, New York, NY, 2017.

[31] M. Bladt and J. Yslas. Heavy-tailed phase-type distributions: a unified approach. *Extremes*, 25:529–565, 2022.

[32] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis. Align your latents: High-resolution video synthesis with latent diffusion models, 2023.

[33] M. Blohm, G. Jagfeld, E. Sood, X. Yu, and N. T. Vu. Comparing attention-based convolutional and recurrent neural networks: Success and limitations in machine reading comprehension. In *22nd Conference on Computational Natural Language Learning (CoNLL)*, pages 108–118, Brussels, Belgium, October 31–November 1 2018.

[34] A. Blum, T. Dick, N. Manoj, and H. Zhang. Random smoothing might be unable to certify l∞ robustness for high-dimensional images. *J. Mach. Learn. Res.*, 21(1), jan 2020.

[35] A. Bojchevski and S. Günnemann. Adversarial attacks on node embeddings via graph poisoning. In *36th International Conference on Machine Learning (ICML)*, pages 695–704, Long Beach, California, USA, June 9–15 2019.

[36] A. Bojchevski and S. Günnemann. Certifiable robustness to graph perturbations. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 8317–8328, 2019.

[37] A. Bojchevski, J. Klicpera, and S. Günnemann. Efficient robustness certificates for discrete data: Sparsity-aware randomized smoothing for graphs, images and more. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[38] R. BONIC and J. FRAMPTON. Smooth functions on banach manifolds. *Journal of Mathematics and Mechanics*, 15(5):877–898, 1966.

[39] A. Breuer, R. Eilat, and U. Weinsberg. Friend or faux: Graph-based early detection of fake accounts on social networks. In *Proceedings of the Web Conference (WWW)*, 2020.

[40] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022.

[41] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

[42] T. Cao, J. Kong, X. Zhao, W. Yao, J. Ding, J. Zhu, and J. Zhang. Product2img: Prompt-free e-commerce product background generation with diffusion model and self-improved LMM. In J. Cai, M. S. Kankanhalli, B. Prabhakaran, S. Boll, R. Subramanian, L. Zheng, V. K. Singh, P. César, L. Xie, and D. Xu, editors, *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pages 10774–10783. ACM, 2024.

[43] Y. Cao, Z. Liu, C. Li, Z. Liu, J. Li, and T.-S. Chua. Multi-channel graph neural network for entity alignment. In *57th Conference of the Association for Computational Linguistics (ACL)*, pages 1452–1461, Florence, Italy, July 28–August 2 2019.

[44] S. Chakravarthy and A. S. Alfa. *Matrix-Analytic Methods in Stochastic Models*. CRC Press, 1996.

[45] A. Chan, Y. Tay, Y.-S. Ong, and A. Zhang. Poison attacks against text datasets with conditional adversarially regularized autoencoder. In *2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4175–4189, Online Event, November 16–20 2020.

[46] H. Chang, Y. Rong, T. Xu, W. Huang, H. Zhang, P. Cui, W. Zhu, and J. Huang. A restricted black-box adversarial framework towards attacking graph embedding models.

In *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, New York, NY, USA, February 7–12 2020.

[47] H. Chang, Y. Rong, T. Xu, W. Huang, H. Zhang, P. Cui, W. Zhu, and J. Huang. A restricted black-box adversarial framework towards attacking graph embedding models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[48] H. Chefer, Y. Alaluf, Y. Vinker, L. Wolf, and D. Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models, 2023.

[49] J. Chen, L. Chen, Y. Chen, M. Zhao, S. Yu, Q. Xuan, and X. Yang. Ga-based q-attack on community detection. *IEEE Transactions on Computational Social Systems*, 6(3):491–503, 2020.

[50] J. Chen, T. Ge, G. Jiang, Z. Zhang, D. Lian, and K. Zheng. Efficient optimal selection for composited advertising creatives with tree structure. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 3967–3975. AAAI Press, 2021.

[51] J. Chen, Y. Wu, X. Xu, Y. Chen, H. Zheng, and Q. Xuan. Fast gradient attack on network embedding. *CoRR*, abs/1809.02797, 2018.

[52] J. Chen, J. Xu, G. Jiang, T. Ge, Z. Zhang, D. Lian, and K. Zheng. Automated creative optimization for e-commerce advertising. In J. Leskovec, M. Grobelnik, M. Najork, J. Tang, and L. Zia, editors, *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 2304–2313. ACM / IW3C2, 2021.

[53] L. Chen, J. Li, J. Peng, T. Xie, Z. Cao, K. Xu, X. He, and Z. Zheng. A survey of adversarial learning on graphs. *CoRR*, abs/2003.05730, 2020.

[54] M. Chen, W. Shi, B. Zhou, and D. Roth. Cross-lingual entity alignment with incidental supervision. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (EACL 2021)*, pages 645–658, Online, April 2021.

[55] M. Chen, Y. Tian, M. Yang, and C. Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*, pages 1511–1517, Melbourne, Australia, August 2017.

[56] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations (ICLR)*, 2021.

[57] R. Chen, Y. Chen, N. Jiao, and K. Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22246–22256, October 2023.

[58] T. Chen, R. Zhang, and G. E. Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations (ICLR)*, 2023.

[59] Y. Chen, Y. Nadji, A. Kountouras, F. Monrose, R. Perdisci, M. Antonakakis, and N. Vasiloglou. Practical attacks against graph-based clustering. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS 2017)*, pages 1125–1142, Dallas, TX, USA, October-November 2017.

[60] Z. Chen, X. Yu, B. Song, J. Gao, X. Hu, and W.-S. Yang. Community-based network alignment for large attributed network. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pages 587–596, 2017.

[61] H. Cheng, D. Lo, Y. Zhou, X. Wang, and X. Yan. Identifying bug signatures using discriminative graph mining. In *Proceedings of the International Symposium on Software Testing and Analysis (ISSTA)*, 2009.

[62] H. Cheng, Y. Zhou, X. Huang, and J. X. Yu. Clustering large attributed information networks: An efficient incremental computing approach. *Data Mining and Knowledge Discovery (DMKD)*, 25(3):450–477, 2012.

[63] H. Cheng, Y. Zhou, and J. X. Yu. Clustering large attributed graphs: A balance between structural and attribute similarities. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):1–33, 2011.

[64] S.-Y. Chou, P.-Y. Chen, and T.-Y. Ho. Villandiffusion: A unified backdoor attack framework for diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[65] X. Chu, X. Fan, D. Yao, Z. Zhu, J. Huang, and J. Bi. Cross-network embedding for multi-network alignment. In *Proceedings of the World Wide Web Conference (WWW)*, 2019.

[66] J. Cohen, E. Rosenfeld, and Z. Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 09–15 Jun 2019.

[67] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song. Adversarial attack on graph structured data. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1123–1132, 2018.

[68] Q. Dai, X. Shen, L. Zhang, Q. Li, and D. Wang. Adversarial training methods for network embedding. In *Proceedings of the World Wide Web Conference (WWW)*, pages 329–339, 2019.

[69] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019)*, pages 4171–4186, 2019.

[70] C. Dewey. You probably haven't even noticed google's sketchy quest to control the world's knowledge, 2016.

[71] P. Dey and S. Medya. Manipulating node similarity measures in networks. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '20)*, Auckland, New Zealand, May 9-13 2020.

[72] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21. Curran Associates Inc., 2021.

[73] P. Dirac. *The Principles of Quantum Mechanics*. Oxford University Press, 1930.

[74] K. D. Doan, Y. Lao, and P. Li. Marksman backdoor: Backdoor attacks with arbitrary target class. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 38260–38273, 2022.

[75] K. D. Doan, Y. Lao, W. Zhao, and P. Li. LIRA: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11946–11956, 2021.

[76] Y. Dou, G. Ma, P. S. Yu, and S. Xie. Robust spammer detection by nash reinforcement learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 924–933, 2020.

[77] X. Du, J. Yan, and H. Zha. Joint link prediction and network alignment via cross-graph embedding. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2251–2257, 2019.

[78] Z. Du, W. Feng, H. Wang, Y. Li, J. Wang, J. Li, Z. Zhang, J. Lv, X. Zhu, J. Jin, J. Shen, Z. Lin, and J. Shao. Towards reliable advertising image generation using human feedback. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XX*, volume 15078 of *Lecture Notes in Computer Science*, pages 399–415. Springer, 2024.

[79] P. Elinas, E. V. Bonilla, and L. Tiao. Variational inference for graph convolutional networks in the absence of graph data and adversarial settings. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[80] N. Entezari, S. Al-Sayouri, A. Darvishzadeh, and E. Papalexakis. All you need is low (rank): Defending against adversarial attacks on graphs. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, 2020.

[81] W. Feller. *An Introduction to Probability Theory and Its Applications, Volume 2*. Wiley, New York, 2nd edition, 1991.

[82] F. Feng, X. He, J. Tang, and T.-S. Chua. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2020.

[83] J. Feng, M. Zhang, H. Wang, Z. Yang, C. Zhang, Y. Li, and D. Jin. Dplink: User identity linkage via deep neural network from heterogeneous mobility data. In *WWW*, pages 459–469, 2019.

[84] W. Feng, J. Zhang, Y. Dong, Y. Han, H. Luan, Q. Xu, Q. Yang, E. Kharlamov, and J. Tang. Graph random neural networks for semi-supervised learning on graphs. In *NeurIPS*, 2020.

[85] Z. Feng, Z. Zhang, X. Yu, Y. Fang, L. Li, X. Chen, Y. Lu, J. Liu, W. Yin, S. Feng, Y. Sun, L. Chen, H. Tian, H. Wu, and H. Wang. ERNIE-ViLG 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[86] P. Fernandez, G. Couairon, H. Jégou, M. Douze, and T. Furon. The stable signature: Rooting watermarks in latent diffusion models. *ICCV*, 2023.

[87] M. Fey, J. E. Lenssen, C. Morris, J. Masci, and N. M. Kriege. Deep graph matching consensus. In *ICLR*, 2020.

[88] K. O. Friedrichs. The identity of weak and strong extensions of differential operators. *Transactions of the American Mathematical Society*, 55:132–151, 1944.

[89] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023.

[90] H. Gao, H. Zhang, Y. Dong, and Z. Deng. Evaluating the robustness of text-to-image diffusion models against real-world attacks, 2023.

[91] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, oct 2020.

[92] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27 (NeurIPS 2014)*, pages 2672–2680, 2014.

[93] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[94] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations (ICLR 2015), Conference Track Proceedings*, San Diego, CA, USA, May 7-9 2015.

[95] S. Goswami, A. Pokhrel, K. Lee, L. Liu, Q. Zhang, and Y. Zhou. Graphmap: Scalable iterative graph processing using nosql. *The Journal of Supercomputing (TSC)*, 76(9):6619–6647, 2020.

[96] L. Guo, Z. Sun, and W. Hu. Learning to exploit long-term relational dependencies in knowledge graphs. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, pages 2505–2514, Long Beach, California, USA, June 9-15 2019.

[97] P. Guo, Y. Zhou, J. Zhuang, T. Chen, and Y.-R. Kang. Efficient algorithm for mining both closed and maximal frequent free subtrees using canonical forms. In *ADMA*, pages 96–107, Wuhan, China, 2005.

[98] S. Guo, Z. Jin, F. Sun, J. Li, Z. Li, Y. Shi, and N. Cao. Vinci: An intelligent graphic design system for generating advertising posters. In Y. Kitamura, A. Quigley, K. Isbister, T. Igarashi, P. Bjørn, and S. M. Drucker, editors, *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, pages 577:1–577:17. ACM, 2021.

[99] F. He, M. Du, A. Filos-Ratsikas, L. Cheng, Q. Song, M. Lin, and J. Vines. AI driven online advertising: Market design, generative ai, and ethics. In T. Chua, C. Ngo,

R. K. Lee, R. Kumar, and H. W. Lauw, editors, *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*, pages 1407–1409. ACM, 2024.

[100] X. He, J. Jia, M. Backes, N. Z. Gong, and Y. Zhang. Stealing links from graph neural networks. In *30th USENIX Security Symposium (USENIX Security 2021)*, August 11-13, 2021.

[101] M. Heimann, H. Shen, T. Safavi, and D. Koutra. Regal: Representation learning-based graph alignment. In *CIKM*, pages 117–126, 2018.

[102] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or. Prompt-to-prompt image editing with cross attention control, 2022.

[103] N. J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.

[104] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet, and T. Salimans. Imagen video: High definition video generation with diffusion models, 2022.

[105] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, 2020.

[106] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pages 8633–8646. Curran Associates, Inc., 2022.

[107] J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, and G. Weikum. Yago2: Exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th International Conference on*

World Wide Web (WWW 2011), Companion Volume, pages 229–232, Hyderabad, India, March 28-April 1 2011.

[108] S. Hou, Y. Fan, Y. Zhang, Y. Ye, J. Lei, W. Wan, J. Wang, and H. Shao. Cyber: Enhancing robustness of android malware detection system against adversarial attacks on heterogeneous graph based model. In *CIKM*, pages 609–618, 2019.

[109] S. Hou, Y. Fan, Y. Zhang, Y. Ye, J. Lei, W. Wan, J. Wang, Q. Xiong, and F. Shao. acyber: Enhancing robustness of android malware detection system against adversarial attacks on heterogeneous graph-based model. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM 2019)*, pages 609–618, Beijing, China, November 3-7 2019.

[110] W. R. Huang, C. Peyser, T. N. Sainath, R. Pang, T. D. Strohman, and S. Kumar. Sentence-select: Large-scale language model data selection for rare-word speech recognition. In *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022, Incheon, Korea, September 18-22, 2022*, pages 689–693. ISCA, 2022.

[111] Y. Huang, F. Juefei-Xu, Q. Guo, J. Zhang, Y. Wu, M. Hu, T. Li, G. Pu, and Y. Liu. Personalization as a shortcut for few-shot backdoor attack against text-to-image diffusion models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21169–21178, Mar. 2024.

[112] Hugging Face. Hugging face hub. `https://huggingface.co`, 2024. Accessed: January 22, 2025.

[113] A. Huq and M. T. Pervin. Adversarial attacks and defense on texts: A survey. *CoRR*, abs/2005.14108, 2020.

[114] T. T. Huynh, V. V. Tong, T. T. Nguyen, H. Yin, M. Weidlich, and N. Q. V. Hung. Adaptive network alignment with unsupervised and multi-order convolutional networks. In *ICDE*, pages 85–96, 2020.

[115] H. Jalalzai, P. Colombo, C. Clavel, E. Gaussier, G. Varni, E. Vignon, and A. Sabourin. Heavy-tailed representations, text polarity classification & data augmentation. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4295–4307. Curran Associates, Inc., 2020.

[116] J. Jia, B. Wang, X. Cao, and N. Z. Gong. Certified robustness of community detection against adversarial structural perturbation via randomized smoothing. In *WWW*, pages 2718–2724, 2020.

[117] R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 2021–2031, Copenhagen, Denmark, September 9-11 2017.

[118] C. Jiang, Z. Chen, B. Zhang, Y. Ren, X. Dong, L. Cheng, X. Yang, L. Li, J. Zhou, and L. Mo. GATS: generative audience targeting system for online advertising. In G. H. Yang, H. Wang, S. Han, C. Hauff, G. Zuccon, and Y. Zhang, editors, *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 2920–2924. ACM, 2024.

[119] H. Jin, Z. Shi, V. J. S. A. Peruri, and X. Zhang. Certified robustness of graph convolution networks for graph classification under topological attacks. In *NeurIPS*, 2020.

[120] H. Jin and X. Zhang. Latent adversarial training of graph convolution networks. In *ICML*, 2019.

[121] R. Jin, D. Li, J. Gao, Z. Liu, L. Chen, and Y. Zhou. Towards a better understanding of linear models for recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21)*, Virtual Event, 2021.

[122] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[123] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019*, pages 4401–4410, 2019.

[124] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani. Imagic: Text-based real image editing with diffusion models. In *Conference on Computer Vision and Pattern Recognition 2023*, 2023.

[125] L. Khachatryan, A. Movsisyan, V. Tadevosyan, R. Henschel, Z. Wang, S. Navasardyan, and H. Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*, 2023.

[126] G. Kim, T. Kwon, and J. C. Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2426–2435, June 2022.

[127] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems 29 (NeurIPS 2016)*, pages 4743–4751, 2016.

[128] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[129] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations (ICLR 2014), Conference Track Proceedings*, 2014.

[130] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

[131] X. Kou, T. Zhao, F. Zhang, S. Li, and Q. Zhang. Self-supervised augmentation and generation for multi-lingual text advertisements at bing. In A. Zhang and H. Rangwala, editors, *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 3187–3196. ACM, 2022.

[132] N. Kshetri. Generative AI in advertising. *IT Prof.*, 26(5):15–19, 2024.

[133] A. Kumar, A. Levine, T. Goldstein, and S. Feizi. Curse of dimensionality on randomized smoothing for certifiable robustness. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5458–5467. PMLR, 13–18 Jul 2020.

[134] N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J.-Y. Zhu. Multi-concept customization of text-to-image diffusion, 2023.

[135] K. Kurita, P. Michel, and G. Neubig. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 2793–2806, Online, July 5-10 2020.

[136] S. Lang. *Differential and Riemannian Manifolds*. Graduate Texts in Mathematics. Springer New York, 2012.

[137] K. Lee, L. Liu, K. Schwan, C. Pu, Q. Zhang, Y. Zhou, E. Yigitoglu, and P. Yuan. Scaling iterative graph computations with graphmap. In *Proceedings of the 27th IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC '15)*, pages 57:1–57:12, Austin, TX, 2015.

[138] K. Lee, L. Liu, Y. Tang, Q. Zhang, and Y. Zhou. Efficient and customizable data partitioning framework for distributed big data processing in the cloud. In *Proceedings of the 2013 IEEE International Conference on Cloud Computing (CLOUD '13)*, pages 327–334, Santa Clara, CA, 2013.

[139] K. J. Lee, B. Jeong, S. Kim, D. Kim, and D. Park. General commerce intelligence: Glocally federated nlp-based engine for privacy-preserving and sustainable personalized services of multi-merchants. In M. J. Wooldridge, J. G. Dy, and S. Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 22752–22760. AAAI Press, 2024.

[140] Z. Lei, C. Zhang, X. Xu, W. Wu, Z. Niu, H. Wu, H. Wang, Y. Yang, and S. Li. Plato-ad: A unified advertisement text generation framework with multi-task prompt learning. In Y. Li and A. Lazaridou, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: EMNLP 2022 - Industry Track, Abu Dhabi, UAE, December 7 - 11, 2022*, pages 512–520. Association for Computational Linguistics, 2022.

[141] C. Li, Y. Cao, L. Hou, J. Shi, J. Li, and T.-S. Chua. Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 2723–2732, Hong Kong, China, November 3-7 2019.

[142] C. Li, S. Wang, H. Wang, Y. Liang, P. S. Yu, Z. Li, and W. Wang. Partially shared adversarial learning for semi-supervised multi-platform user identity linkage. In *CIKM*, pages 249–258, 2019.

[143] C. Li, S. Wang, Y. Wang, P. S. Yu, Y. Liang, and Z. Li. Adversarial learning for weakly-supervised social network alignment. In *AAAI*, pages 996–1003, 2019.

[144] D. Li, J. Li, and S. Hoi. BLIP-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[145] H. Li, X. Zhang, X. Liu, Y. Gong, Y. Wang, Q. Chen, and P. Cheng. Enhancing large language model performance with gradient-based parameter selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24431–24439, Apr. 2025.

[146] J. Li, H. Zhang, Z. Han, Y. Rong, H. Cheng, and J. Huang. Adversarial attack on community detection by hiding individuals. In *WWW*, pages 917–927, 2020.

[147] J. Li, H. Zhang, Z. Han, Y. Rong, H. Cheng, and J. Huang. Adversarial attack on community detection by hiding individuals. In *Proceedings of The Web Conference 2020 (WWW '20)*, pages 917–927, Taipei, Taiwan, April 20-24 2020.

[148] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu. Bert-attack: Adversarial attack against bert using bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 6193–6202, Online, November 16-20 2020.

[149] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 4328–4343, 2022.

[150] Y. Li, B. Hu, W. Luo, L. Ma, Y. Ding, and M. Zhang. A multimodal in-context tuning approach for e-commerce product description generation. In N. Calzolari, M. Kan,

V. Hoste, A. Lenci, S. Sakti, and N. Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 850–861. ELRA and ICCL, 2024.

[151] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16443–16452, 2021.

[152] Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, and Y. J. Lee. Gligen: Open-set grounded text-to-image generation. *CVPR*, 2023.

[153] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 300–309, June 2023.

[154] J. Lin, L. Xu, Y. Liu, and X. Zhang. Composite backdoor attack for deep neural network by mixing existing benign features. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 113–131, 2020.

[155] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, 2014.

[156] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley. AudioLDM: Text-to-audio generation with latent diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 21450–21474. PMLR, 23–29 Jul 2023.

[157] H. Liu, Y. Wu, S. Zhai, B. Yuan, and N. Zhang. Riatig: Reliable and imperceptible adversarial text-to-image generation with natural prompts. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 20585–20594, June 2023.

[158] J. Liu, J. Huang, Y. Zhou, X. Li, S. Ji, H. Xiong, and D. Dou. From distributed machine learning to federated learning: A survey. *CoRR*, abs/2104.14362, 2021.

[159] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[160] Y. Liu, H. Ding, D. Chen, and J. Xu. Novel geometric approach for global alignment of ppi networks. In *AAAI*, pages 31–37, 2017.

[161] Y. Liu, X. Ma, J. Bailey, and F. Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision (ECCV)*, pages 182–199, 2020.

[162] Z. Liu, Y. Cao, L. Pan, J. Li, and T.-S. Chua. Exploring and evaluating attributes, values, and structures for entity alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 6355–6364, Online, November 16-20 2020.

[163] A. Logins, Y. Li, and P. Karras. On the robustness of cascade diffusion under node attacks. In *WWW*, pages 2711–2717, 2020.

[164] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[165] J. Ma, S. Ding, and Q. Mei. Towards more practical adversarial attacks on graph neural networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, Online, 2020.

[166] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations (ICLR 2018)*, Vancouver, BC, Canada, April 30-May 3 2018.

[167] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[168] X. Mao, W. Wang, Y. Wu, and M. Lan. Boosting the speed of entity alignment 10x: Dual attention matching network with normalized hard sample mining. In *WWW '21: The Web Conference 2021*, April 19-23 2021.

[169] X. Mao, W. Wang, H. Xu, M. Lan, and Y. Wu. Mraea: An efficient and robust entity alignment approach for cross-lingual knowledge graph. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining*, pages 420–428, Houston, TX, USA, February 3-7 2020.

[170] X. Mao, W. Wang, H. Xu, Y. Wu, and M. Lan. Relational reflection entity alignment. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management*, pages 1095–1104, Virtual Event, Ireland, October 19-23 2020.

[171] G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley, 2000.

[172] E. Meguellati, L. Han, A. Bernstein, S. W. Sadiq, and G. Demartini. How good are llms in generating personalized advertisements? In T. Chua, C. Ngo, R. K. Lee, R. Kumar, and H. W. Lauw, editors, *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*, pages 826–829. ACM, 2024.

[173] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *NeurIPS*, 26:3111–3119, 2013.

[174] G. A. Miller. Wordnet: A lexical database for english. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, USA*, pages 23–26, February 1992.

[175] P. Minervini, T. Demeester, T. Rocktäschel, and S. Riedel. Adversarial sets for regularising neural link predictors. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence (UAI 2017)*, Sydney, Australia, August 11-15 2017.

[176] M. Mita, S. Murakami, A. Kato, and P. Zhang. Striking gold in advertising: Standardization and exploration of ad text generation. In L. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 955–972. Association for Computational Linguistics, 2024.

[177] J. Mohapatra, C.-Y. Ko, T.-W. Weng, P.-Y. Chen, S. Liu, and L. Daniel. Higher-order certification for randomized smoothing. In *Advances in Neural Information Processing Systems*, volume 33, pages 4501–4511. Curran Associates, Inc., 2020.

[178] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022.

[179] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2020 - Demos)*, pages 119–126, Online, November 16-20 2020.

[180] H. Nasir, N. Veldt, S. Mohammadi, A. Grama, and D. F. Gleich. Robust spectral network alignment. In *WWW*, pages 619–628, 2020.

[181] E. G. Ng, B. Pang, P. Sharma, and R. Soricut. Understanding guided image captioning performance across domains. *arXiv preprint arXiv:2012.02339*, 2020.

[182] T. A. Nguyen and A. Tran. Input-aware dynamic backdoor attack. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 3454–3464, 2020.

[183] T. A. Nguyen and A. T. Tran. WaNet: Imperceptible warping-based backdoor attack. In *9th International Conference on Learning Representations (ICLR)*, 2021.

[184] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8162–8171, 2021.

[185] A. Q. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR, 2022.

[186] H. Nie, X. Han, L. Sun, C. M. Wong, Q. Chen, S. Wu, and W. Zhang. Global structure and local semantics-preserved embeddings for entity alignment. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020)*, pages 3658–3664, 2020.

[187] T. Niu and M. Bansal. Adversarial oversensitivity and over-stability strategies for dialogue models. In *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018)*, pages 486–496, Brussels, Belgium, October 31-November 1 2018.

[188] X. Niu, P. Mathur, G. Dinu, and Y. Al-Onaizan. Evaluating robustness to input perturbations for neural machine translation. In *Proceedings of the 58th Annual Meeting of*

the Association for Computational Linguistics (ACL 2020), pages 8538–8544, Online, July 5-10 2020.

[189] C. A. O'Cinneide. Characterization of phase-type distributions. *Communications in Statistics: Stochastic Models*, 6(1):1–57, 1990.

[190] OpenAI, J. Achiam, and S. A. et al. Gpt-4 technical report, 2024.

[191] B. Palanisamy, L. Liu, K. Lee, S. Meng, Y. Tang, and Y. Zhou. Anonymizing continuous queries with delay-tolerant mix-zones over road networks. *Distributed and Parallel Databases (DAPD)*, 32(1):91–118, 2014.

[192] B. Palanisamy, L. Liu, K. Lee, S. Meng, Y. Tang, and Y. Zhou. Answering continuous queries with delay-tolerant mix-zones over road networks. In *Distributed and Parallel Databases (DAPD)*, volume 32, pages 91–118, 2014.

[193] B. Palanisamy, L. Liu, Y. Zhou, and Q. Wang. Privacy-preserving publishing of multilevel utility-controlled graph datasets. *ACM Transactions on Internet Technology (TOIT)*, 18(2):24:1–24:21, 2018.

[194] X. Pan, M. Zhang, B. Sheng, J. Zhu, and M. Yang. Hidden trigger backdoor attack on NLP models via linguistic style manipulation. In *Proceedings of the 31st USENIX Security Symposium (USENIX Security)*, pages 3611–3628, 2022.

[195] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[196] S. Pei, L. Yu, R. Hoehndorf, and X. Zhang. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *The World Wide Web Conference (WWW 2019)*, pages 3130–3136, San Francisco, CA, USA, May 13-17 2019.

[197] S. Pei, L. Yu, G. Yu, and X. Zhang. Rea: Robust cross-lingual entity alignment between knowledge graphs. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2175–2184, Virtual Event, CA, USA, August 23-27 2020.

[198] S. Pei, L. Yu, and X. Zhang. Improving cross-lingual entity alignment via optimal transport. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)*, pages 3231–3237, Macao, China, August 10-16 2019.

[199] B. Peng, X. Ling, Z. Chen, H. Sun, and X. Ning. ecellm: Generalizing large language models for e-commerce from large-scale, high-quality instruction data. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[200] J. Pennington, S. S. Schoenholz, and S. Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: Theory and practice. In *NeurIPS*, pages 4785–4795, 2017.

[201] J. Pennington, S. S. Schoenholz, and S. Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 4785–4795, Long Beach, CA, USA, December 4-9 2017.

[202] P. Pezeshkpour, Y. Tian, and S. Singh. Investigating robustness and interpretability of link prediction via adversarial modifications. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 3336–3347, Minneapolis, MN, USA, June 2-7 2019.

[203] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

[204] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023.

[205] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8599–8608, 2021.

[206] J. Pujara, E. Augustine, and L. Getoor. Sparsity and noise: Where knowledge graph embeddings fall short. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 1751–1756, Copenhagen, Denmark, September 9-11 2017.

[207] F. Qi, M. Li, Y. Chen, Z. Zhang, Z. Liu, Y. Wang, and M. Sun. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 443–453, 2021.

[208] F. Qi, Y. Yao, S. Xu, Z. Liu, and M. Sun. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4873–4883, 2021.

[209] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989.

[210] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations (ICLR) 2016, Conference Track Proceedings*, 2016.

[211] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020.

[212] M. Raman, S. Agarwal, P. Wang, A. Chan, H. Wang, S. Kim, R. Rossi, H. Zhao, N. Lipka, and X. Ren. Learning to deceive knowledge graph augmented models via targeted perturbation. In *9th International Conference on Learning Representations (ICLR 2021)*, Online, May 4-7 2021.

[213] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[214] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR, 18–24 Jul 2021.

[215] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR, 18–24 Jul 2021.

[216] J. Ren, Z. Zhang, J. Jin, X. Zhao, S. Wu, Y. Zhou, Y. Shen, T. Che, R. Jin, and D. Dou. Integrated defense for resilient graph matching. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, pages 8982–8997, Virtual Event, 2021.

[217] J. Ren, Y. Zhou, R. Jin, Z. Zhang, D. Dou, and P. Wang. Dual adversarial learning based network alignment. *ICDM*, pages 1288–1293, 2019.

[218] J. Ren, Y. Zhou, R. Jin, Z. Zhang, D. Dou, and P. Wang. Dual adversarial learning based network alignment. In *Proceedings of the 19th IEEE International Conference on Data Mining (ICDM'19)*, pages 1288–1293, Beijing, China, 2019.

[219] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, volume 32 of *PMLR*, pages 1278–1286, 2014.

[220] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[221] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022*, pages 10674–10685, 2022.

[222] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[223] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 1995.

[224] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pages 36479–36494. Curran Associates, Inc., 2022.

[225] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang. Dynamic backdoor attacks against machine learning models. In *Proceedings of the 7th IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 703–718, 2022.

[226] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *5th International Conference on Learning Representations (ICLR) 2017, Conference Track Proceedings*, 2017.

[227] C. Schuhmann, R. Beaumont, R. Vencu, C. W. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. R. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[228] P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018.

[229] X. Shen, C. Du, T. Pang, M. Lin, Y. Wong, and M. Kankanhalli. Finetuning text-to-image diffusion models for fairness. In *The Twelfth International Conference on Learning Representations*, 2024.

[230] Y. Shi, C. Xue, J. Pan, W. Zhang, V. Y. Tan, and S. Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. *arXiv preprint arXiv:2306.14435*, 2023.

[231] I. Shumailov, Z. Shumaylov, D. Kazhdan, Y. Zhao, N. Papernot, M. A. Erdogdu, and R. Anderson. Manipulating SGD with data ordering attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 18021–18032, 2021.

[232] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, D. Parikh, S. Gupta, and Y. Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2023.

[233] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

[234] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

[235] W. Song, S. Wang, B. Yang, Y. Lu, X. Zhao, and X. Liu. Learning node and edge embeddings for signed networks. *Neurocomputing*, 319:42–54, 2018.

[236] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[237] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.

[238] Y. Song and S. Ermon. Improved techniques for training score-based generative models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[239] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

[240] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[241] H. Souri, M. Goldblum, L. Fowl, R. Chellappa, and T. Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 19165–19178, 2022.

[242] R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451, San Francisco, California, USA, February 4-9 2017.

[243] StabilityAI. Deepfloyd-if-i-m-v1.0, 2023. Accessed: 2024-09-20.

[244] L. Struppek, D. Hintersdorf, and K. Kersting. Rickrolling the artist: Injecting backdoors into text encoders for text-to-image synthesis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4561–4573, Los Alamitos, CA, USA, oct 2023. IEEE Computer Society.

[245] Z. Su, L. Liu, M. Li, X. Fan, and Y. Zhou. Servicetrust: Trust management in service provision networks. In *Proceedings of the 10th IEEE International Conference on Services Computing (SCC'13)*, pages 272–279, Santa Clara, CA, 2013.

[246] Z. Su, L. Liu, M. Li, X. Fan, and Y. Zhou. Reliable and resilient trust management in distributed service provision networks. *ACM Transactions on the Web (TWEB)*, 9(3):1–37, 2015.

[247] J. Sun, Y. Zhou, and C. Zong. Dual attention network for cross-lingual entity alignment. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020)*, pages 3190–3201, Barcelona, Spain (Online), December 8-13 2020.

[248] Y. Sun, S. Wang, X. Tang, T.-Y. Hsieh, and V. G. Honavar. Adversarial attacks on graph neural networks via node injections: A hierarchical reinforcement learning approach. In *WWW '20: The Web Conference 2020*, pages 673–683, Taipei, Taiwan, April 20-24 2020.

[249] Z. Sun, M. Chen, W. Hu, C. Wang, J. Dai, and W. Zhang. Knowledge association with hyperbolic knowledge graph embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 5704–5716, Online, November 16-20 2020.

[250] Z. Sun, W. Hu, and C. Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *The Semantic Web - ISWC 2017, 16th International Semantic Web Conference*, pages 628–644, Vienna, Austria, October 21-25 2017.

[251] Z. Sun, W. Hu, Q. Zhang, and Y. Qu. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI 2018)*, pages 4396–4402, Stockholm, Sweden, July 13-19 2018.

[252] Z. Sun, J. Huang, W. Hu, M. Chen, L. Guo, and Y. Qu. Transedge: Translating relation-contextualized embeddings for knowledge graphs. In *The Semantic Web - ISWC 2019, 18th International Semantic Web Conference*, pages 612–629, Auckland, New Zealand, October 26-30 2019.

[253] Z. Sun, C. Wang, W. Hu, M. Chen, J. Dai, W. Zhang, and Y. Qu. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020)*, pages 222–229, New York, NY, USA, February 7-12 2020.

[254] T. Takahashi. Indirect adversarial attacks via poisoning neighbors for graph convolutional networks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1395–1400, Los Angeles, CA, USA, December 9-12 2019.

[255] S. Tan, S. R. Joty, M.-Y. Kan, and R. Socher. It's morphin' time! combating linguistic discrimination with inflectional perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 2920–2932, Online, July 5-10 2020.

[256] X. Tang, J. Zhang, B. Chen, Y. Yang, H. Chen, and C. Li. Bert-int: A bert-based interaction model for knowledge graph alignment. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020)*, pages 3174–3180, 2020.

[257] S. Tulyakov, M. Liu, X. Yang, and J. Kautz. MoCoGAN: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018*, pages 1526–1535, 2018.

[258] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *Proceedings of the 9th ISCA Speech Synthesis Workshop (SSW) 2016*, page 125, 2016.

[259] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1747–1756, 2016.

[260] S. Vashishtha, A. Prakash, L. Morishetti, K. Nag, Y. Arora, S. Kumar, and K. Achan. Chaining text-to-image and large language model: A novel approach for generating

personalized e-commerce banners. In R. Baeza-Yates and F. Bonchi, editors, *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 5825–5835. ACM, 2024.

[261] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, pages 5998–6008, 2017.

[262] J. Vice, N. Akhtar, R. Hartley, and A. Mian. Bagm: A backdoor attack for manipulating text-to-image generative models. *IEEE Transactions on Information Forensics and Security*, pages 1–1, 2024.

[263] V. Voleti, A. Jolicoeur-Martineau, and C. Pal. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[264] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 2153–2162, Hong Kong, China, November 3-7 2019.

[265] B. Wang and N. Z. Gong. Attacking graph-based classification via manipulating the graph structure. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS 2019)*, pages 2023–2040, London, UK, November 11-15 2019.

[266] B. Wang, H. Pei, B. Pan, Q. Chen, S. Wang, and B. Li. T3: Tree-autoencoder constrained adversarial text generation for targeted attack. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 6134–6150, Online, November 16-20 2020.

[267] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. S. Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

[268] T. Wang, Y. Yao, F. Xu, S. An, H. Tong, and T. Wang. An invisible black-box backdoor attack through frequency domain. In *European Conference on Computer Vision (ECCV)*, pages 396–413, 2022.

[269] W. Wang, L. Wang, R. Wang, Z. Wang, and A. Ye. Towards a robust deep neural network in texts: A survey. *CoRR*, abs/1902.07285, 2019.

[270] X. Wang, X. Gu, J. Cao, Z. Zhao, Y. Yan, B. Middha, and X. Xie. Reinforcing pretrained models for generating attractive text advertisements. In F. Zhu, B. C. Ooi, and C. Miao, editors, *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 3697–3707. ACM, 2021.

[271] Z. Wang, Q. Lv, X. Lan, and Y. Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 349–357, Brussels, Belgium, October 31-November 4 2018.

[272] M. Waniek, T. P. Michalak, M. J. Wooldridge, and T. Rahwan. Hiding individuals and communities in a social network. *Nature Human Behaviour*, 2:139–147, 2018.

[273] P. Wei, S. Liu, X. Yang, L. Wang, and B. Zheng. Towards personalized bundle creative generation with contrastive non-autoregressive decoding. In E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, and G. Kazai, editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 2634–2638. ACM, 2022.

[274] Y. Wen, J. Kirchenbauer, J. Geiping, and T. Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[275] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao. Backdoor attacks against deep learning systems in the physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6202–6211, 2021.

[276] T. Wolf and et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics.

[277] H. Wu, C. Wang, Y. Tyshetskiy, A. Docherty, K. Lu, and L. Zhu. Adversarial examples for graph data: Deep insights into attack and defense. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)*, pages 4816–4823, 2019.

[278] J. Z. Wu, Y. Ge, X. Wang, S. W. Lei, Y. Gu, Y. Shi, W. Hsu, Y. Shan, X. Qie, and M. Z. Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023.

[279] S. Wu, Y. Li, D. Zhang, Y. Zhou, and Z. Wu. Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*, pages 5811–5820, Online, 2020.

[280] S. Wu, Y. Li, D. Zhang, Y. Zhou, and Z. Wu. Topicka: Generating commonsense knowledge-aware dialogue responses towards the recommended topic fact. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI'20)*, pages 3766–3772, Online, 2021.

[281] S. Wu, M. Wang, D. Zhang, Y. Zhou, Y. Li, and Z. Wu. Knowledge-aware dialogue generation via hierarchical infobox accessing and infobox-dialogue interaction graph network. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21)*, Online, 2021.

[282] Y. Wu, X. Liu, Y. Feng, Z. Wang, R. Yan, and D. Zhao. Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)*, pages 5278–5284, Macao, China, August 10-16 2019.

[283] Y. Wu, X. Liu, Y. Feng, Z. Wang, and D. Zhao. Jointly learning entity and relation representations for entity alignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 240–249, Hong Kong, China, November 3-7 2019.

[284] Y. Wu, X. Liu, Y. Feng, Z. Wang, and D. Zhao. Neighborhood matching network for entity alignment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 6477–6487, Online, July 5-10 2020.

[285] Z. Wu, Y. Chen, B. Kao, and Q. Liu. Perturbed masking: Parameter-free probing for analyzing and interpreting bert. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 4166–4176, Online, July 5-10 2020.

[286] Z. Xi, R. Pang, S. Ji, and T. Wang. Graph backdoor. In *30th USENIX Security Symposium (USENIX Security 2021)*, Online, August 11-13 2021.

[287] C. Xie, K. Huang, P. Chen, and B. Li. DBA: Distributed backdoor attacks against federated learning. In *8th International Conference on Learning Representations (ICLR)*, 2020.

[288] H. Xu, J. Bao, and G. Zhang. Dynamic knowledge graph-based dialogue generation with improved adversarial meta-learning. *CoRR*, abs/2004.08833, 2020.

[289] H. Xu, Y. Ma, H. Liu, D. Deb, H. Liu, J. Tang, and A. K. Jain. Adversarial attacks and defenses in images, graphs and text: A review. *CoRR*, abs/1909.08072, 2019.

[290] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057, 2015.

[291] K. Xu, H. Chen, S. Liu, P.-Y. Chen, T.-W. Weng, M. Hong, and X. Lin. Topology attack and defense for graph neural networks: An optimization perspective. *IJCAI*, pages 3961–3967, 2019.

[292] K. Xu, H. Chen, S. Liu, P.-Y. Chen, T.-W. Weng, M. Hong, and X. Lin. Topology attack and defense for graph neural networks: An optimization perspective. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)*, pages 3961–3967, Macao, China, August 10-16 2019.

[293] K. Xu, L. Song, Y. Feng, Y. Song, and D. Yu. Coordinated reasoning for cross-lingual knowledge graph alignment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020)*, pages 9354–9361, New York, NY, USA, February 7-12 2020.

[294] K. Xu, L. Wang, M. Yu, Y. Feng, Y. Song, Z. Wang, and D. Yu. Cross-lingual knowledge graph alignment via graph matching neural network. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019), Volume 1: Long Papers*, pages 3156–3161, Florence, Italy, July 28-August 2 2019.

[295] X. Xu, Z. Wang, E. Zhang, K. Wang, and H. Shi. Versatile diffusion: Text, images and variations all in one diffusion model, 2024.

[296] Y. Yan, L. Liu, Y. Ban, B. Jing, and H. Tong. Dynamic knowledge graph alignment. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021)*, 2021.

[297] G. Yang, T. Duan, J. E. Hu, H. Salman, I. Razenshteyn, and J. Li. Randomized smoothing of all shapes and sizes. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10693–10705. PMLR, 13–18 Jul 2020.

[298] Y. Yang, B. Hui, H. Yuan, N. Gong, and Y. Cao. Sneakyprompt: Jailbreaking text-to-image generative models. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 122–122, Los Alamitos, CA, USA, may 2024. IEEE Computer Society.

[299] A. Yasar and Ümit V. Çatalyürek. An iterative global structure-assisted labeled network aligner. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2018)*, pages 2614–2623, 2018.

[300] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023.

[301] R. Ye, X. Li, Y. Fang, H. Zang, and M. Wang. A vectorized relational graph convolutional network for multi-relational network alignment. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)*, pages 4135–4141, Macao, China, August 10-16 2019.

[302] D. Yu, Y. Yang, R. Zhang, and Y. Wu. Knowledge embedding based graph convolutional network. In *WWW '21: The Web Conference 2021*, April 19-23 2021.

[303] W. Yu, C. Zhu, Y. Fang, D. Yu, S. Wang, Y. Xu, M. Zeng, and M. Jiang. Dict-bert: Enhancing language model pre-training with dictionary. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1907–1918, 2022.

[304] S. Zahedi and A.-K. Tornberg. Delta function approximations in level set methods by distance function extension. *Journal of Computational Physics*, 229(6):2199–2219, 2010.

[305] Y. Zang, F. Qi, C. Yang, Z. Liu, M. Zhang, Q. Liu, and M. Sun. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 6066–6080, Online, July 5-10 2020.

[306] W. Zeng, X. Zhao, W. Wang, J. Tang, and Z. Tan. Degree-aware alignment for entities in tail. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)*, pages 811–820, Virtual Event, China, July 25-30 2020.

[307] S. Zhai, Y. Dong, Q. Shen, S. Pu, Y. Fang, and H. Su. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 1577–1587, New York, NY, USA, 2023. Association for Computing Machinery.

[308] G. Zhang, Y. Zhou, S. Wu, Z. Zhang, and D. Dou. Cross-lingual entity alignment with adversarial kernel embedding and adversarial knowledge translation. *CoRR*, abs/2104.07837, 2021.

[309] H. Zhang, T. Zheng, J. Gao, C. Miao, L. Su, Y. Li, and K. Ren. Data poisoning attack against knowledge graph embedding. In *Proceedings of the Twenty-Eighth International*

Joint Conference on Artificial Intelligence (IJCAI 2019), pages 4853–4859, Macao, China, August 10-16 2019.

[310] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3836–3847, 2023.

[311] S. Zhang and H. Tong. Final: Fast attributed network alignment. In *KDD*, pages 1345–1354, 2016.

[312] S. Zhang, H. Tong, R. Maciejewski, and T. Eliassi-Rad. Multi-level network alignment. In *WWW*, pages 2344–2354, 2019.

[313] S. Zhang, H. Tong, J. Tang, J. Xu, and W. Fan. ineat: Incomplete network alignment. In *ICDM*, pages 1189–1194, 2017.

[314] S. Zhang, H. Yin, T. Chen, Q. V. N. Hung, Z. Huang, and L. Cui. Gcn-based user representation learning for unifying robust recommendation and fraudster detection. In *SIGIR*, pages 689–698, 2020.

[315] W. E. Zhang, Q. Z. Sheng, A. R. F. Alhazmi, and C. Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):24:1–24:41, 2020.

[316] X. Zhang and M. Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. In *NeurIPS*, 2020.

[317] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu. Cosnet: Connecting heterogeneous social networks with local and global consistency. In *KDD*, pages 1485–1494, 2015.

[318] Z. Zhang, Q. Zhang, Z. Gao, R. Zhang, E. Shutova, S. Zhou, and S. Zhang. Gradient-based Parameter Selection for Efficient Fine-Tuning . In *2024 IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 28566–28577, Los Alamitos, CA, USA, June 2024.

[319] Z. Zhang, Z. Zhang, Y. Zhou, Y. Shen, R. Jin, and D. Dou. Adversarial attacks on deep graph matching. In *NeurIPS*, 2020.

[320] K. Zhao, X. Zhao, Z. Jin, Y. Yang, W. Tao, C. Han, S. Li, and L. Liu. Enhancing baidu multimodal advertisement with chinese text-to-image generation via bilingual alignment and caption synthesis. In G. H. Yang, H. Wang, S. Han, C. Hauff, G. Zuccon, and Y. Zhang, editors, *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 2855–2859. ACM, 2024.

[321] S. Zhao, D. Chen, Y.-C. Chen, J. Bao, S. Hao, L. Yuan, and K.-Y. K. Wong. Unicontrolnet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 2023.

[322] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14431–14440, 2020.

[323] X. Zhao, Z. Zhang, Z. Zhang, L. Wu, J. Jin, Y. Zhou, R. Jin, D. Dou, and D. Yan. Expressive 1-lipschitz neural networks for robust multiple graph learning against adversarial attacks. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, pages 12719–12735, Virtual Event, 2021.

[324] C. Zheng, B. Zong, W. Cheng, D. Song, J. Ni, W. Yu, H. Chen, and W. Wang. Robust graph representation learning via neural sparsification. In *ICML*, 2020.

[325] F. Zhou, L. Liu, K. Zhang, G. Trajcevski, J. Wu, and T. Zhong. Deeplink: A deep learning approach for user identity linkage. In *Proceedings of the 37th IEEE International Conference on Computer Communications (INFOCOM 2018)*, pages 1313–1321, 2018.

[326] K. Zhou, T. P. Michalak, and Y. Vorobeychik. Adversarial robustness of similarity-based link prediction. In *ICDM*, pages 2199–2219, 2019.

[327] K. Zhou, T. P. Michalak, M. Waniek, T. Rahwan, and E. Vorobeychik. Attacking similarity-based link prediction in social networks. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '19)*, pages 305–313, Montreal, QC, Canada, May 13-17 2019.

[328] X. Zhou, L. Cao, M. Guo, and Z. Nie. Colink: An unsupervised framework for user identity linkage. In *AAAI*, pages 5714–5721, 2018.

[329] X. Zhou, X. Liang, X. Du, and J. Zhao. Structure based user identification across social networks. In *TKDE*, volume 30, pages 1178–1191, 2018.

[330] Y. Zhou. Innovative mining, processing, and application of big graphs. *Ph.D. Dissertation*, 2017.

[331] Y. Zhou. *Innovative Mining, Processing, and Application of Big Graphs*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2017.

[332] Y. Zhou, A. Amimeur, C. Jiang, D. Dou, R. Jin, and P. Wang. Density-aware local siamese autoencoder network embedding with autoencoder graph clustering. In *Proceedings of the 2018 IEEE International Conference on Big Data (BigData'18)*, pages 1162–1167, Seattle, WA, 2018.

[333] Y. Zhou, A. Amineur, C. Jiang, D. Dou, R. Jin, and P. Wang. Density-aware local siamese autoencoder network embedding with autoencoder graph clustering. In *BigData*, pages 1162–1167, 2018.

[334] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment (PVLDB)*, 2(1):718–729, 2009.

[335] Y. Zhou, H. Cheng, and J. X. Yu. Clustering large attributed graphs: An efficient incremental approach. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM'10)*, pages 689–698, Sydney, Australia, 2010.

[336] Y. Zhou, C. Jiang, Z. Zhang, D. Dou, R. Jin, and P. Wang. Integrating local vertex/edge embedding via deep matrix fusion and siamese multi-label classification. In *Proceedings of the 2019 IEEE International Conference on Big Data (BigData'19)*, pages 1018–1027, Los Angeles, CA, 2019.

[337] Y. Zhou, K. Lee, L. Liu, Q. Zhang, and B. Palanisamy. Enhancing collaborative filtering with multi-label classification. In *Proceedings of the 2019 International Conference on Computational Data and Social Networks (CSoNet'19)*, pages 323–338, Ho Chi Minh City, Vietnam, 2019.

[338] Y. Zhou and L. Liu. Clustering analysis in large graphs with rich attributes. In D. E. Holmes and L. C. Jain, editors, *Data Mining: Foundations and Intelligent Paradigms, Volume 1: Clustering, Association and Classification*, pages 23–44. 2012.

[339] Y. Zhou and L. Liu. Social influence based clustering of heterogeneous information networks. In *Proceedings of the 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'13)*, pages 338–346, Chicago, IL, 2013.

[340] Y. Zhou and L. Liu. Activity-edge centric multi-label classification for mining heterogeneous information networks. In *Proceedings of the 20th ACM SIGKDD Conference*

on Knowledge Discovery and Data Mining (KDD'14), pages 1276–1285, New York, NY, 2014.

[341] Y. Zhou and L. Liu. Social influence based clustering and optimization over heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1):1–53, 2015.

[342] Y. Zhou and L. Liu. Approximate deep network embedding for mining large-scale graphs. In *Proceedings of the 2019 IEEE International Conference on Cognitive Machine Intelligence (CogMI'19)*, pages 53–60, Los Angeles, CA, 2019.

[343] Y. Zhou, L. Liu, and D. Buttler. Integrating vertex-centric clustering with edge-centric clustering for meta path graph analysis. In *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'15)*, pages 1563–1572, Sydney, Australia, 2015.

[344] Y. Zhou, L. Liu, K. Lee, B. Palanisamy, and Q. Zhang. Improving collaborative filtering with social influence over heterogeneous information networks. *ACM Transactions on Internet Technology (TOIT)*, 20(4):36:1–36:29, 2020.

[345] Y. Zhou, L. Liu, K. Lee, C. Pu, and Q. Zhang. Fast iterative graph computation with resource aware graph parallel abstractions. In *Proceedings of the 24th ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC'15)*, pages 179–190, Portland, OR, 2015.

[346] Y. Zhou, L. Liu, K. Lee, and Q. Zhang. Graphtwist: Fast iterative graph computation with two-tier optimizations. *Proceedings of the VLDB Endowment (PVLDB)*, 8(11):1262–1273, 2015.

[347] Y. Zhou, L. Liu, C.-S. Perng, A. Sailer, I. Silva-Lepe, and Z. Su. Ranking services by service network structure and service attributes. In *Proceedings of the 20th International Conference on Web Services (ICWS'13)*, pages 26–33, Santa Clara, CA, 2013.

[348] Y. Zhou, L. Liu, C. Pu, X. Bao, K. Lee, B. Palanisamy, E. Yigitoglu, and Q. Zhang. Clustering service networks with rich attribute, link and heterogeneity. In *Proceedings of the 22nd International Conference on Web Services (ICWS'15)*, pages 257–264, New York, NY, 2015.

[349] Y. Zhou, L. Liu, S. Seshadri, and L. Chiu. Analyzing enterprise storage workloads with graph modeling and clustering. *IEEE Journal on Selected Areas in Communications (JSAC)*, 34(3):551–574, 2016.

[350] Y. Zhou, J. Ren, D. Dou, R. Jin, J. Zheng, and K. Lee. Robust meta network embedding against adversarial attacks. In *ICDM*, pages 1448–1453, 2020.

[351] Y. Zhou, J. Ren, D. Dou, R. Jin, J. Zheng, and K. Lee. Robust network embedding against adversarial attacks. In *Proceedings of the 20th IEEE International Conference on Data Mining (ICDM'20)*, pages 1448–1453, Sorrento, Italy, 2020.

[352] Y. Zhou, J. Ren, R. Jin, Z. Zhang, D. Dou, and D. Yan. Unsupervised multiple network alignment with multinomial gan and variational inference. In *BigData*, page Online, 2020.

[353] Y. Zhou, J. Ren, R. Jin, Z. Zhang, D. Dou, and D. Yan. Unsupervised multiple network alignment with multinomial gan and variational inference. In *Proceedings of the 2020 IEEE International Conference on Big Data (BigData'20)*, pages 868–877, Atlanta, GA, 2020.

[354] Y. Zhou, J. Ren, R. Jin, Z. Zhang, J. Zheng, Z. Jiang, D. Yan, and D. Dou. Unsupervised adversarial network alignment with reinforcement learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2021. To appear.

[355] Y. Zhou, J. Ren, S. Wu, D. Dou, R. Jin, Z. Zhang, and P. Wang. Semi-supervised classification-based local vertex ranking via dual generative adversarial nets. In *BigData*, pages 1267–1273, 2019.

[356] Y. Zhou, J. Ren, S. Wu, D. Dou, R. Jin, Z. Zhang, and P. Wang. Semi-supervised classification-based local vertex ranking via dual generative adversarial nets. In *Proceedings of the 2019 IEEE International Conference on Big Data (BigData'19)*, pages 1267–1273, Los Angeles, CA, 2019.

[357] Y. Zhou, S. Seshadri, L. Chiu, and L. Liu. Graphlens: Mining enterprise storage workloads using graph analytics. *BigData*, pages 1–8, 2014.

[358] Y. Zhou, S. Wu, C. Jiang, Z. Zhang, D. Dou, R. Jin, and P. Wang. Density-adaptive local edge representation learning with generative adversarial network multi-label edge classification. In *ICDM*, pages 1464–1469, 2018.

[359] Y. Zhou, Z. Zhang, S. Wu, V. Sheng, X. Han, Z. Zhang, and R. Jin. Robust network alignment via attack signal scaling and adversarial perturbation elimination. In *Proceedings of the 30th Web Conference (WWW'21)*, pages 3884–3895, Virtual Event / Ljubljana, Slovenia, 2021.

[360] D. Zhu, Z. Zhang, P. Cui, and W. Zhu. Robust graph convolutional networks against adversarial attacks. In *KDD*, pages 1399–1407, 2019.

[361] H. Zhu, R. Xie, Z. Liu, and M. Sun. Iterative entity alignment via joint knowledge embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017)*, pages 4258–4264, Melbourne, Australia, August 19-25 2017.

[362] Q. Zhu, H. Wei, B. Sisman, D. Zheng, C. Faloutsos, X. L. Dong, and J. Han. Collective multi-type entity alignment between knowledge graphs. In *WWW '20: The Web Conference 2020*, pages 2241–2252, Taipei, Taiwan, April 20-24 2020.

[363] Q. Zhu, X. Zhou, J. Wu, J. Tan, and L. Guo. Neighborhood-aware attentional representation for multilingual knowledge graphs. In *IJCAI*, pages 1943–1949, 2019.

[364] Q. Zhu, X. Zhou, J. Wu, J. Tan, and L. Guo. Neighborhood-aware attentional representation for multilingual knowledge graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)*, pages 1943–1949, Macao, China, August 10-16 2019.

[365] Y. Zhu, H. Liu, Z. Wu, and Y. Du. Relation-aware neighborhood matching model for entity alignment. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021)*, 2021.

[366] D. Zügner, A. Akbarnejad, and S. Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2018)*, pages 2847–2856, London, UK, August 19-23 2018.

[367] D. Zügner, O. Borchert, A. Akbarnejad, and S. Günnemann. Adversarial attacks on graph neural networks: Perturbations and their patterns. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(5):57:1–57:31, 2020.

[368] D. Zügner and S. Günnemann. Adversarial attacks on graph neural networks via meta learning. In *ICLR*, 2019.

[369] D. Zügner and S. Günnemann. Certifiable robustness and robust training for graph convolutional networks. In *KDD*, pages 246–256, 2019.

[370] D. Zügner and S. Günnemann. Certifiable robustness of graph convolutional networks under structure perturbations. In *KDD*, 2020.