On the Robustness of Metric Learning: An Adversarial Perspective

MENGDI HUAI, University of Virginia
TIANHANG ZHENG, University of Toronto
CHENGLIN MIAO, University of Georgia
LIUYI YAO, Alibaba Group
AIDONG ZHANG, University of Virginia

Metric learning aims at automatically learning a distance metric from data so that the precise similarity between data instances can be faithfully reflected, and its importance has long been recognized in many fields. An implicit assumption in existing metric learning works is that the learned models are performed in a reliable and secure environment. However, the increasingly critical role of metric learning makes it susceptible to a risk of being malicious attacked. To well understand the performance of metric learning models in adversarial environments, in this article, we study the robustness of metric learning to adversarial perturbations, which are also known as the imperceptible changes to the input data that are crafted by an attacker to fool a well-learned model. However, different from traditional classification models, metric learning models take instance pairs rather than individual instances as input, and the perturbation on one instance may not necessarily affect the prediction result for an instance pair, which makes it more difficult to study the robustness of metric learning. To address this challenge, in this article, we first provide a definition of pairwise robustness for metric learning, and then propose a novel projected gradient descent-based attack method (called AckMetric) to evaluate the robustness of metric learning models. To further explore the capability of the attacker to change the prediction results, we also propose a theoretical framework to derive the upper bound of the pairwise adversarial loss. Finally, we incorporate the derived bound into the training process of metric learning and design a novel defense method to make the learned models more robust. Extensive experiments on real-world datasets demonstrate the effectiveness of the proposed methods.

CCS Concepts: • **Information systems** → **Data mining**; • **Security and privacy**; • **Computing methodologies** → *Machine learning*;

Additional Key Words and Phrases: Metric learning, robustness, adversarial perturbations

This work is supported in part by the US National Science Foundation under grants IIS-1938167, IIS-1955151, IIS-2008208, IIS-2106913, and OAC-1934600. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
Authors' addresses: M. Huai and A. Zhang, Department of Computer Science, University of Virginia, 85 Engineer's Way, Charlottesville, VA 22904; emails: {mh6ck, aidong}@virginia.edu; T. Zheng, Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Rd, Toronto, ON M5S 3G8, Canada; email: th.zheng@mail.utoronto.ca; C. Miao, Department of Computer Science, University of Georgia, Boyd Graduate Studies Research Center, D. W. Brooks Drive, Athens, GA 30602; email: cmiao@uga.edu; L. Yao, Alibaba Group, 969 West Wen Yi Road, Yu Hang District, Hangzhou, Zhejiang 311121, China; email: yly287738@alibaba-inc.com.

1556-4681/2022/04-ART95 $15.00
https://doi.org/10.1145/3502726

ACM Reference format:
Mengdi Huai, Tianhang Zheng, Chenglin Miao, Liuyi Yao, and Aidong Zhang. 2022. On the Robustness of Metric Learning: An Adversarial Perspective. *ACM Trans. Knowl. Discov. Data.* 16, 5, Article 95 (April 2022), 25 pages.
https://doi.org/10.1145/3502726

1 INTRODUCTION

The calculation of distance or similarity between data instances serves as an important basis for many machine learning and data mining tasks. For example, in the task of face verification [44], the distance measurement between two instances plays an important role in determining whether they belong to the same class or not; in the task of medical diagnosis, the diagnosis for a patient is usually dependent on the similarity measurement between this patient and others [49, 72]; in the task of image retrieval [9], images are typically ranked according to the similarity scores, which measure their relevance to a given query. Although some simple metrics (e.g., Euclidean distance) can be used to measure the similarity between data instances, they usually fail to capture the idiosyncrasies of the data of interest. To address this challenge, the topic of metric learning, which aims at automatically learning a distance metric from data so that the precise similarity between data instances can be faithfully reflected, has drawn significant attention [17, 18, 24, 60].

An implicit assumption in existing metric learning works is that the learned models are performed in a reliable and secure environment [20]. However, as metric learning plays an increasingly critical role in many machine learning and data mining tasks, it is susceptible to a risk of being malicious attacked. For example, in face verification, an attacker might attempt to introduce imperceptible perturbations to some face images belonging to the same class so that they will be deemed to be quite different by a well-learned metric model. In medical diagnosis, an attacker might collude with a drug maker, and in order to recommend a particular drug to some patients, he may carefully perturb the patients' clinical information and let a well performed metric model output high similarity scores when comparing these patients with those who take this drug. In both cases, the well-learned metric models may be fooled by the malicious activities and provide misleading similarity measurements. To well understand the performance of metric learning models in adversarial environments, in this article, we study their robustness to adversarial perturbations, which are also known as the imperceptible changes to the input data that are crafted by an attacker to fool a well-learned model.

However, the robustness analysis of metric learning is more challenging than that of the traditional classification models. The robustness of a traditional classification model is usually defined based on the minimal perturbation on an instance that is required to change the assigned label, and such robustness is called pointwise robustness [2] as it treats each instance independently. But for a learned metric model, since its goal is to precisely measure the similarity between instances and further used to predict their similarity label (i.e., similar or dissimilar), the input are usually instance pairs rather than individual instances. Thus, the predicted similarity label of each instance pair is determined by the relative comparison of the two instances, and the perturbations on one of them do not necessarily affect the assigned similarity label, which makes it more difficult to study the robustness of metric learning models.

To address the above challenges, in this article, we first provide a definition of *pairwise robustness* for metric learning. Based on this definition, we can study the effect of adversarial perturbations on the similarity degrees of instance pairs. Then, we evaluate the robustness of metric learning via designing a novel projected gradient descent-based attack method (called AckMetric), based on which the attacker can craft adversarial instance pairs to fool a well-learned metric model. To

further explore the capability of the attacker to change the prediction results of a metric learning model, we also propose a theoretical framework to derive the *upper bound* of the pairwise adversarial loss. The derived upper bound is attack-independent, and it serves as a certificate that for a given metric learning model and test input, there is no attack that can force the introduced error to exceed a certain value. Last but not least, to make the learned metric models more robust to adversarial perturbations, we propose to incorporate the derived upper bound into the training process of metric learning and design a novel *defense mechanism*. To the best of our knowledge, this is the first work to study the robustness of metric learning to adversarial perturbations. Extensive experiments on real-world datasets demonstrate the effectiveness of the proposed methods.

2 ROBUSTNESS ANALYSIS FOR METRIC LEARNING

In this section, we study the robustness of metric learning models via crafting adversarial perturbations. Specifically, we firstly revisit the metric learning model for the sake of self-containedness. Then, we define the notion of pairwise robustness of metric learning via generalizing the concept of adversarial examples to adversarial instance pairs. For simplicity and without loss of generality, in the following, we mainly focus (unless otherwise stated) on the scenarios where the attacker aims at changing a similar instance pair to a dissimilar pair via adversarial perturbations. Lastly, we propose a variant of projected gradient attack on metric learning (AckMetric), which shows that state-of-the-art metric learning models are vulnerable to adversarial perturbations.

2.1 A Revisit to Metric Learning

Suppose there is a set of instances $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_i\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is d-dimensional. The goal of metric learning is to learn a mapping from the training set so that each instance can be projected into a new feature space, based on which the similarity degree of the pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ can be calculated. Without loss of generality; here, we normalize data to range $[0, 1]$. Existing metric learning models can be divided into the following two categories: *linear* and *nonlinear*.

The *linear* models aim at learning a linear mapping $\boldsymbol{W} \in \mathbb{R}^{d*d}$, based on which each instance $\boldsymbol{x}_i$ can be projected into a new feature space $f(\boldsymbol{x}_i) = \boldsymbol{W}\boldsymbol{x}_i$ [6, 17, 57, 71]. Although we aim at developing a general method to study the robustness of metric learning models with different distance functions, without loss of generality, we use the widely adopted Mahalanobis distance function to present our idea. Based on the Mahalanobis distance, the similarity degree of the instance pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ can be calculated as follows:

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = (f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j))^T (f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)). \tag{1}$$

The *nonlinear* metric learning models (i.e., deep metric learning models) aim at capturing the nonlinear structures of the instances. In practice, the nonlinear models [7, 8, 16, 24, 32, 55] usually adopt deep neural networks to capture the nonlinear structures of the instances. For example, [7] proposes a new deep metric learning framework that utilizes the generated adequate hard negatives (instead of the observed negatives) to train the distance metric to fully exploit the potential of each negative sample. Hence, we mainly focus on the deep metric learning models that have drawn significant attention recently due to the success of deep learning. The basic idea of deep metric learning models is to explicitly train a L-layer deep neural network, based on which a set of hierarchical nonlinear mappings can be derived to project the original input instances into a new feature space for comparing. The derived nonlinear mappings are capable of guaranteeing that the distance between similar samples is small and the distance between dissimilar samples is large in the new feature space [45, 62, 70]. Assume that the trained L-layer neural network is parameterized by the weights $\{\boldsymbol{W}^l \in \mathbb{R}^{h_l * h_{l-1}}\}_{l=1}^L$ (note that the biases are included in the weights with a corresponding fixed input of 1 for simplicity), where h_l represents the number of neurons in

the lth layer of the network and $h_0 = d$. Then, given the input instance $\boldsymbol{x}_i \in \mathbb{R}^d$, the output of the lth layer in the network can be written as $f^l(\boldsymbol{x}_i) = \boldsymbol{W}^l\sigma(f^{l-1}(\boldsymbol{x}_i)) = \boldsymbol{W}^l\sigma(\boldsymbol{W}^{l-1}\sigma(\cdots\sigma(\boldsymbol{W}^1\boldsymbol{x}_i)))$, where $\sigma(\cdot)$ denotes the activation function. In particular, $f^1(\boldsymbol{x}_i) = \boldsymbol{W}^1\boldsymbol{x}_i$. In this case, the similarity degree of the instance pair $(\boldsymbol{x}_i,\boldsymbol{x}_j)$ in the learned nonlinear feature space is calculated as

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(f^L(\boldsymbol{x}_i) - f^L(\boldsymbol{x}_j)\right)^T \left(f^L(\boldsymbol{x}_i) - f^L(\boldsymbol{x}_j)\right), \tag{2}$$

where $f^L(\boldsymbol{x}_i) = \boldsymbol{W}^L\sigma(\boldsymbol{W}^{L-1}\sigma(\cdots\sigma(\boldsymbol{W}^1\boldsymbol{x}_i)))$.

Existing metric learning approaches are usually formulated based on some pre-defined similarity and dissimilarity constraints during the training process, which require that the distance between two instances should be less than a threshold if they are in the same class, otherwise the distance should be larger than this threshold [6, 14, 43, 58, 60, 61, 72]. Formally, the constraints usually have the following forms

$$\begin{cases} D(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq \gamma, & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ have the same class label,} \\ D(\boldsymbol{x}_i, \boldsymbol{x}_j) > \gamma, & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ have different class labels,} \end{cases} \tag{3}$$

where γ denotes the threshold that is used to train the learning model. By enforcing these constraints in the designed optimization framework, existing metric learning approaches can build a model that maximizes the between-class distance and minimizes the within-class distance.

2.2 Pairwise Robustness of Metric Learning

Our ultimate goal is to ensure the robustness of a well-learned metric model in the testing stage. For each instance pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, after calculating its similarity degree according to Equation (1) or Equation (2), we assume that it will be assigned a similarity label (i.e., similar or dissimilar) by the learned model based on the threshold γ. If $D(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq \gamma$, it will be treated as a similar pair, otherwise it will be treated as a dissimilar pair. Please note that the value of γ is specified according to the practical demand. Since the assigned similarity label of each instance pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is determined by the relative comparison of the two instances (i.e., $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$) and the perturbation on one of them may not necessarily affect the assigned similarity label, the notion of pointwise robustness [2] for traditional classification models does not fit for the analysis of metric learning. To address the above challenge, we formulate a definition of pairwise robustness for metric learning, based on which we can study the effect of adversarial perturbations on the similarity degrees of instance pairs. For simplicity and without loss of generality, in the following, we mainly focus (unless otherwise stated) on the scenarios where the attacker aims at changing a similar instance pair to a dissimilar pair via adversarial perturbations.

Attack model. Following the line of work on adversarial attacks [4, 10, 13, 40, 41], we here assume a white-box setting, which is a conservative and realistic assumption. The attacker in this setting tries to evade the system by manipulating malicious instance pairs during the testing phase. The attacker cannot change the metric learning algorithm used for the training of the learner, and the attacker can only change the instance pairs during the testing stage. The attacker's goal is to deceive the trained metric learning model. Specifically, since the goal of metric learning is to learn a distance metric that is used to calculate the similarity degree of different instance pairs, the attacker here aims at crafting adversarial perturbations to alter the similarity degrees of instance pairs. For simplicity and without loss of generality, in the following, we mainly focus (unless otherwise stated) on the scenarios where the attacker aims at changing a similar instance pair to a dissimilar pair via adversarial perturbations.[1]

[1]More discussions on the scenarios where the attacker aims at changing a dissimilar instance pair to a similar instance pair are given in Section 5.1.

Our work is inspired by the recent developments of adversarial attacks against deep learning, which show that deep learning models are vulnerable to adversarial examples. Here, we give an brief introduction of existing adversarial attack works [4, 10, 13, 40, 41, 74, 75]. Adversarial attacks are manipulative actions that aim at undermining machine-learning performance and cause model misbehavior. Specifically, an adversarial attack is a technique to find an adversarial perturbation to craft an adversarial example, which is intentionally designed to cause the machine-learning model to make a mistake. In other words, an adversarial example is an input that has been modified in a way that is imperceptible to humans, but is misclassified by a machine-learning system whereas the original input was correctly classified [74, 75]. For the generation of the adversarial examples, two conditions should be satisfied. The first condition is that the added adversarial perturbations are imperceptible to humans when comparing an original input $\boldsymbol{x}$ and its perturbed version $\tilde{\boldsymbol{x}}$ side by side. The second one is that the original clean input $\boldsymbol{x}$ and its perturbed version $\tilde{\boldsymbol{x}}$ are correctly and incorrectly classified by the prediction model, respectively. That is to say, an adversarial example is an instance with imperceptible and intentional feature perturbations that cause a machine-learning model to make a false prediction. In the following, we generalize the definition of pointwise adversarial examples to adversarial instance pairs where the adversarial perturbations are simultaneously added to both of the two instances in each instance pair. Note that the predicted similarity label of each instance pair is determined by the relative comparison of the two instances, and the perturbations on one of them do not necessarily affect the assigned similarity label, which makes it more difficult to study the robustness of metric learning models. In contrast, the robustness of a traditional pointwise model is usually defined based on the minimal perturbation on a specific instance that is required to change the assigned prediction result [2].

Definition 2.1 (Pairwise Robustness of Metric Learning). Note that given a well-trained metric learning model (i.e., D) and an instance pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the similarity degree of this instance pair in the newly learned feature space can be calculated as $D(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Without loss of generality, we further assume that this given instance pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a similar instance pair that satisfies $D(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq \gamma$. The pairwise robustness of the given distance metric function D with respect to the instance pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is defined as $\rho_{pair}(D; (\boldsymbol{x}_i, \boldsymbol{x}_j)) := \min_{\boldsymbol{\delta}_i \in \mathbb{R}^d, \boldsymbol{\delta}_j \in \mathbb{R}^d} \|\boldsymbol{\delta}_i\|_p + \|\boldsymbol{\delta}_j\|_p$ s.t. $D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) > \gamma$, where $\boldsymbol{\delta}_i \in \mathbb{R}^d$ and $\boldsymbol{\delta}_j \in \mathbb{R}^d$ denote the perturbations on instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively. $\|\cdot\|_p$ denotes the p-norm.

Note that in the above definition, the types of the adversarial perturbations applied in the proposed pairwise adversarial attacks depend on the targeted data and desired effect, and the adversarial perturbations need to be customized for different data to be reasonably adversarial. Specifically, for each targeted data, we generate its adversarial perturbations in the direction of the gradient, which means that the images are intentionally altered so that the model fails. Formally, based on the above definition, the attacker can generate the adversarial instance pair through maximizing the margin $\Delta(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j)$, and the attack will succeed if $\Delta(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) > 0$. That is to say, the computation of an adversarial pairwise perturbation for a new data requires solving a data-dependent optimization problem from scratch, which uses the full knowledge of the model. Additionally, the types of perturbations applied in adversarial attacks also depend on the target data type. For instance, for the image and audio data, it makes sense to consider small data perturbation as a threat model because it will not be easily perceived by a human but may make the target model to misbehave, causing inconsistency between human and machine. However, for some data types such as text, perturbation (by simply changing a word or a character) may disrupt the semantics and can easily be detected by humans. Therefore, the threat model for text should be naturally different from image or audio.

In the above definition, the magnitude of the added adversarial perturbations reflects the robustness of the attacked metric learning model. Note that the above definition of the pairwise robustness of metric learning corresponds to finding pairwise adversarial perturbations on an instance pair. Importantly, one character of the above defined pairwise robustness is that the smaller the magnitude of the adversarial perturbations (i.e., the value of $\rho_{pair}(D;(\boldsymbol{x}_i,\boldsymbol{x}_j))$) is, the easier an adversarial instance pair (i.e., $(\boldsymbol{x}_i+\boldsymbol{\delta}_i,\boldsymbol{x}_j+\boldsymbol{\delta}_j)$) can be generated, the less robust the learned metric learning model is. Another character of the adversarial perturbations is their intrinsic dependence on datapoints: The pairwise adversarial perturbations are specifically crafted for each data independently. As a result, the computation of an adversarial perturbation for a new data requires solving a data-dependent optimization problem from scratch, which uses the full knowledge of the model. One of the most intriguing characters about the defined adversarial pairwise perturbations is their transferability across different models. The perturbed adversarial instance pairs can transfer between different metric learning models: Adversarial instance pairs generated based on a specific model will often fool other unseen models with a significant success rate. This allows the attacker to leverage it to attack the deployed systems without any query, which can raise severe security issue particularly in safety-critical scenarios. In practice, the pairwise adversarial transferability is usually influenced by several important factors, e.g., the model architecture and local smoothness of loss surface for generating adversarial pairs.

2.3 Adversarial Attacks against Metric Learning

In this section, we first design a projected gradient descent-based attack framework to study the robustness of metric learning, which can craft adversarial instance pairs to fool the well-learned metric models. Note that an adversarial attack on a metric learning model is a process for generating adversarial perturbations. Then, we present the optimization solution to the formulated attack framework to find adversarial instance pairs by using the gradient of the underlying metric learning model. After that, we present the optimization solutions for the linear and nonlinear metric learning models. Note that the robustness of metric learning can be reflected by how easy it is to craft adversarial pairs.

Overview. The key idea of the proposed attack is to find an adversarial instance pair through solving a constrained optimization problem. Specifically, the proposed AckMetric attempts to find the pairwise adversarial perturbation that minimizes the pairwise adversarial loss of a metric learning model on a particular instance pair while keeping the size of the perturbation smaller than a specified amount referred to as ϵ. In our adversarial setting, this constraint on the pairwise adversarial perturbations is expressed as the L_∞ norm of the adversarial perturbation and it is added so the content of the adversarial instance pair is the same as the unperturbed instance pair' or even such that the adversarial instance pair is imperceptibly different to humans. Here, we follow the general security analysis methodology for adversarial learning [59] and assume that the attacker has full knowledge of the attacked metric learning model.

The proposed attack. Here, we present the proposed adversarial attack (i.e., AckMetric) on the metric learning models, which is a constrained optimization problem. Note that the adversarial attack on a metric learning model is a process for generating adversarial perturbations. In our proposed adversarial attack against metric learning, for each similar instance pair $(\boldsymbol{x}_i,\boldsymbol{x}_j)$ whose similarity is no larger than the pre-defined threshold (i.e., $\Delta(\boldsymbol{x}_i,\boldsymbol{x}_j)=D(\boldsymbol{x}_i,\boldsymbol{x}_j)-\gamma\leq 0$), the attacker simultaneously crafts its adversarial instance pair through adding the adversarial perturbations to both of the two instances in this instance pair, such that the well-learned distance function D is fooled to report a dissimilar label. More specifically, we formalize the generation of the adversarial

instance pairs as a solution to the following optimization problem:

$$\max_{||\boldsymbol{\delta}_i||_\infty + ||\boldsymbol{\delta}_j||_\infty < \epsilon} \quad D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) - \gamma,$$
$$\text{s.t.} \quad \boldsymbol{x}_i + \boldsymbol{\delta}_i \in [0,1]^d,\ \boldsymbol{x}_j + \boldsymbol{\delta}_j \in [0,1]^d, \tag{4}$$

where ϵ controls the magnitude of adversarial pairwise perturbations. By solving the above optimization problem, the attacker can achieve the attack goal, i.e., finding an optimal pair $(\boldsymbol{\delta}_i, \boldsymbol{\delta}_j)$ that can maximize the margin $\Delta(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j)$. The attack will succeed if $\Delta(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) > 0$. The constraints in the above equation enforce that the generated adversarial examples lie in the range of $[0, 1]$. Specifically, by following existing adversarial works [10, 38, 51], we use the clipping function to ensure that the generated adversarial examples are in the valid range (i.e., $[0, 1]$).

By solving the above optimization problem in Equation (4), the attacker can craft adversarial instance pairs to fool the metric learning models to make wrong pairwise predictions. In the above, the goal of the attacker is to find an optimal adversarial perturbation pair $(\boldsymbol{\delta}_i, \boldsymbol{\delta}_j)$ that can maximize the margin $\Delta(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) = D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) - \gamma$. In other words, the attacker in the above launches the optimal attack by choosing the perturbed pair $(x_i + \delta_i, x_j + \delta_j)$ that maximizes the distance function $D(x_i + \delta_i, x_j + \delta_j)$. And, this optimal attack is successful if $\Delta(x_i + \delta_i, x_j + \delta_j) = D(x_i + \delta_i, x_j + \delta_j) - \gamma > 0$, which means that the adversarial loss over the instance pair (x_i, x_j) is equal to 1 (i.e., $\mathcal{L}_A(x_i, x_j) = \mathbb{I}[\Delta(x_i + \delta_i, x_j + \delta_j) > 0] = 1$).

Optimization. Next, we discuss how to solve the optimization problem in Equation (4) to craft adversarial instance pairs to fool the metric learning models. The solution here is based on the **projected gradient descent** (**PGD**) method, and it is an iterative procedure. In each iteration, we compare the L_1 norm of $\frac{\partial D(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_i}$ with that of $\frac{\partial D(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}$, and then perform a PGD-like update for the instance pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ along the direction of the gradient with the larger L_1 norm. Specifically, in the $(r+1)$-th iteration, $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is updated as

$$\boldsymbol{x}_k^{r+1} = \Pi_{clip}\left(\boldsymbol{x}_k^r + \xi \cdot \text{sign}\left(\frac{\partial D(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)}{\partial \boldsymbol{x}_k^r}\right)\right), \tag{5}$$

$$k = \arg\max_{\{i,j\}} \left\{ \left\| \frac{\partial D(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)}{\partial \boldsymbol{x}_i^r} \right\|_1, \left\| \frac{\partial D(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)}{\partial \boldsymbol{x}_j^r} \right\|_1 \right\}. \tag{6}$$

Here, $\boldsymbol{x}_i^0 = \boldsymbol{x}_i$, $\boldsymbol{x}_j^0 = \boldsymbol{x}_j$, and ξ is the step size that specifies the value changed by each iteration. In each iteration, by using the element-wise clip function Π_{clip}, we can guarantee that the updated $\boldsymbol{x}_k^{r+1}$ resides in a valid range.

Discussion. Here, we discuss how to use the derived optimization solution (i.e., Equations (5) and (6)) to craft adversarial instance pairs for the linear and nonlinear metric learning models. The above proposed attack (in Equation (4)) is a general attack schema, which can not only attack metric learning models but also can attack deep metric learning models. More specifically, the attacker can use the gradient information to maximize the loss within a small perturbation region to craft adversarial instance pairs. Next, we discuss how to use the gradient information to iteratively calculate the adversarial pairs for the linear and nonlinear metric learning models.

— Based on Equations (5) and (6), we here discuss how to iteratively craft adversarial instance pairs for linear metric learning models. For linear metric learning models, $\frac{\partial D(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)}{\partial \boldsymbol{x}_i^r}$ and

$\frac{\partial D(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)}{\partial \boldsymbol{x}_j^r}$ in Equation (6) are calculated as

$$\frac{\partial D(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)}{\partial \boldsymbol{x}_i^r} = -\frac{\partial D(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)}{\partial \boldsymbol{x}_j^r} = 2\boldsymbol{W}^T\boldsymbol{W}(\boldsymbol{x}_i^r - \boldsymbol{x}_j^r), \tag{7}$$

where $\boldsymbol{W}$ denotes the linear mapping of the linear metric learning models. Note that for nonlinear metric learning models (i.e., deep metric learning models) that aim at training an L-layer deep neural network to learn a set of hierarchical nonlinear mappings, we use $\{\boldsymbol{W}^l\}_{l=1}^L$ to denote the weights (i.e., the hierarchical nonlinear mappings) of the trained L-layer neural network (note that the biases are included in the weights with a corresponding fixed input of 1 for simplicity), where $\boldsymbol{W}^l$ represents the weight of the lth layer of the network. Since the two gradients have the same L_1 norm, we can update either $\boldsymbol{x}_i$ or $\boldsymbol{x}_j$ according to

$$\boldsymbol{x}_i^{r+1} = \Pi_{clip}(\boldsymbol{x}_i^r + \xi \cdot \text{sign}(2\boldsymbol{W}^T\boldsymbol{W}(\boldsymbol{x}_i^r - \boldsymbol{x}_j)), \tag{8}$$

$$\boldsymbol{x}_j^{r+1} = \Pi_{clip}(\boldsymbol{x}_j^r + \xi \cdot \text{sign}(-2\boldsymbol{W}^T\boldsymbol{W}(\boldsymbol{x}_i - \boldsymbol{x}_j^r)). \tag{9}$$

Here, Π_{clip} is the element-wise clip function, and it can guarantee that the updated adversarial pair resides in the valid range. Based on the gradients in Equations (8) and (9), for the instance pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, we can derive its corresponding adversarial instance pair by using Equations (5) and (6).

— Then, we talk about how to use Equations (5) and (6) to calculate the gradient information to iteratively craft adversarial instance pairs for nonlinear metric learning models. For nonlinear metric learning models, the gradient $\frac{\partial D(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)}{\partial \boldsymbol{x}_i^r}$ in Equation (6) is calculated as

$$\begin{aligned}\frac{\partial D(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)}{\partial \boldsymbol{x}_i^r} &= \frac{\partial (f^L(\boldsymbol{x}_i^r) - f^L(\boldsymbol{x}_j^r))^T (f^L(\boldsymbol{x}_i^r) - f^L(\boldsymbol{x}_j^r))}{\partial \boldsymbol{x}_i^r} \\ &= 2(f^L(\boldsymbol{x}_i^r) - f^L(\boldsymbol{x}_j^r))^T \boldsymbol{W}^L \frac{\partial \sigma(\boldsymbol{W}^{L-1} f^{L-1}(\boldsymbol{x}_i^r))}{\partial (\boldsymbol{W}^{L-1} f^{L-1}(\boldsymbol{x}_i^r))} \cdot \boldsymbol{W}^{L-1} \frac{\partial \sigma(\boldsymbol{W}^{L-2} f^{L-2}(\boldsymbol{x}_i^r))}{\partial (\boldsymbol{W}^{L-2} f^{L-2}(\boldsymbol{x}_i^r))} \cdots \boldsymbol{W}^1,\end{aligned} \tag{10}$$

where $f^1(\boldsymbol{x}_i^r) = \boldsymbol{x}_i^r$ and $f^1(\boldsymbol{x}_j^r) = \boldsymbol{x}_j^r$. Note that $\boldsymbol{x}_i^0 = \boldsymbol{x}_i$ and $\boldsymbol{x}_j^0 = \boldsymbol{x}_j$. For the nonlinear metric learning models, the set of hierarchical nonlinear mappings (i.e., the parameters of the trained L-layer neural network) is denoted as $\{\boldsymbol{W}^l\}_{l=1}^L$. Let J^l denote the matrix $\frac{\partial \sigma(\boldsymbol{W}^l f^l(\boldsymbol{x}_i^r))}{\partial (\boldsymbol{W}^l f^l(\boldsymbol{x}_i^r))}$, and its entries $J^l[p, q]$ is defined as follows:

$$J^l[p,q] = \frac{\partial \sigma(\boldsymbol{W}^l f^l(\boldsymbol{x}_i^r))[p]}{\partial (\boldsymbol{W}^l f^l(\boldsymbol{x}_i^r))[q]} = \begin{cases} 1, & \text{if } p = q \text{ and} \\ & (\boldsymbol{W}^l f^l(\boldsymbol{x}_i))[q] \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Obviously, matrix J^l is a diagonal matrix. The gradient $\frac{\partial D(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)}{\partial \boldsymbol{x}_j^r}$ can be calculated in a similar way as follows:

$$\begin{aligned}\frac{\partial D(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)}{\partial \boldsymbol{x}_j^r} &= \frac{\partial (f^L(\boldsymbol{x}_i^r) - f^L(\boldsymbol{x}_j^r))^T (f^L(\boldsymbol{x}_i^r) - f^L(\boldsymbol{x}_j^r))}{\partial \boldsymbol{x}_j^r} \\ &= -2(f^L(\boldsymbol{x}_i^r) - f^L(\boldsymbol{x}_j^r))^T \boldsymbol{W}^L \frac{\partial \sigma(\boldsymbol{W}^{L-1} f^{L-1}(\boldsymbol{x}_j^r))}{\partial (\boldsymbol{W}^{L-1} f^{L-1}(\boldsymbol{x}_j^r))} \cdot \boldsymbol{W}^{L-1} \frac{\partial \sigma(\boldsymbol{W}^{L-2} f^{L-2}(\boldsymbol{x}_j^r))}{\partial (\boldsymbol{W}^{L-2} f^{L-2}(\boldsymbol{x}_j^r))} \cdots \boldsymbol{W}^1,\end{aligned} \tag{11}$$

where $f^1(\boldsymbol{x}_j^r) = \boldsymbol{x}_j^r$. Based on the gradients in Equations (10) and (11), for each instance pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, we can derive its corresponding adversarial instance pair by using Equations (5) and (6).

Based on the above proposed attack, the attacker can generate the pairwise adversarial perturbations that can be added to the original instance pairs to craft the adversarial instance pairs. Importantly, the robustness of the attacked metric learning model can be reflected by the magnitude of the added pairwise adversarial perturbations. Specifically, the smaller the magnitude of the added pairwise adversarial perturbations is, the less robust the learned metric learning model is.

3 CERTIFICATE ON THE PAIRWISE ADVERSARIAL LOSS

Note that the proposed AckMetric allows an attacker to generate adversarial instance pairs and further introduce errors to the similarity prediction results of the attacked metric learning model. More specifically, by following the proposed AckMetric (see Equation (4)), the attacker can craft adversarial perturbations to change a similar instance pair to a dissimilar instance pair. To explore the capability of an attacker to change the prediction results, in this section, we propose a theoretical framework to derive the upper bound of the pairwise adversarial loss. The derived upper bound serves as a certificate that for a given metric learning model and test input, there is no attack that can force the introduced error to exceed a certain value. Based on this certificate, we can improve current metric learning models and make them more robust to adversarial perturbations. In the following, we mainly focus on the derivation for the metric learning models that aim at learning the linear mappings. For the nonlinear metric learning models, we leave the derivation to the future work.

According to Definition 2.2, the pairwise adversarial loss $\mathcal{L}_A(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for an attacker A is determined by $\Delta(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) = D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) - \gamma$. In order to derive the upper bound of the pairwise adversarial loss, we can first bound $D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j)$. The integration expression of the distance function $D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j)$ can be written as follows:

$$D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) = \int_{\boldsymbol{x}_i}^{\boldsymbol{x}_i+\boldsymbol{\delta}_i} \frac{\partial D(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_i} d\boldsymbol{x}_i + \int_{\boldsymbol{x}_j}^{\boldsymbol{x}_j+\boldsymbol{\delta}_j} \frac{\partial D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_j} d\boldsymbol{x}_j + D(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{12}$$

Then, based on Hölder's inequality, the above distance function $D(\boldsymbol{x}_i+\boldsymbol{\delta}_i, \boldsymbol{x}_j+\boldsymbol{\delta}_j)$ can be bounded as

$$\begin{aligned}
D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) \leq & \sup_{[\boldsymbol{x}_i, \boldsymbol{x}_i+\boldsymbol{\delta}_i]} \left\| \frac{\partial D(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_i} \right\|_1 \cdot \|\boldsymbol{\delta}_i\|_\infty \\
& + \sup_{[\boldsymbol{x}_j, \boldsymbol{x}_j+\boldsymbol{\delta}_j]} \left\| \frac{\partial D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_j} \right\|_1 \cdot \left\|\boldsymbol{\delta}_j\right\|_\infty + D(\boldsymbol{x}_i, \boldsymbol{x}_j) \\
= & D(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sup_{[\boldsymbol{x}_i, \boldsymbol{x}_i+\boldsymbol{\delta}_i]\times[\boldsymbol{x}_j, \boldsymbol{x}_j+\boldsymbol{\delta}_j]} \left\{ \left\| \frac{\partial D(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_i} \right\|_1, \left\| \frac{\partial D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_j} \right\|_1 \right\} \cdot \left\|\boldsymbol{\delta}_j\right\|_\infty \\
\leq & D(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sup_{[\boldsymbol{x}_i, \boldsymbol{x}_i+\boldsymbol{\delta}_i]\times[\boldsymbol{x}_j, \boldsymbol{x}_j+\boldsymbol{\delta}_j]} \epsilon \cdot \left\{ \left\| \frac{\partial D(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_i} \right\|_1, \left\| \frac{\partial D(\tilde{\boldsymbol{x}}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_j} \right\|_1 \right\},
\end{aligned} \tag{13}$$

where $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + \boldsymbol{\delta}_i$, $\|\boldsymbol{\delta}_i\|_\infty \leq \epsilon$, and $\|\boldsymbol{\delta}_j\|_\infty \leq \epsilon$. Next, we bound the L_1 norm of the gradient $\frac{\partial D(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_i}$. Based on the fact that $\boldsymbol{x}_i \in [0, 1]^d$, we have

$$\begin{aligned}
\epsilon \cdot \left\| \frac{\partial D(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_i} \right\|_1 &= \epsilon \cdot \left\| 2\boldsymbol{W}^T\boldsymbol{W}(\boldsymbol{x}_i - \boldsymbol{x}_j) \right\|_1 \leq \epsilon \cdot \max_{\boldsymbol{s}\in[-1,1]^d} \left\| 2\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{s} \right\|_1 \\
&= \epsilon \cdot \max_{\boldsymbol{s}\in[-1,1]^d, \boldsymbol{t}\in[-1,1]^d} 2\boldsymbol{t}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{s},
\end{aligned} \tag{14}$$

where the first equality follows the chain rule, the first inequality is derived based on the fact that $\boldsymbol{x}_i \in [0, 1]^d$, and the last equality follows from the identity $\|\boldsymbol{z}\|_1 = \max_{\boldsymbol{t}\in[-1,1]^d} \boldsymbol{t}^T\boldsymbol{z}$. In a similar

way, we can bound the term $\epsilon \cdot \|\frac{\partial D(\boldsymbol{x}_i+\boldsymbol{\delta}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}\|_1$, where $\boldsymbol{x}_i + \boldsymbol{\delta}_i \in [0,1]^d$. Substituting Equation (14) into Equation (13), we can get the following:

$$D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) \leq D(\boldsymbol{x}_i, \boldsymbol{x}_j) + \epsilon \cdot \max_{\boldsymbol{s}\in[-1,1]^d, \boldsymbol{t}\in[-1,1]^d} 2\boldsymbol{t}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{s}. \tag{15}$$

Since $\boldsymbol{W}^T\boldsymbol{W}$ is a positive semi-definite matrix, we have

$$(\boldsymbol{t}-\boldsymbol{s})^T\boldsymbol{W}^T\boldsymbol{W}(\boldsymbol{t}-\boldsymbol{s}) \geq 0, \tag{16}$$

which means that $\boldsymbol{t}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{t} + \boldsymbol{s}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{s} - 2\boldsymbol{t}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{s} \geq 0$. Then, we can further derive

$$\begin{aligned}\max_{\boldsymbol{s}\in[-1,1]^d, \boldsymbol{t}\in[-1,1]^d} 2\boldsymbol{t}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{s} &\leq \max_{\boldsymbol{t}\in[-1,1]^d} \boldsymbol{t}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{t} \\ + \max_{\boldsymbol{s}\in[-1,1]^d} \boldsymbol{s}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{s} &= \max_{\boldsymbol{t}\in[-1,1]^d} 2\boldsymbol{t}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{t}.\end{aligned} \tag{17}$$

Since the symmetric matrix $\boldsymbol{W}^T\boldsymbol{W}$ is semi-definite, the eigenvalues $\{\lambda_i \in \mathbb{R}\}_{i=1}^d$ of this matrix are real. Assume that those eigenvalues are ordered as $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_d$. Based on the spectral theorem, we can find the corresponding orthonormal eigenvectors $\{\boldsymbol{u}_i \in \mathbb{R}^d\}_{i=1}^d$ of these eigenvalues, which satisfy $\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{u}_i = \lambda_i\boldsymbol{u}_i$ and $\boldsymbol{W}^T\boldsymbol{W} = \sum_{i=1}^d \lambda_i\boldsymbol{u}_i\boldsymbol{u}_i^T$. For vector $\boldsymbol{t} \in [-1,1]^d$, it can be written as $\boldsymbol{t} = \sum_{i=1}^d c_i\boldsymbol{u}_i$ with length $||\boldsymbol{t}||_2^2 = \sum_{i=1}^d c_i^2$, where $\boldsymbol{u}_i$ is the found eigenvector. Based on the above decomposition, the quadratic form $\boldsymbol{t}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{t}$ can then be bounded as

$$\max_{\boldsymbol{t}\in[-1,1]^d} \boldsymbol{t}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{t} = \max_{\boldsymbol{t}\in[-1,1]^d} \sum_{i=1}^d \lambda_i c_i^2 \leq \max_{\boldsymbol{t}\in[-1,1]^d} \lambda_{max}^+ \sum_{i=1}^d c_i^2 = \max_{\boldsymbol{t}\in[-1,1]^d} \lambda_{max}^+ ||\boldsymbol{t}||_2^2, \tag{18}$$

where $\boldsymbol{t} = \sum_{i=1}^d c_i\boldsymbol{u}_i$ and $\lambda_{max}^+ = \lambda_d$ denotes the maximum eigenvalue. In Equation (18), the first equation is derived based on the fact that the quadratic form $\boldsymbol{t}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{t}$ can be rewritten as $\boldsymbol{t}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{t} = \sum_{i=1}^d \lambda_i c_i^2$, and the first inequality follows that λ_{max}^+ is the maximum eigenvalue, and hence, $\lambda_i c_i^2 \leq \lambda_{max}^+ c_i^2$. Since $\boldsymbol{t} \in [-1,1]^d$, we can derive $||\boldsymbol{t}||_2^2 \leq d$. Therefore, $D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j)$ can be further upper bounded as

$$D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) \leq D(\boldsymbol{x}_i, \boldsymbol{x}_j) + 2d\epsilon\lambda_{max}^+, \tag{19}$$

where $\lambda_{max}^+ = \lambda_d$. Finally, the pairwise adversarial loss $\mathcal{L}_A(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbb{I}[\Delta(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_i) > 0]$ for the attacker A can be upper bounded by $\mathbb{I}[(D(\boldsymbol{x}_i, \boldsymbol{x}_j) + 2d\epsilon\lambda_{max}^+ - \gamma) > 0]$.

4 DEFENSE AGAINST ADVERSARIAL PERTURBATIONS

In the above section, we provided the upper bound for adversarial attacks against the well-learned metric models. Note that in the previous sections, we mainly focus on the scenarios where the attacker aims at launching adversarial attacks to change a similar instance pair to a dissimilar pair via adversarial perturbations. However, in practice, the normal training process of existing metric learning approaches with their developed loss functions does not necessarily cause the derived attack upper bound to be small, which leaves much room for the attacker to change the prediction results. To address this challenge, we propose to incorporate the derived upper bound into these developed metric learning losses as a regularizer, and let this bound guide the choice of the robust distance metric. Without loss of generality, in the following, we take the widely adopted pairwise constrained metric learning loss as an example, and discuss how to use the proposed defense method to improve its robustness to adversarial perturbations.

Pairwise constrained metric learning loss function [6, 14, 58, 61, 72]. Suppose the associated class label for each instance $\boldsymbol{x}_i \in \boldsymbol{\mathcal{X}}$ is denoted as $y_i \in \{-1, 1\}$. For each pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, we can

derive a pairwise similarity label $y_{ij} = y_i y_j$ that denotes whether the two instances are similar (i.e., have the same class label) or not. If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar, y_{ij} is equal to 1, otherwise it is equal to -1. After constructing the similarity labels for all pairs, we can develop the following loss function of metric learning to learn the distance metric

$$\min_{\boldsymbol{W} \in \mathbb{R}^{d*d}} \mathcal{L}_1 = \frac{2}{N(N-1)} \sum_{i<j} \max\{0, 1 + y_{ij}(D(\boldsymbol{x}_i, \boldsymbol{x}_j) - \gamma)\}, \tag{20}$$

where γ is the unit margin. The above loss is developed based on the margin criterion. It guarantees that the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the new feature space is smaller than the pre-defined margin γ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar, and larger than γ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are dissimilar.

Defense mechanism. Here, we incorporate the derived attack upper bound into the training process of the above loss and design a novel defense mechanism. Note that the derived upper bound in Equation (19) only involves the largest eigenvalue $\lambda^+_{max} = \lambda_d$. According to Equation (18), the equality in Equation (19) holds iff $||\boldsymbol{t}||_2^2 = d$ and $\boldsymbol{t}/||\boldsymbol{t}||_2 = \boldsymbol{u}_d$, where $\boldsymbol{u}_d$ is the eigenvector corresponding to the largest eigenvalue λ_d. With the fact that $\boldsymbol{t} \in [-1, 1]^d$, we can know that $||\boldsymbol{t}||_2^2 = d$ holds iff all the elements in $\boldsymbol{t}$ are -1 or 1. However, in this case, the equality $\boldsymbol{t}/||\boldsymbol{t}||_2 = \boldsymbol{u}_d$ may not be satisfied, which means the derived upper bound in Equation (19) may not be the least upper bound (i.e., the supremum). Thus, directly using the upper bound in the training process cannot guarantee good defense performance. Considering that the supreme is affected by the top k maximum eigenvalues of $\boldsymbol{W}^T\boldsymbol{W}$, where the value of k is dependent on the concrete form of $\boldsymbol{W}^T\boldsymbol{W}$, in our proposed defense mechanism, we incorporate the top k eigenvalues into the developed loss function. Specifically, the final objective function for metric learning is formulated as

$$\min_{\boldsymbol{W} \in \mathbb{R}^{d*d}} \mathcal{L}_2 = \mathcal{L}_1 + \sum_{j=1}^{k} \alpha_j \lambda_{d-k+j}, \tag{21}$$

where the first term $\mathcal{L}_1$ is defined in Equation (20), and the second term is used to enhance the robustness of $\mathcal{L}_1$ to adversarial perturbations. In the second term, we introduce the top k eigenvalues $\{\lambda_{d-k+j}\}_{j=1}^k$, and k and α_j are tunable parameters.

Please note that the second term in Equation (21) can also be incorporated into the developed loss of other metric learning models to improve their robustness against adversarial perturbations. The key idea of the proposed defense mechanism is to reduce the attacker's exploration space by minimizing the attack upper bound.

5 DISCUSSION

In this section, we first describe how to generate adversarial similar sample pairs for truly dissimilar sample pairs. Then, we discuss the situation where the attacker aims at crafting adversarial perturbations to alter its classification result (i.e., the class label) based on our proposed AckMetric. Lastly, we give other forms of the upper bound of the pairwise adversarial loss by using other simple ways.

5.1 Adversarial Similar Instance Pairs

Note that in Section 2.3, we mainly focused on the scenarios where the attacker aims at changing a similar instance pair to a dissimilar pair via adversarial perturbations. Specifically, to achieve the attack goal, the attacker needs to solve the optimization problem in Equation (4) to find an optimal pair $(\boldsymbol{\delta}_i, \boldsymbol{\delta}_j)$ that can maximize the margin $\Delta(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j)$. In this section, we discuss the scenarios where the attacker aims at changing a dissimilar instance pair to a similar instance

pair via adversarial perturbations. As for this scenarios where the attacker aims at changing a dissimilar pair to a similar pair, this attack goal can be achieved by solving the following optimization problem

$$\min_{||\boldsymbol{\delta}_i||_\infty + ||\boldsymbol{\delta}_j||_\infty < \epsilon} \quad D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) - \gamma,$$
$$\text{s.t.} \quad \boldsymbol{x}_i + \boldsymbol{\delta}_i \in [0,1]^d,\ \boldsymbol{x}_j + \boldsymbol{\delta}_j \in [0,1]^d, \tag{22}$$

where ϵ controls the magnitude of adversarial pairwise perturbations. With the fact that maximizing an objective optimization over its argument is equivalent to minimizing that function over the same argument with a sign change, in these scenarios, we can easily conduct the analysis as before.

5.2 Adversarial Attacks Against Metric Learning-Based Classification

Note that in Sections 2.3 and 5.1, we discuss how to change a similar instance pair to a dissimilar pair and how to change a dissimilar instance pair to a similar pair, respectively. In practice, a common application of metric learning is the classification task where an incoming unlabeled test instance is classified by the majority label among its m-nearest (labeled) instances in the training set. In this section, we discuss how to attack a well-learned metric model in a classification task. More specifically, for a given test instance, we aim at crafting adversarial perturbations to alter its classification result (i.e., the class label) based on our proposed AckMetric.

By following most of existing metric learning works [5, 6, 48, 57, 60, 71, 72], in this article, we use the k-nearest neighbors algorithm as the classifier, based on which a given test instance is labeled by majority voting over its m nearest instances in the training set. Specifically, to determine the class label of a given test instance $\boldsymbol{x}_k$, we first need to calculate the distances between $\boldsymbol{x}_k$ and all the instances in the training set according to the learned distance metric. Then, we derive its corresponding ranked list of the training instances $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ sorted by the calculated distances. Here, for a given test instance $\boldsymbol{x}_k$, the goal of the attacker is to alter its original predicted class label via adversarial perturbations. We denote the ranked list of $\boldsymbol{x}_k$ as $\boldsymbol{x}_{\pi_k} = \{\boldsymbol{x}_{\pi_{k,1}}, \ldots, \boldsymbol{x}_{\pi_{k,r}}, \ldots, \boldsymbol{x}_{\pi_{k,N}}\}$, where $\pi_{k,1}$ denotes the index of the first closest training instance. Finally, the class label of $\boldsymbol{x}_k$ can be determined by majority voting over its top-m closest (nearest) training instances, i.e., $\boldsymbol{x}_{\pi_{k,1}}, \boldsymbol{x}_{\pi_{k,2}}, \ldots, \boldsymbol{x}_{\pi_{k,m}}$. To fool the learned metric model and alter the classification result of $\boldsymbol{x}_k$, we design the following attack strategy based on AckMetric: in the tth iteration, the attacker first finds $\boldsymbol{x}_k$'s M nearest training instances $\boldsymbol{x}_{\pi_k^t} = \{\boldsymbol{x}_{\pi_{k,1}^t}, \boldsymbol{x}_{\pi_{k,2}^t}, \ldots, \boldsymbol{x}_{\pi_{k,M}^t}\}$ by calculating its distance from each of the training instances, where $M > m$. Here, we use $\pi_{k,1}^t$ to denote the index of the first closest training instance derived in the tth iteration. Then, the distance between $\boldsymbol{x}_k$ and each of the selected M nearest instances $\{\boldsymbol{x}_{\pi_{k,1}^t}, \boldsymbol{x}_{\pi_{k,2}^t}, \ldots, \boldsymbol{x}_{\pi_{k,M}^t}\}$ is classified as positive if the selected training instance shares the same label with $\boldsymbol{x}_k$, otherwise the distance is classified as negative. Subsequently, we construct the following adversarial loss:

$$\max_{||\Delta_i||_\infty < \epsilon} \sum_{r=1}^{M} (-1)^{\mathbb{I}\{y_k == y_{k,r}^t\}+1} D\left(\boldsymbol{x}_k + \Delta_i, \boldsymbol{x}_{\pi_{k,r}^t}\right), \tag{23}$$

where y_k and $y_{k,r}^t$ denote the class labels of $\boldsymbol{x}_k$ and $\boldsymbol{x}_{\pi_{k,r}^t}$, respectively. $\mathbb{I}\{y_k == y_{k,r}^t\}$ is the indicator function that is equal to 1 if y_k and $y_{k,r}^t$ are equal, otherwise it is equal to 0. Finally, the targeted test instance is updated by AckMetric using the gradient of the constructed loss in Equation (23), which will increase the positive distances and decrease the negative distances. Based on this attack strategy, the attacker can generate adversarial instance $\hat{\boldsymbol{x}} = \boldsymbol{x}_k + \Delta_i$ to fool the learned metric model and alter the classification result.

5.3 Other Upper Bounds

In the above, we derive the upper bound of the pairwise adversarial loss by using convex relaxations. In fact, based on the spectral norm and the Frobenius norm, we can derive another two simple bounds (i.e. the spectral bound and the Frobenius bound) that can be used to upper bound the difference between $D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j)$ and $D(\boldsymbol{x}_i, \boldsymbol{x}_j)$. And the spectral bound and the Frobenius bound are respectively given as follows:

$$D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) \leq D(\boldsymbol{x}_i, \boldsymbol{x}_j) + 4d(\epsilon^2 + \epsilon)\|\boldsymbol{W}^T\boldsymbol{W}\|_2, \tag{24}$$

$$D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) \leq D(\boldsymbol{x}_i, \boldsymbol{x}_j) + 4d(\epsilon^2 + \epsilon)\|\boldsymbol{W}^T\boldsymbol{W}\|_F, \tag{25}$$

where $\|\boldsymbol{W}^T\boldsymbol{W}\|_2$ and $\|\boldsymbol{W}^T\boldsymbol{W}\|_F$ denote the spectral and Frobenius norm of $\boldsymbol{W}^T\boldsymbol{W}$, respectively. In the experiments, we empirically compare the proposed training objective (i.e., Equation (21)) with those using the above two simple bounds as regularization terms. The detailed derivations for the spectral bound and the Frobenius bound can be found in Section A of the supplementary Appendix.

6 EXPERIMENTS

In this section, we first evaluate the performance of the proposed attack method (i.e., AckMetric) in Section 6.1. Then, the effectiveness of the proposed defense mechanism is evaluated in Section 6.2.

6.1 The Performance of the Attack Method

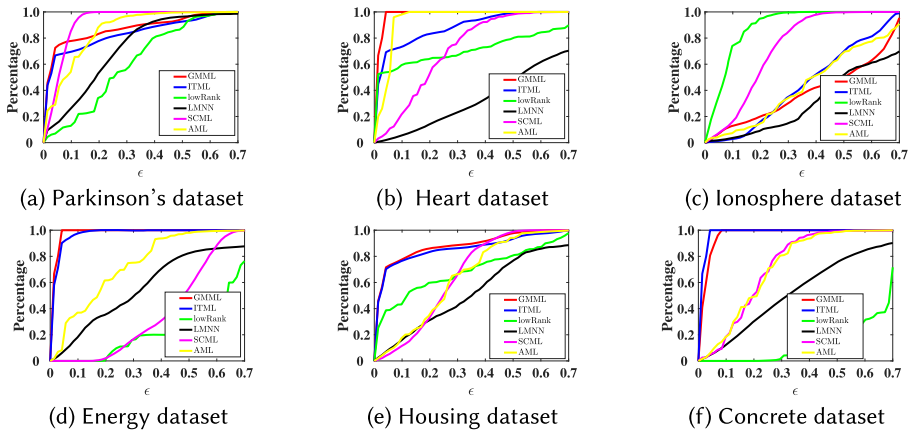
Metric learning models. In this experiment, we adopt the following state-of-the-art metric learning models to evaluate the performance of AckMetric. **LMNN** [57] is a method that aims at letting the k-nearest neighbors belong to the same class, while the instances from different classes are separated by a large margin. **GMML** [71] formulates the metric learning process as a smooth, strongly convex optimization problem by using pairs of similar and dissimilar points. **ITML** [6] models the metric learning problem in an information-theoretic setting by leveraging the relationship between the multivariate Gaussian distribution and the set of Mahalanobis distances. **LowRank** [72] presents a similarity algorithm by encoding low-rank structures to the learning process to conduct the sparse feature selection. **SCML** [48] aims at learning a sparse combination of locally discriminative metrics that are inexpensive to generate. **AML** [5] first generates synthetic hard samples based on GANs, and then uses these generated hard samples to boost the discriminability of the learned metric learning model.

Datasets. The details of the adopted real-world datasets are described as follows: The **Parkinson's disease dataset** [35] contains 22 features and 195 biomedical voice samples collected from 31 humans, in which 23 were diagnosed with Parkinson's Disease. The **Heart dataset** and the **Ionosphere dataset** are two binary classification datasets from UCI machine-learning repository.[2] The **MNIST 8v9 dataset** [33] is a subset of the 784-dimensional MNIST set, and it contains 2,016 images. The **AT&T face recognition dataset**[3] constains 400 grayscale images of 40 individuals in 10 different poses. The task of this face dataset is to determine whether two face images are from the same identity or not. Additionally, we also adopt three UCI regression datasets (i.e., **Energy**, **Housing,** and **Concrete**). For each of them, we first normalize the real-valued output of each instance to [0,1], and then label the top 30% of the instances with the positive category and the remaining instances with negative category. The statistical information of the adopted datasets are described in Table 1.

[2] https://archive.ics.uci.edu/ml/datasets.html.
[3] https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

Table 1. The Statistics of the Adopted Datasets

Dataset	Size	Dimension	Classes
Parkinson	195	22	2
Heart	303	23	2
Ionosphere	351	34	2
AT&T	400	92×112	40
8v9	2,016	28×28	2
Energy	768	8	2
Housing	506	13	2
Concrete	1,030	8	2



(a) Parkinson's dataset (b) Heart dataset (c) Ionosphere dataset

(d) Energy dataset (e) Housing dataset (f) Concrete dataset

Fig. 1. The percentage of the successfully generated adversarial pairs.

Performance. We evaluate the performance of AckMetric through measuring the percentage of the successfully generated adversarial pairs that can fool the target metric model on the above eight real-world datasets. For each adopted dataset, we first randomly select a subset of instances as the training set, and then randomly sample the testing instance pairs from the remaining instances. The number of training instances and testing instance pairs for each dataset is provided in Table 4 (in the supplementary Appendix). In this experiment, the parameters of each adopted metric learning model are the same as that in its original work. Figure 1 reports the results for the six models on the Parkinson's disease dataset, the Heart dataset, the Ionosphere dataset, the Energy dataset, the Housing dataset, and the Concrete dataset. Here, we vary ϵ from 0 to 0.7. From this figure, we can see that the adopted models are vulnerable to adversarial perturbations, and the proposed AckMetric can easily fool the six models. For example, when the parameter ϵ is set as 0.6, the attacker is able to craft adversarial pairs against the adopted metric models with almost 100% success on the Parkinson's disease dataset. As for the two image datasets (i.e., 8v9 and AT&T), we vary ϵ from 0.05 to 0.15 and report the results for the models GMML, LMNN, and SCML in Table 2, from which we can see AckMetric still has good performance. By setting ϵ as 0.15, the attacker can successfully generate adversarial pairs on 63% of the MNIST 8V9 testing data when the model is SCML, and let GMML misclassify 54% of the AT&T testing data. All these results show that the learned models using given metric learning methods are vulnerable to adversarial perturbations and the proposed AckMetric can effectively generate adversarial pairs to fool well-learned metric models.

Table 2. The Percentage of the Successfully Generated Adversarial Pairs on the Image Datasets

		ϵ				
Datasets	**Methods**	0.05	0.075	0.1	0.125	0.15
8V9	GMML	0.38	0.39	0.39	0.44	0.44
	LMNN	0.40	0.41	0.43	0.45	0.47
	SCML	0.05	0.14	0.28	0.47	0.63
AT&T	GMML	0.23	0.32	0.33	0.36	0.38
	LMNN	0.25	0.27	0.33	0.36	0.36
	SCML	0.25	0.32	0.43	0.47	0.54

(a) (b) (c) (d) (e)

Fig. 2. Adversarial dissimilar image pairs crafted by AckMetric when LMNN is adopted on the MNIST 8V9 dataset.

Inspection of adversarial dissimilar pairs. Here, we provide the visualization results for the crafted adversarial dissimilar pairs that are generated by the proposed AckMetric. Specifically, for each original (clean) similar instance pair, we aim at generating the adversarial perturbations that is added to the original similar instance pair cause a metric learning model to make a false similarity prediction (i.e., the adversarial dissimilar pair). In this experiment, we conduct experiments on the two adopted image datasets (i.e., the MNIST and AT&T datasets), and the parameter ϵ is set as 0.01. The reason is that by setting $\epsilon = 0.01$, we can ensure that the pairwise adversarial perturbations are imperceptible to humans. In this way, the added slight adversarial perturbations could not be recognized by human eyes, and the attacker can avoid being detected. Meanwhile, the added adversarial pairwise perturbations can construct adversarial instance pairs that largely change the pairwise prediction results given by the metric learning models. Then, we can evaluate whether the proposed AckMetric can easily generate adversarial instance pairs with imperceptible changes to fool these metric learning models. Figure 2 shows some examples of the adversarial dissimilar pairs generated by AckMetric when LMNN is adopted on the MNIST 8V9 dataset. For simplicity, we take Figure 2(a) as an example to give an intuitive understanding of the generated dissimilar pairs. In the bottom row of Figure 2(a), we show the original similar image pair of two digit images, which is also treated as a similar pair by LMNN when there is no adversarial perturbations. In the middle row of Figure 2(a), we show the adversarial perturbations that is added to the original similar pair (in the bottom row of Figure 2(a)) to craft the adversarial instance pair to fool the trained metric learning model. In the top row of Figure 2(a), we show the generated adversarial image pair that is crafted by adding the adversarial perturbations in the middle row to the original similar image pair in the bottom row. From this figure, we can observe that the adversarial dissimilar image pair in the top row is almost the same as the original similar image pair in the bottom row and the added adversarial perturbations are imperceptible to humans, but the crafted adversarial image pair can successfully mislead LMNN to make wrong similarity predictions. We also visualize the adversarial

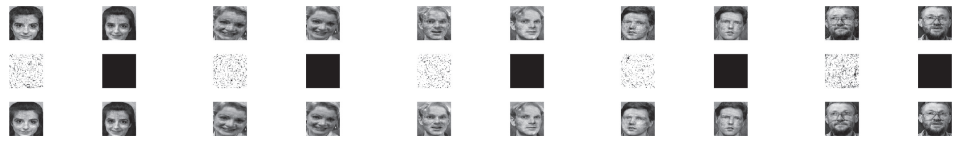


Fig. 3. Adversarial dissimilar image pairs crafted by AckMetric when SCML is adopted on the AT&T dataset.

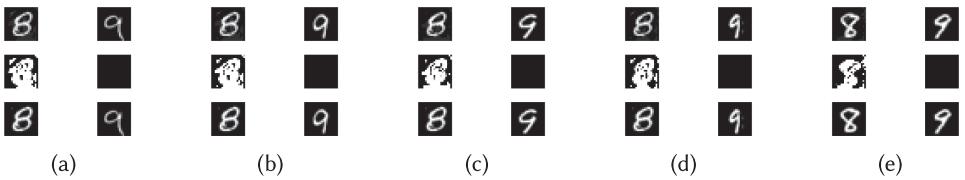
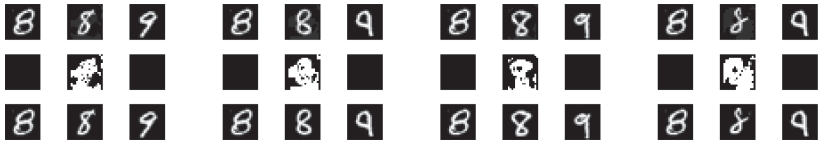


Fig. 4. Adversarial similar image pairs crafted by AckMetric when LMNN is adopted on the MNIST 8V9 dataset.

dissimilar pairs crafted by AckMetric when the SCML model is adopted on the AT&T dataset. Some examples are provided in Figure 3, from which we can also observe the similar observation. The reported visualization results in the Figures 2 and 3 also verify that current metric learning models are not robust enough to adversarial perturbations and the proposed AckMetric can easily generate adversarial pairs with imperceptible changes to fool these metric learning models.

Inspection of adversarial similar pairs. In addition to providing the visualization results of the crafted dissimilar pairs, we also visualize the adversarial similar image pairs crafted by AckMetric (with the optimization framework described in Section 5.1) when LMNN is adopted on the MNIST 8V9 dataset. Here, ϵ is still set as 0.01. Figure 4 shows the crafted adversarial similar instance pairs. For simplicity, we take Figure 4(a) as an example to give an intuitive understanding of the generated similar instance pairs. In Figure 4(a), the image pair in the bottom row is the original clean dissimilar pair, which are also correctly classified by LMNN as dissimilar. The image pair in the top row of Figure 4(a) is the generated adversarial similar pair, which is crafted by the proposed AckMetric and misclassified by LMNN as similar. The generated adversarial perturbations added to the original dissimilar pair (in the bottom row of Figure 4(a)) to generate the adversarial similar pair is shown in the middle row of Figure 4(a). From this figure, we can observe that the crafted adversarial similar pair (in the top row of Figure 4(a)) is visually indistinguishable from the original clean dissimilar pair (in the bottom row of Figure 4(a)). Importantly, the crafted adversarial similar pair (in the top row of Figure 4(a)) can successfully fool LMNN, which further verifies the effectiveness of the proposed AckMetric and that current metric learning models are not robust enough to adversarial perturbations.

Crafting adversarial triplets. In addition to generating adversarial pairs, we also conduct experiments to evaluate the robustness of current metric learning models by crafting adversarial triplets based on the proposed AckMetric. The details about how to generate adversarial triplets are described in Section C of the supplementary Appendix. Figure 5 reports some examples of the adversarial triplets crafted by AckMetric when LMNN is adopted on the MNIST 8V9 dataset. In this experiment, we randomly select half of the instances in the dataset to train LMNN, and the parameter ϵ is set as 0.01. Take Figure 5(a) as an example. The image triplet in the bottom row denotes the original triplet. For this original triplet, based on LMNN, we can derive that the left image is more similar to the middle image than to the right image. The triplet in the top row is the



(a) (b) (c) (d)

Fig. 5. Adversarial triplets crafted by AckMetric when LMNN is adopted on the MNIST 8V9 dataset.

crafted adversarial triplet, which has the reverse relationship based on LMNN (i.e., the left image is more similar to the right image than to the middle image). The perturbations added to the original triplet to generate the adversarial triplet are shown in the middle row. As we can see, the crafted adversarial image triplets are almost the same as the original image triplet and the changes are imperceptible to humans, but the crafted adversarial triplets can successfully fool LMNN.

6.2 Evaluating the Proposed Training Objective

In this section, we evaluate the effectiveness of the proposed defense mechanism. Here, we still take the widely adopted pairwise constrained metric learning loss (i.e., Equation (20)) as an example and evaluate whether the proposed training objective (i.e., Equation (21)) can improve its robustness to adversarial perturbations.

Baselines. Note that the proposed training objective (i.e., Equation (21)) is derived by adding the upper bound to the pairwise constrained loss function (i.e., Equation (20)). In experiments, we compare the proposed training objective with the following three different objectives:

- **Normal training (NT-ML).** The pairwise constrained loss $\mathcal{L}_1$ and no explicit regularization.
- **Spectral norm regularization (Spe-ML).** The pairwise constrained loss $\mathcal{L}_1$ and the regularizer $\beta_1\|\boldsymbol{W}^T\boldsymbol{W}\|_2$ with $\beta_1 = 0.2$ (i.e., $\mathcal{L}_1 + \beta_1\|\boldsymbol{W}^T\boldsymbol{W}\|_2$).
- **Frobenius norm regularization (Fro-ML).** The pairwise constrained loss $\mathcal{L}_1$ and the regularizer $\beta_2\|\boldsymbol{W}^T\boldsymbol{W}\|_F$ with $\beta_2 = 0.2$ (i.e., $\mathcal{L}_1 + \beta_2\|\boldsymbol{W}^T\boldsymbol{W}\|_F$).

The robustness of the proposed training objective. In this experiment, we evaluate the effectiveness of the proposed training objective in the metric learning based classification task. Specifically, for each training objective, we first use the training data to learn a distance metric. Then, we generate the adversarial instance for each of the instances in the test set based on the attack strategy described in Section 5.2. Finally, we calculate the classification accuracy of the adversarial instances. Here, the class labels of the adversarial instances are assigned based on the KNN classifier. The higher the classification accuracy, the more robust the metric learning model, which means the corresponding defense objective is more effective.

The parameter setting is detailed in Table 5 (in the supplementary Appendix). We tune the number of the eigenvalues used in the certificated loss (i.e., Equation (21)), and Table 3 shows the experimental results on the Parkinson's disease, Ionosphere, and Heart datasets. From this table, we can see that the metric learning model (i.e., NT-ML) without any defense mechanism achieves the worst performance under attack. The testing accuracy of NT-ML on the Ionosphere dataset is only 0.26 under attack, which further verifies the vulnerability of the learned metric models. The results also show that the performance of the learned metric models under attack can be improved after considering the derived upper bounds. Although the proposed training objective (i.e., Equation (21)) performs slightly worse than Spe-ML when there is no attack, it can achieve much better performance under attack. When we incorporate the top three eigenvalues, the testing accuracy of the proposed training objective on the Heart dataset is 0.64, while those of Spe-ML and

Table 3. Classification Accuracy: Testing Accuracy Without Attack/ Testing Accuracy Under Attack

	Parkinson	Ionosphere	Heart
NT-ML	0.93/0.37	0.88/0.26	0.77/0.54
Spe-ML	**0.97**/0.54	**0.89**/0.64	**0.83**/0.57
Fro-ML	0.85/0.42	0.86/0.70	0.69/0.52
1 **eigenvalue**	0.89/0.51	0.82/0.71	0.79/0.59
3 **eigenvalues**	0.93/**0.56**	0.88/**0.73**	0.77/**0.64**
5 **eigenvalues**	0.95/0.54	0.88/0.71	0.80/0.62

Fro-ML are only 0.57 and 0.52, respectively. These results demonstrate that the proposed training objective (i.e., Equation (21)) can make metric learning more robust against adversarial perturbations.

7 RELATED WORK

Existing metric learning models can be divided into two categories: linear and nonlinear. The linear models [5, 6, 18, 19, 21, 22, 26, 45, 53, 57, 60, 62, 63, 67, 70, 71] are constructed to learn a linear mapping to project the original instances into a new feature space, while the nonlinear models (i.e., deep metric learning models) [7, 8, 16, 23, 24, 28, 29, 32, 34, 55, 56, 68, 73] usually adopt neural networks to capture the nonlinear structures of the instances. In both cases, the similarity degrees of instances can be determined in the newly learned feature space. For example, [28] presents a new loss and tuple mining strategy for deep metric learning using continuous labels. Reference [32] proposes a new loss function (i.e., Group Loss) for deep metric learning that considers the similarity between all samples in a mini-batch. Reference [56] designs a new ranking-motivated structured loss for deep metric learning to learn discriminative embeddings with the setting of few-shot retrieval. Reference [34] proposes a deep variational metric learning framework to explicitly model the intra-class variance and disentangle the intra-class invariance, namely, the class centers. Reference [24] introduces a **Position-Dependent Deep Metric** (**PDDM**) unit, which is capable of learning a similarity metric adaptive to local feature structure and the learned metric can be used to select genuinely hard samples in a local neighborhood to guide the deep embedding learning in an online and robust manner. Reference [53] proposes a new angular loss to augment conventional distance metric learning by encoding the third-order relation inside triplet in terms of the angle at the negative point. Motivated by that the linear metric models often fail to produce reliable distances on the ambiguous test pairs due to the different samplings between training set and test set, the authors in [5] discuss how to generate adversarial pairs in the linear case to remedy the sampling bias and facilitate robust metric learning. The recent work [7] considers the nonlinear models (i.e., deep metric learning models), and proposes a new deep metric learning framework (by using generative adversarial networks) to generate synthetic hard negative samples to train deep metric learning models. However, this article fails to consider the adversarial side of the proposed methods and does not study the robustness of the learned deep metric models to adversarial perturbations. Additionally, the authors in [39] use metric learning as a tool to produce robust classifiers. Specifically, based on the observation that adversarial attacks can cause the internal deep representation to shift closer to the "false" class, they propose to leverage the triplet loss of metric learning to bring near both the natural and adversarial samples of the same class. However, in our article, we study the robustness of metric learning itself. Thus, their problem setting is totally different from ours.

This work is inspired by the recent developments of adversarial deep learning works. Among these works, FGSM [11], PGD and its distributional variant [31], and CW [4] are the three most

prevalent attacks. Simultaneously, there also exists many defenses against adversarial samples. Adversarial training is currently the most prevalent defense method [3, 38, 50]. However, a weakness of adversarial training is that its defensive effectiveness is not theoretically guaranteed. Thus, some researchers then try to develop provable defenses [46], where a certain prediction accuracy can be guaranteed as long as adversarial perturbations are bounded. However, it is specially designed for deep learning models, and cannot be applied to metric learning due to the symmetric positive semi-definite property of the well-trained metric. Specifically, for the metric learning models studied in this article, the matrix $\boldsymbol{W}^T\boldsymbol{W}$ in Equation (17) is a symmetric positive semi-definite matrix, whose eigenvalues are real and non-negative. This property leads to the unique certificate and objective for metric learning derived in Equations (19) and (21). While for [46], the matrix $\boldsymbol{W}^T diag\{v\}$ in its Equation (5) is not necessarily semi-definite, which motivates [46] to use the semi-definite programming relaxation for MAXCUT to approximate an upper bound for the optimization problem.

There are some existing adversarial works [74, 75] that aim at training robust deep neural networks. For example, [75] aims at synthesizing hard triplets to achieve faster convergence rates and improve the global structure of the embedding space. The synthesized hard triplets contain harder examples that cannot be well handled by the current embedding network where the irrelevant example is closer to the query than the relevant counterpart. Specifically, given that existing works fail to generate hard triplets that really matter in globally optimizing the network, [75] proposes an adversarial learning algorithm, in which a hard triplet generator and an embedding network are jointly optimized in an adversarial fashion to mutually benefit each other. The authors in [74] present a regularization mechanism for training deep neural networks by minimizing the **worse-case perturbation** (**WCP**). In other words, [74] encourages the model to avoid putting its decision boundary through the dense areas of data points such that the perturbations are least likely to incur a large change to the outputs of the model. Specifically, [74] considers two forms of WCP regularizations' additive and DropConnect perturbations, which imposes additive noises on network weights and make structural changes by dropping the network connections, respectively. And the network is trained by minimizing the change of model predictions subject to these perturbations. However, the problem settings of these works are significantly different from ours, and they cannot be directly adopted in our setting. In our work, we aim at exploring the capability of the attacker to change the prediction results of the metric learning models. To achieve this goal, we first design a novel attack method to show that existing metric learning models are vulnerable to adversarial perturbations, and then present an upper bound for the pairwise adversarial loss, which serves as a certificate for the pairwise adversarial loss and is incorporated into the developed loss of metric learning as a regularizer to enhance the robustness of metric learning against adversarial attacks. However, [74, 75] only focus on designing robust training mechanisms for training robust deep neural networks (instead of metric learning models).

Migrating learning can migrate knowledge from the source domain to the target domain to help the learning tasks in new environments. Metric migration learning aims at mitigating the insufficient label information issue for distance metric learning in the domain of interest (target domain) by leveraging knowledge/information from other related domains (source domains) [37]. Metric migration learning is able to find data embeddings that perform well on a testing domain, called a target domain, by using labeled and/or unlabeled data in source domains [37]. Currently, many metric migration learning works have been proposed [1, 15, 25, 30, 42, 47, 64–66]. For example, [15] proposes a new deep transfer metric learning method to learn a set of hierarchical nonlinear transformations for cross-domain visual recognition by transferring discriminative knowledge from the labeled source domain to the unlabeled target domain. Given that samples in the source domain might be extracted into different groups and the samples in the same group would have similar intrinsic attributes, [64] proposes a metric transfer learning framework to encode metric learning

in transfer learning. Reference [30] proposes a transfer metric learning method to infer domain-specifc data embeddings for unseen domains, from which no data are given in the training phase, by using knowledge transferred from related domains. Reference [37] groups metric transfer learning into different categories according to different settings and metric transfer strategies, such as direct metric approximation, subspace approximation, distance approximation, and distribution approximation. However, the problem settings in above works are significantly different from that in our work. In addition, they also fail to study the model robustness to adversarial perturbations.

Confrontation training [12, 27, 36, 52, 54, 69] is the process of training a model to correctly classify both unmodified examples and adversarial examples. As an unsupervised learning model with strong scalability, **Generative Adversarial Networks (GANs)** provide a confrontation training method for deep networks, which solves tough issues for classical training methods [36]. In practice, the whole training process of confrontation learning is usually divided into the following steps [69]: First, the generator network is pre-trained to capture the probability distribution of the real data in the training set and transform the input random perturbation into new samples. Then, the discriminator network observes both real and fake data to determine the authenticity of this data. Then, the two networks alternately confront each other until convergence. The authors in [12] propose LeakGAN to leak the features extracted by the discriminator to the generator in the process of confrontation learning; thus, helping the generator obtain more useful information to improve the quality of the generated text. Reference [52] proposes a **principal component analysis optimized generative adversarial networks (PCA-GAN)**. In the proposed method, the original data are compressed to generate the input of the confrontation network, so that the input data retain the characteristics of the original data to some extent, thereby improving the data generation performance and reducing the training time cost. Reference [54] uses GANs to extract the hidden features of fusion information objectively and effectively in the way of confrontation learning. Reference [27] adopts an adversarial risk analysis perspective to model the confrontation between attackers and defenders mitigating questionable common knowledge assumptions. However, the above mentioned methods require making small perturbations to numerous entries of the input vector, which is inappropriate for sparse high-dimensional inputs. Most importantly, they cannot provide the theoretical guarantee.

8 CONCLUSION

In this article, for the first time, we studied the robustness of metric learning to adversarial perturbations. Specifically, we first proposed a novel projected gradient descent-based attack method (i.e., AckMetric) to show that current metric learning models are vulnerable to adversarial perturbations. To further explore the capability of the attacker to affect the prediction results of a learned metric model, we also derived an upper bound for the pairwise adversarial loss, which serves as a certificate that for a given metric learning model and test input, no attack can force the introduced error to exceed a certain value. Moreover, we proposed to incorporate the derived upper bound into the developed loss of metric learning as a regularizer to enhance the robustness of metric learning against adversarial attacks. The experimental results not only show that current metric learning models are vulnerable to adversarial perturbations, but also demonstrate the effectiveness of the proposed defense mechanism (i.e., the proposed training objective).

APPENDICES

A THE DERIVATIONS OF OTHER UPPER BOUNDS

In Section 5.3, based on the spectral norm and the Frobenius norm, we propose another two simple upper bounds (i.e., the spectral bound and the Frobenius bound). In the following, we give detailed derivations for the proposed two simple upper bounds.

Table 4. The Number of Training Instances and Testing Instance Pairs

Dataset	#training instances	#testing instance pairs
Parkinson	97	4,753
Heart	151	11,476
Ionosphere	175	15,753
Energy	384	6,555
Housing	253	2,775
A&T	240	780
Concrete	515	11,781
8v9	1,008	20,100

Spectral bound: Firstly, we rewrite the difference between $D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j)$ and $D(\boldsymbol{x}_i, \boldsymbol{x}_j)$ as follows:

$$\begin{aligned} D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) - D(\boldsymbol{x}_i, \boldsymbol{x}_j) &= (\boldsymbol{x}_i + \boldsymbol{\delta}_i - \boldsymbol{x}_j - \boldsymbol{\delta}_j)^T \boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{x}_i + \boldsymbol{\delta}_i - \boldsymbol{x}_j - \boldsymbol{\delta}_j) \\ &\quad - (\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{x}_i - \boldsymbol{x}_j) \\ &= -(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{x}_i - \boldsymbol{x}_j) + ((\boldsymbol{x}_i - \boldsymbol{x}_j) + (\boldsymbol{\delta}_i - \boldsymbol{\delta}_j))^T \boldsymbol{W}^T \boldsymbol{W} ((\boldsymbol{x}_i - \boldsymbol{x}_j) + (\boldsymbol{\delta}_i - \boldsymbol{\delta}_j)) \\ &= 2(\boldsymbol{\delta}_i - \boldsymbol{\delta}_j)^T \boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{x}_i - \boldsymbol{x}_j) + (\boldsymbol{\delta}_i - \boldsymbol{\delta}_j)^T \boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{\delta}_i - \boldsymbol{\delta}_j). \end{aligned} \tag{26}$$

Then, based on the Cauchy–Schwarz inequality, we can further derive that

$$\begin{aligned} (\boldsymbol{\delta}_i - \boldsymbol{\delta}_j)^T \boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{x}_i - \boldsymbol{x}_j) &\le \|\boldsymbol{\delta}_i - \boldsymbol{\delta}_j\|_2 \|\boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{x}_i - \boldsymbol{x}_j)\|_2 \\ &\le \|\boldsymbol{\delta}_i - \boldsymbol{\delta}_j\|_2 \|\boldsymbol{W}^T \boldsymbol{W}\|_2 \sqrt{d}, \end{aligned} \tag{27}$$

where $\boldsymbol{x}_i \in [0, 1]$, and $\|\boldsymbol{W}^T \boldsymbol{W}\|_2$ is the spectral norm of $\boldsymbol{W}^T \boldsymbol{W}$. Since the L_2-norm satisfies the triangle inequality, we also have that $\|\boldsymbol{\delta}_i - \boldsymbol{\delta}_j\|_2 \le \|\boldsymbol{\delta}_i\|_2 + \|\boldsymbol{\delta}_j\|_2 \le 2\epsilon\sqrt{d}$. Consequently, we can derive that

$$2(\boldsymbol{\delta}_i - \boldsymbol{\delta}_j)^T \boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{x}_i - \boldsymbol{x}_j) \le 4d\epsilon \|\boldsymbol{W}^T \boldsymbol{W}\|_2. \tag{28}$$

Similarly, the term $(\boldsymbol{\delta}_i - \boldsymbol{\delta}_j)^T \boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{\delta}_i - \boldsymbol{\delta}_j)$ can be upper bounded as follows:

$$\begin{aligned} &(\boldsymbol{\delta}_i - \boldsymbol{\delta}_j)^T \boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{\delta}_i - \boldsymbol{\delta}_j) \\ &\le \|\boldsymbol{\delta}_i - \boldsymbol{\delta}_j\|_2 \|\boldsymbol{W}^T \boldsymbol{W}\|_2 \|\boldsymbol{\delta}_i - \boldsymbol{\delta}_j\|_2 \le 4d\epsilon^2 \|\boldsymbol{W}^T \boldsymbol{W}\|_2. \end{aligned} \tag{29}$$

Combining all of the above results (i.e., Equations (26), (28), and (29)), we obtain the following upper bound:

$$D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) - D(\boldsymbol{x}_i, \boldsymbol{x}_j) \le 4d(\epsilon^2 + \epsilon) \|\boldsymbol{W}^T \boldsymbol{W}\|_2.$$

The above measure of vulnerability to adversarial pairs is based on the spectral norms of $\boldsymbol{W}^T \boldsymbol{W}$.

Frobenius bound: To make the training easier, the Frobenius norm is usually regularized instead of the spectral norm [17]. Since $\|\boldsymbol{W}^T \boldsymbol{W}\|_2 \le \|\boldsymbol{W}^T \boldsymbol{W}\|_F$, we can derive another upper bound

$$D(\boldsymbol{x}_i + \boldsymbol{\delta}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) - D(\boldsymbol{x}_i, \boldsymbol{x}_j) \le 4d(\epsilon^2 + \epsilon) \|\boldsymbol{W}^T \boldsymbol{W}\|_F.$$

B THE NUMBER OF TRAINING INSTANCES AND TESTING INSTANCE PAIRS

For each of the adopted real-world datasets that are used to evaluate the performance of the proposed AckMetric (Section 6.1), we provide the details about the number of training instances and testing instance pairs in Table 4.

Table 5. The Setting of Parameters

Training parameters	learning rate	0.01
	iterations	500
	m	3
	α_j in certificated loss	$1.0/k$ $(j = 1..k)$
Attack parameters	step size	0.01
	iterations	20
	M	10
	maximum perturbation	0.1 (normalized)

C CRAFTING ADVERSARIAL TRIPLETS

In the definition of the pairwise Robustness for Metric Learning (i.e., Definition 2.1), there is a threshold γ that is used to judge whether the attacker can successfully craft an adversarial pair. Based on this definition, we can also craft adversarial triplets by assigning a proper value to the parameter γ. For a given triplet $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$ where $\boldsymbol{x}_i$ is more similar to $\boldsymbol{x}_j$ than to $\boldsymbol{x}_k$ (i.e., $D(\boldsymbol{x}_i, \boldsymbol{x}_j) < D(\boldsymbol{x}_i, \boldsymbol{x}_k)$), the attacker can craft an adversarial triplet by the following steps: Firstly, the attacker sets the value of γ as $D(\boldsymbol{x}_i, \boldsymbol{x}_k)$ (i.e., $\gamma = D(\boldsymbol{x}_i, \boldsymbol{x}_k)$. Then, the attacker perturbs $\boldsymbol{x}_j$ by adding an optimal noise $\boldsymbol{\delta}_j$, which satisfies $||\boldsymbol{\delta}_j||_\infty < \epsilon$ and $D(\boldsymbol{x}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j) > \gamma$. Compared with the original triplet $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$, the crafted adversarial triplet $(\boldsymbol{x}_i, \boldsymbol{x}_j + \boldsymbol{\delta}_j, \boldsymbol{x}_k)$ has the reverse relative comparison relationship, i.e., the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j + \boldsymbol{\delta}_j$ is larger that between $\boldsymbol{x}_i$ and $\boldsymbol{x}_k$.

D PARAMETER SETTING FOR THE EVALUATION OF THE TRAINING OBJECTIVE

Table 5 shows the setting the parameters when evaluating the effectiveness of the proposed training objective in Section 6.2.

REFERENCES

[1] Mahya Ahmadvand and Jafar Tahmoresnezhad. 2020. Metric transfer learning via geometric knowledge embedding. *Applied Intelligence* 51, 2 (2020), 921–934.

[2] Osbert Bastani, Yani Ioannou, Leonidas Lampropoulos, Dimitrios Vytiniotis, Aditya Nori, and Antonio Criminisi. 2016. Measuring neural net robustness with constraints. In *Proceedings of the NeurIPS*.

[3] Qi-Zhi Cai, Min Du, Chang Liu, and Dawn Song. 2018. Curriculum adversarial training. arXiv:1805.04807. Retrieved from https://arxiv.org/abs/1805.04807.

[4] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[5] Shuo Chen, Chen Gong, Jian Yang, Xiang Li, Yang Wei, and Jun Li. 2018. Adversarial metric learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

[6] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. 2007. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*. ACM, 209–216.

[7] Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. 2018. Deep adversarial metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2780–2789.

[8] Ismail Elezi, Sebastiano Vascon, Alessandro Torcinovich, Marcello Pelillo, and Laura Leal-Taixe. 2020. The group loss for deep metric learning. In *Proceedings of the European Conference on Computer Vision*. Springer, 277–294.

[9] Xingyu Gao, Steven CH Hoi, Yongdong Zhang, Ji Wan, and Jintao Li. 2014. Soml: Sparse online metric learning with application to image retrieval. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. 1206–1212.

[10] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. arXiv:1412.6572. Retrieved from https://arxiv.org/abs/1412.6572.

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. arXiv:1412.6572. Retrieved from https://arxiv.org/abs/1412.6572.

[12] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[13] Jamie Hayes and George Danezis. 2017. Machine learning as an adversarial service: Learning black-box adversarial examples. arXiv preprint arXiv:1708.05207.
[14] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. 2014. Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1875–1882.
[15] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. 2015. Deep transfer metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 325–333.
[16] Juhua Hu, De-Chuan Zhan, Xintao Wu, Yuan Jiang, and Zhi-Hua Zhou. 2015. Pairwised specific distance learning from physical linkages. *ACM Transactions on Knowledge Discovery from Data* 9, 3 (2015), 1–27.
[17] Mengdi Huai, Chenglin Miao, Yaliang Li, Qiuling Suo, Lu Su, and Aidong Zhang. 2018. Metric learning from probabilistic labels. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
[18] Mengdi Huai, Chenglin Miao, Yaliang Li, Qiuling Suo, Lu Su, and Aidong Zhang. 2020. Learning distance metrics from probabilistic information. *ACM Transactions on Knowledge Discovery from Data* 14, 5 (2020), 1–33.
[19] Mengdi Huai, Chenglin Miao, Jinduo Liu, Di Wang, Jingyuan Chou, and Aidong Zhang. 2020. Global interpretation for patient similarity learning. In *Proceedings of the 2020 IEEE International Conference on Bioinformatics and Biomedicine*. IEEE, 589–594.
[20] Mengdi Huai, Jianhui Sun, Renqin Cai, Liuyi Yao, and Aidong Zhang. 2020. Malicious attacks against deep reinforcement learning interpretations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 472–482.
[21] Mengdi Huai, Di Wang, Chenglin Miao, Jinhui Xu, and Aidong Zhang. 2020. Pairwise learning with differential privacy guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 694–701.
[22] Mengdi Huai, Di Wang, Chenglin Miao, and Aidong Zhang. 2020. Towards interpretation of pairwise learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4166–4173.
[23] Mengdi Huai, Hongfei Xue, Chenglin Miao, Liuyi Yao, Lu Su, Changyou Chen, and Aidong Zhang. 2019. Deep metric learning: The generalization analysis and an adaptive algorithm.. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 2535–2541.
[24] Chen Huang, Chen Change Loy, and Xiaoou Tang. 2016. Local similarity-aware deep feature embedding. In *Proceedings of the Neural Information Processing Systems*.
[25] Junchu Huang and Zhiheng Zhou. 2019. Transfer metric learning for unsupervised domain adaptation. *IET Image Processing* 13, 5 (2019), 804–810.
[26] Sho Inaba, Carl T. Fakhry, Rahul V. Kulkarni, and Kourosh Zarringhalam. 2019. A free energy based approach for distance metric learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 5–13.
[27] David Rios Insua, Roi Naveiro, Victor Gallego, and Jason Poulos. 2020. Adversarial machine learning: Perspectives from adversarial risk analysis. arXiv:2003.03546. Retrieved from https://arxiv.org/abs/2003.03546.
[28] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. 2019. Deep metric learning beyond binary supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2288–2297.
[29] Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. 2018. Attention-based ensemble for deep metric learning. In *Proceedings of the European Conference on Computer Vision*. 736–751.
[30] Atsutoshi Kumagai, Tomoharu Iwata, and Yasuhiro Fujiwara. 2020. Transfer metric learning for unseen domains. *Data Science and Engineering* 5 (2020), 140–151.
[31] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial machine learning at scale. In *Proceedings of the International Conference on Learning Representations*.
[32] Marc T. Law, Raquel Urtasun, and Richard S. Zemel. 2017. Deep spectral clustering learning. In *Proceedings of the International Conference on Machine Learning*.
[33] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*.
[34] Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. 2018. Deep variational metric learning. In *Proceedings of the European Conference on Computer Vision*. 689–704.
[35] Max A. Little, Patrick E. McSharry, Eric J Hunter, Jennifer Spielman, and Lorraine O Ramig. 2008. Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. *Nature Precedings* (2008), 1–1.
[36] Xinyue Liu, Wenbo Tian, Wenxin Liang, and Hua Shen. 2019. Goal-directed sequence generation with simulation feedback method. In *Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference*. IEEE, 287–294.
[37] Yong Luo, Yonggang Wen, Ling-Yu Duan, and Dacheng Tao. 2018. Transfer metric learning: Algorithms, applications and outlooks. arXiv:1810.03944. Retrieved from https://arxiv.org/abs/1810.03944.

[38] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. arXiv:1706.06083. Retrieved from https://arxiv.org/abs/1706.06083.
[39] Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. 2019. Metric learning for adversarial robustness. In *Proceedings of the Advances in Neural Information Processing Systems*. 480–491.
[40] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1765–1773.
[41] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2574–2582.
[42] Tongguang Ni, Xiaoqing Gu, Hongyuan Wang, Zhongbao Zhang, Shoubing Chen, and Cui Jin. 2018. Discriminative deep transfer metric learning for cross-scenario person re-identification. *Journal of Electronic Imaging* 27, 4 (2018), 043026.
[43] Gang Niu, Bo Dai, Makoto Yamada, and Masashi Sugiyama. 2014. Information-theoretic semi-supervised metric learning via entropy regularization. *Neural Computation* 26, 8 (2014), 1717–1762.
[44] Vahid Noroozi, Lei Zheng, Sara Bahaadini, Sihong Xie, and Philip S Yu. 2017. Seven: Deep semi-supervised verification networks. arXiv:1706.03692. Retrieved from https://arxiv.org/abs/1706.03692.
[45] Yaxin Peng, Lingfang Hu, Shihui Ying, and Chaomin Shen. 2018. Global nonlinear metric learning by gluing local linear metrics. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 423–431.
[46] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified defenses against adversarial examples. arXiv:1801.09344. Retrieved from https://arxiv.org/abs/1801.09344.
[47] Rakesh Kumar Sanodiya and Jimson Mathew. 2019. A framework for semi-supervised metric transfer learning on manifolds. *Knowledge-Based Systems* 176 (2019), 1–14.
[48] Yuan Shi, Aurélien Bellet, and Fei Sha. 2014. Sparse compositional metric learning.. In *Proceedings of the Association for the Advancement of Artificial Intelligence*. 2078–2084.
[49] Jimeng Sun, Fei Wang, Jianying Hu, and Shahram Edabollahi. 2012. Supervised patient similarity measure of heterogeneous patient records. *ACM SIGKDD Explorations Newsletter* 14, 1 (2012), 16–24.
[50] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble adversarial training: Attacks and defenses. arXiv:1705.07204. Retrieved from https://arxiv.org/abs/1705.07204.
[51] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. The space of transferable adversarial examples. arXiv:1704.03453. Retrieved from https://arxiv.org/abs/1704.03453.
[52] Chunzhi Wang, Pan Wu, Lingyu Yan, Zhiwei Ye, Hongwei Chen, and Hefei Ling. 2021. Image classification based on principal component analysis optimized generative adversarial networks. *Multimedia Tools and Applications* 80, 6 (2021), 9687–9701.
[53] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. 2017. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*. 2593–2601.
[54] Lei Wang, Zhu-Hong You, Li-Ping Li, Kai Zheng, and Yan-Bin Wang. 2019. Predicting circRNA-disease associations using deep generative adversarial network based on multi-source fusion information. In *Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine*. IEEE, 145–152.
[55] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. 2019. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5022–5030.
[56] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. 2019. Ranked list loss for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5207–5216.
[57] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. 2006. Distance metric learning for large margin nearest neighbor classification. In *Proceedings of the Conference on Neural Information Processing Systems*.
[58] Kilian Q Weinberger and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10, 2 (2009), 207–244.
[59] Eric Wong and Zico Kolter. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the IInternational Conference on Machine Learning*.
[60] Pengtao Xie, Wei Wu, Yichen Zhu, and Eric Xing. 2018. Orthogonality-promoting distance metric learning: Convex relaxation and theoretical analysis. In *Proceedings of the IInternational Conference on Machine Learning*.
[61] Eric P. Xing, Michael I. Jordan, Stuart J. Russell, and Andrew Y. Ng. 2003. Distance metric learning with application to clustering with side-information. In *Proceedings of the Advances in Neural Information Processing Systems*. 521–528.
[62] Feiyu Xiong, Moshe Kam, Leonid Hrebien, Beilun Wang, and Yanjun Qi. 2016. Kernelized information-theoretic metric learning for cancer diagnosis using high-dimensional molecular profiling data. *ACM Transactions on Knowledge Discovery from Data* 10, 4 (2016), 1–23.

[63] Jie Xu, Lei Luo, Cheng Deng, and Heng Huang. 2018. New robust metric learning model using maximum correntropy criterion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
[64] Yonghui Xu, Sinno Jialin Pan, Hui Xiong, Qingyao Wu, Ronghua Luo, Huaqing Min, and Hengjie Song. 2017. A unified framework for metric transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 29, 6 (2017), 1158–1171.
[65] Yonghui Xu, Bo Xu, Jingtang Zhong, Zhen Zhu, Pengshuai Yin, Huaqing Min, et al. 2018. A novel transfer metric learning approach based on multi-group. In *Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2184–2189.
[66] Yonghui Xu, Han Yu, Yuguang Yan, Yang Liu, et al. 2020. Multi-component transfer metric learning for handling unrelated source domain samples. *Knowledge-Based Systems* 203 (2020), 106132.
[67] Zhiyu Xue, Shaoyang Yang, Mengdi Huai, and Di Wang. 2021. Differentially private pairwise learning revisited. IJCAI.
[68] Pengshuai Yang, Yupeng Zhai, Lin Li, Hairong Lv, Jigang Wang, Chengzhan Zhu, and Rui Jiang. 2020. A deep metric learning approach for histopathological image retrieval. *Methods* 179 (2020), 14–25.
[69] Zhongliang Yang, Nan Wei, Qinghe Liu, Yongfeng Huang, and Yujin Zhang. 2019. GAN-TStega: Text steganography based on generative adversarial networks. In *Proceedings of the International Workshop on Digital Watermarking*. Springer, 18–31.
[70] Han-Jia Ye, De-Chuan Zhan, Xue-Min Si, Yuan Jiang, and Zhi-Hua Zhou. 2016. What makes objects similar: A unified multi-metric learning approach. In *Proceedings of the Advances in Neural Information Processing Systems*. 1235–1243.
[71] Pourya Zadeh, Reshad Hosseini, and Suvrit Sra. 2016. Geometric mean metric learning. In *Proceedings of the International Conference on Machine Learning*. 2464–2471.
[72] Mengting Zhan, Shilei Cao, Buyue Qian, Shiyu Chang, and Jishang Wei. 2016. Low-rank sparse feature selection for patient similarity learning. In *Proceedings of the 2016 IEEE 16th International Conference on Data Mining*. IEEE, 1335–1340.
[73] Dingyi Zhang, Yingming Li, and Zhongfei Zhang. 2020. Deep metric learning with spherical embedding. *Advances in Neural Information Processing Systems* 33 (2020).
[74] Liheng Zhang and Guo-Jun Qi. 2020. Wcp: Worst-case perturbations for semi-supervised deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3912–3921.
[75] Yiru Zhao, Zhongming Jin, Guo-jun Qi, Hongtao Lu, and Xian-sheng Hua. 2018. An adversarial approach to hard triplet generation. In *Proceedings of the European Conference on Computer Vision*. 501–517.

Received August 2020; revised August 2021; accepted November 2021