Adaptive Margin Based Deep Adversarial Metric Learning

Zhuoyi Wang, Yigong Wang, Bo Dong, Sahoo Pracheta, Kevin Hamlen, Latifur Khan
Department of Computer Science, The University of Texas at Dallas, Richardson TX, USA
{zxw151030, yxw158830, Bo.Dong3, Pracheta.Sahoo, hamlen, lkhan}@utdallas.edu

***Abstract*—In the past decades, learning an effective distance metric between pairs of instances has played an important role in the classification and retrieval task, for example, the person identification or malware retrieval in the IoT service. The core motivation of recent efforts focus on improving the metric forms, and already showed promising results on the various applications. However, such models often fail to produce a reliable metric on the ambiguous test set. It happens mainly due to the sampling process of the training set, which is not representative of the distribution of the negative samples, especially the examples that are closer to the boundary of different categories (also called hard negative samples). In this paper, we focus on addressing such problems and propose an adaptive margin deep adversarial metric learning (AMDAML) framework. It exploits numerous common negative samples to generate potential hard (adversarial) negatives and applies them to facilitate robust metric learning. Apart from the previous approaches that typically depend on the search or data augmentation to find hard negative samples, the generation of adversarial negative instances could avoid the limitation of domain knowledge and constraint pairs' amount. Specifically, in order to prevent over fitting or under-fitting during the training step, we propose an adaptive margin loss that preserves a flexible margin between the negative (include the adversarial and original) and positive samples. We simultaneously train both the adversarial negative generator and conventional metric objective in an adversarial manner and learn the feature representations that are more precise and robust. The experimental results on practical data sets clearly demonstrate the superiority of AMDAML to representative state-of-the-art metric learning models.**

***Index Terms*—Adversarial Learning, Deep Metric Learning, Deep Neural Network, Adaptive Margin**

I. Introduction

Metric learning aims to learn a distance metric from example pairs to measure their similarities, which makes the classification or retrieval tasks more efficient. It plays a fundamental role in a variety of machine learning and pattern recognition applications, such as person re-identification [1], [2], image classification [3], [4], multi-output tasks [5], [6], or security application [7], [8]. Currently, the existing metric learning training methods are typically based on an objective function that minimizes the distance between similar examples and maximizes the distance between dissimilar examples. The distance of a pair of examples $D(\boldsymbol{x}, \boldsymbol{y})$ typically uses the Mahalanobis distance [9], which could be described as: $D^2_M(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T M(\boldsymbol{x} - \boldsymbol{y})$. Here, the symmetric positive definite (SPD) matrix M can be learned through training data pairs to reflect the similarity, which represents a linear transformation. Furthermore, some recent methods were proposed to replace the linear Mahalanobis matrix by non-linear methods, like kernel trick [10], [11], or the deep neural networks, e.g. CNN [3], [12]–[14].

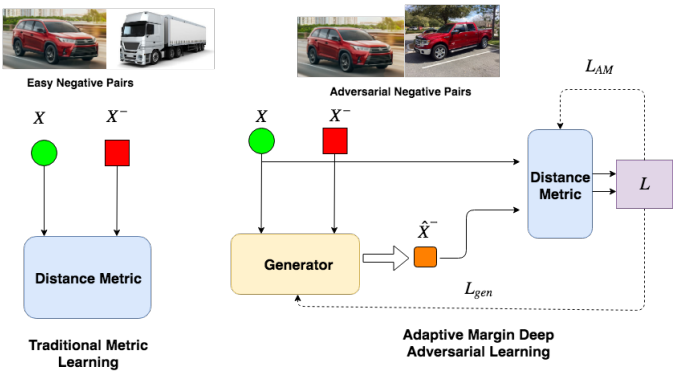


Fig. 1: The comparison of traditional metric learning and proposed AMDAML. Different colors represent various classes. We utilize the generated adversarial negatives to train the distance metric instead of the observed negatives, then simultaneously train the adversarial generator and metric objective in an adversarial manner.

During the training phase, positive and negative pairs will be sampled from the training set to train the distance metric. Compared with the "easy negative samples" that are relatively easier to be distinguished from positive samples. A common example of them is in the object recognition scenario: car seems different from lorry which could be regarded as an easy negative example, but it will become a potential hard negative example if the car is replaced by a pick-up. Although pairs of hard negative [15] and positive examples would produce gradients with larger magnitudes [16], they usually account for the tiny minorities of training samples; the vast majority of negative ones, which are called as "easy negatives" in the previous description, make limited contribution to the optimization in metric learning. So we propose that enough hard negative samples could impact the learned distance metric performance significantly.

Some existing methods have been proposed to address the previous problems, such as the Data Augmentations [12], [17] and Hard Negative Searching [17], [18]. However, these methods strongly depend on the heuristic rules or the negative distribution of the training set to select proper pairs. Therefore, their performances would be limited. Following the idea of adversarial training [19]–[21], we provide a solution that

generates ambiguous but critical adversarial negative samples to represent potential "hard negatives", and further enhance the algorithm robustness. Apart from the existing adversarial learning strategy [19], which aims to search the negative samples in the neighborhood of current instances to confuse the learned metric, we focus on generating "adversarial negatives", which used to simulate potential hard negatives in the unobserved space. We expect them to be able to attack/confuse the learned metric as much as possible.

This paper proposes an adaptive margin based on deep adversarial metric learning (AMDAML) framework to address the previous challenges. The distance metric will be learned from both original training pairs and the generated adversarial negative pairs. Specifically, we design an adaptive margin loss to dynamically adapt its margin to distinguish the different categories of instances, under the assistance of the adversarial samples. It also applied to avoid over-fitting or under-fitting during the learning process, especially when relative large amount of adversarial negatives that participated into the training step. We aim to generate potential adversarial negatives from the existing instances, which works as an important synthetic complements (as hard negative ones) during the learning step. We jointly train the adversarial negative generator and metric objective in an adversarial manner, so that the discrimination and robustness of our learned feature representation will be enhanced. The main contributions of this paper are summarized as follows: (1) We propose a novel framework AMDAML, which is able to generate adversarial negative pairs from the existing negative samples in training data, and enhance the robustness and discriminating power of the model; (2) We implement an adaptive margin loss for AMDAML that could dynamically preserve an adaptive margin between negative (both original and adversarial) and positive pairs. To our knowledge, it is the first work to introduce an adaptive margin with the distance metric for adversarial learning. (3) AMDAML empirically outperforms the state-of-the-art metric learning models on real-world image sets.

II. Relative Work

Based on the types of Deep neural networks (DNNs), several deep metric learning (DML) works are proposed to solve real-world applications such as structured feature embedding [3], face recognition [13]. The most famous one is the Siamese network [22], [23], which is proposed with a contrastive loss for image recognition. It uses a two-channel neural network architecture to learn a non-linear distance metric to minimize the distance between similar sample pairs, and push the pairwise distance between dissimilar pairs larger than a fixed margin. Furthermore, studies [24]–[26] have also explored computing feature representations using triplet or N-pair loss, where constraints are specified by relative similarity and dissimilarity among instances; Smart-mining [27] applied the global loss to optimize the deep metric with hard-samples mining, and Proxy-NCA [28] utilized a proxy-agent view as classification loss. Previous approaches mainly generate the metric by designing discriminative losses, they do not consider the aforementioned issue about generalization capability. Additionally, studies exploring sample-mining strategies aim to improve the performances of metric learning through the selection [29], [30], or generation [31] of hard-samples. Different from previous methods, AMDAML provides a further analysis of the adversarial samples of DML, shows how to improve robustness by introducing the adversarial sample term. Furthermore, we consider the adaptive margin bound of the distance metric, which is more practical in the learning step with adversarial samples.

III. Approach

As shown in Figure. 1, the AMDAML consists of two steps: generating the adversarial negative samples, and learning the reliable distance metric. We introduce necessary notations in Sec. III-A, and explain two parts of AMDAML with relative discussion in Sec. III-B and Sec. III-C. Finally, we provide the optimization step and relative prove in Sec. III-D.

A. Preliminaries

Given a set of training instances $D = \{(x_i)\}_{i=1}^N$, $x_i \in \mathcal{R}^d$ with associated label set $Y = \{(y_i)\}_{i=1}^N$, $y_i \in [1 \dots C]$, where C is the number of existing classes. In general, the goal of (deep) metric learning is to learn a nonlinear embedding function $\phi : \mathcal{R}^d \to \mathcal{R}^{d_1}$, which could map the d-dimensional input into an embedding space with d_1-dimension. Here, given an anchor point $x_i \in D$, x_i^+ is the positive point with $y_i = y_i^+$, we define the (x_i, x_i^+) as such positive pairs; if $y_i \neq y_i^-$, the negative point with corresponding pair could be (x_i, x_i^-). The ϕ is typically applied to make the distance of a positive pair examples $D_\phi(x_i, x_i^+)$ as small as possible, and expand the distance between negative pairs (x_i, x_i^-) as much as possible simultaneously. We also define the description of the generated adversarial negative samples as $\hat{x}$.

B. Adaptive Margin Loss

In general, the parameter of deep metric learning is ϕ that comes from the network parameters, existing works typically use it to measure the squared Euclidean distance of an input pair: $D_\phi(x_i, x_j)^2 = ||f(x_i;\phi) - f(x_j;\phi)||_2^2$, $f(x_i;\phi)$. It means the representation of the final output layer for x_i. In the paper, we apply a M-layers Deep Neural Nets (DNN) for the learning step of ϕ. For X_i^m, which is the feature representation generated at layer m could be described as:

$$f(X_i;\phi) = X_i^M, X_i^m = \sigma(W^{(m)} X_i^{(m-1)} + b^{(m)}),$$
$$\forall i = 1, \dots, N, \forall m = 1, \dots, M; X_i^{(0)} = X_i. \quad (1)$$

where $W^{(m)}$ are the weights in the m^{th} layer, b^m refers to the corresponding biases in this layer, and σ is a non-linear activation function. For simplicity, we consider the network parameters as a whole: $\phi^{(m)} = \left[W^{(m)}, b^{(m)}\right]$, and $\phi = \{\phi^{(1)}, \dots, \phi^{(M)}.\}$ Our adaptive margin loss function consists of two terms: the pairwise constrain term and the regularization part, which are formulated as follows:

$$\min_{\phi} L_{AM}(X;\phi) = \min_{\phi}(L_{PC}(X;\phi) + \lambda L_R) \tag{2}$$

here the λL_R is the regularization term used to smooth the parameters of the DNN. Our adaptive margin loss aims to compact the intra-class and separate the inter-class, while preserving an adaptive margin between them. For the pairwise constrain term, we expect there will be a suitable adaptive margin between the distance of positive pairs and negative pairs (negative pairs include the original pairs and generated pairs). Here, we define the mean distance of the positive pairs D_p and the negative pairs D_n as: $D_p = \frac{1}{N^+}\sum_{i=1}^{N^+} D_\phi(x_i, x_i^+)$, $D_n = \frac{1}{N^-}\sum_{i=1}^{N^-} D_\phi(x_i, x_i^-)$. For the positive pair (x_i, x_i^+), $D_p \leq \mathcal{T}_{up}$ ($\mathcal{T}_{up}$ is an adaptive up-threshold); and for the negative pair (x_i, x_i^-), $D_n \geq \mathcal{T}_{down}$ ($\mathcal{T}_{down}$ is an adaptive down-threshold). Considering the mutation of pairwise distance, $\mathcal{T}_{up}$ and $\mathcal{T}_{down}$ should be flexibly determined. When the negative-pair distances are increased, $\mathcal{T}_{up}$ should be smaller than a moderate upper bound to make the inter-class samples more distinguishable; the similar things also happened for the dynamic adaptation of the $\mathcal{T}_{down}$ value. Based on the softplus function [1], [32](the smooth approximation of the RELU), we use the following equations for the adaptive margins:

$$\mathcal{T}_{up} = \frac{1}{\alpha+1}(1 - e^{(-\alpha D_n)}) \tag{3}$$

$$\mathcal{T}_{down} = \frac{1}{\beta-1}\log(1 + e^{(\beta D_p)}) \tag{4}$$

For the adaptive margins $\mathcal{T}_{down}$ and $\mathcal{T}_{up}$, these could be regarded as a nonlinear mapping of the average positive and negative pairs distances D_p, D_n, respectively. The underlying mechanism of the Eq. 3 is that, when the D_p goes up in further iterations, the $\mathcal{T}_{down}$, also the lower bound of D_n, will increase further more. So the mapping function will gradually increase the $\mathcal{T}_{up}$ to avoid the under-fitting problem by excessively penalizing the positive distances into an upper bound value. Meanwhile, increasing of $\mathcal{T}_{down}$ into a proper bound will also enhance the discrimination of positive pairs from the negative ones and avoid over-fitting. We will prove in the Sec. III-D that whenever the value of D_p and D_n changed, $\mathcal{T}_{up}$, $\mathcal{T}_{down}$ will be bounded in a certain value, which can well avoid the overfitting or underfitting problem caused by the fixed margin strategy in the training process.

Here, in order to simplify the representation of pairwise constraint, we preset the $\mathcal{T}_{up} = \tau - \gamma$, $\mathcal{T}_{down} = \tau + \gamma$. From the middle plot from the Figure. 2, it shows that the intra-class distance should be smaller than $\tau - \gamma$ ($\mathcal{T}_{up}$), and the distance between instances of different classes will be larger than $\tau + \gamma$ ($\mathcal{T}_{down}$). And $\tau > \gamma$, so given the input pair (x_i, x_j),the adaptive margin between the intra-class samples and inter-class instances can be enforced by: $y_i(\tau - D_\phi^2(x_i, x_j)) - \gamma > 0$. By applying the hinge-like loss function, the final adaptive margin loss could be described as follows:

$$\begin{aligned}\min_{\phi} L_{AM}(X;\phi) &= L_{PC}(X;\phi) + \lambda L_R \\ &= \sum_{i=1}^{N}\left[\gamma - y_i(\tau - D_\phi^2(x_i, x_j))\right]_+ + \\ &\frac{\lambda}{2}\sum_{m=1}^{M}(||W^{(m)}||_F^2 + ||b^{(m)}||_2^2)\end{aligned} \tag{5}$$

where m is the number of layers in the network structure, $||W||_F$ represents the Frobenius norm of the weight W, $||b||_2$ represents the Euclid norm of bias b, and λ is a regularization parameter for the corresponding term L_R.

We compare our function with two previous common fixed margin functions in the Figure. 2, including the contrastive and triplet loss. For the L_{PC} in Eq. 5 and the contrastive loss [22] which penalizes the inter-class distances bigger than a fixed positive margin, we expand them as follow:

$$L_{PC} = \sum_{i=1}^{N^+}\left[\gamma - \tau + D_\phi^2(x_i, x_i^+)\right]_+ + \sum_{j=1}^{N^-}\left[\gamma + \tau - D_\phi^2(x_j, x_j^-)\right]_+ \tag{6}$$

$$L_{Cont} = \sum_{i=1}^{N^+}\left[0 + D_\phi^2(x_i, x_i^+)\right]_+ + \sum_{j=1}^{N^-}\left[\omega - D_\phi^2(x_j, x_j^-)\right]_+ \tag{7}$$

where (x_i, x_i^+) means the positive pairwise, N^+ is the corresponded number of the positive pairs. Eq. 7 shows that we could regard contrastive loss as a special case of L_{PC} when $\tau = \gamma$ (then $\omega_1 = \tau + \gamma$, ω_1 is the fixed margin of the contrastive). Figure. 2 shows that adaptive margin loss will constrain the similar pairs distance which is smaller than $\tau - \gamma$, and negative pairs distance larger than $\tau + \gamma$. For contrastive loss, when $\tau = \gamma$, the distance between similar pairs may be limited to zero, and its constraint may be too strong and lead to over-fitting. The triplet loss needs a fixed margin to penalize the distance between negative samples and positive (anchor) instances. However, if the dissimilar instances are closer than similar instances (especially when positive x_i^+ is closer to negative x_i^- and further from anchor x_i), it means the triplet loss $\left[\omega_2 + D^2(x_i, x_i^+) - D^2(x_i, x_i^-)\right]_+$ may be equal to 0. The optimization of triplet loss will incorrectly ignore this constraint due to its zero loss value. For this problem, we apply regularization term to avoid zero loss value. Furthermore, benefits from the adaptive margin strategy, our model could update the $\mathcal{T}_{up}$ and $\mathcal{T}_{down}$ which jointly considered the variations of negative distance and positive distance, and it is more suitable for the scenario that generated adversarial samples dynamically inserted.

C. Adversarial Negative Samples Generator

The existing metric learning approaches take advantage of the observed pairs to train distance metrics. However, the sampling strategies of the positive/negative pairs will make

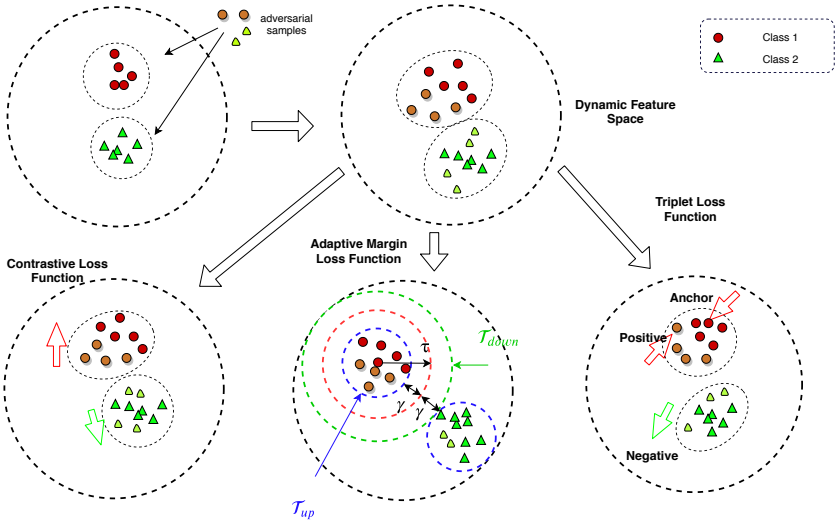


Fig. 2: Comparison of our method with two fixed margin approaches, the left one shows the contrastive loss function penalizes the positive distances and negative distances with a fixed margin; the right show the triplet loss function minimizes the relative distances between the intra-class samples and inter-class samples with a fixed margin; the middle one shows adaptive margin loss function preserves the positive distances and negative distances with an adaptive margin.

the selected "hard negative samples" not enough to fully describe the distributions of negative samples, especially when the number of sampling pairs is limited. The main target of the generator is applied to produce the adversarial negatives from original instances and use them as complements to the observed data. As described in Sec. III-B, our metric learning approaches aim to obtain the parameters ϕ through optimization of the adaptive margin objective function in Eq. 5. In this section, we aim to enhance the training procedure through adversarial negative generator. We simultaneously optimize the adversarial samples and the distance metric in an adversarial manner by utilizing the adversarial negative samples as the adversary in Eq. 5. Here, different from existing negative pairs (x_i^+, x_i^-), we apply $\hat{x}$ to attack the metric objective, it is produced through an advanced mix-up process [33].

In the generating process of $\hat{x}$, as the adversarial instances will be generated dynamically according to different sampling examples, we simultaneously collect x_i^+, x_i^- as the input of the producing of $\hat{x}$. The adversarial negative samples should be: 1) close to both the selected instances and their corresponding negative examples in the original feature space; 2) misclassified by the learned metric in sec. III-B. We describe the objective function of the adversarial generator with parameter θ_g as follows:

$$\begin{aligned}\min_{\theta_g} L_{adv} &= \max L_{PC}((x_i^+, \hat{x}_i); \phi) + [\theta_g x_i^+ + (1-\theta_g)x_i^-] \\ &= [\theta_g x_i^+ + (1-\theta_g)x_i^-] - \min L_{PC}((x_i^+, \hat{x}_i); \phi)\end{aligned} \tag{8}$$

In Eq. 8, $[\theta_g x_i^+ + (1-\theta_g)x_i^-]$ aims to generate adversarial samples, that close to both the original sampled instances and negatives. The core idea of this process is to extend the training distribution, through incorporating the prior knowledge that linear interpolations of feature vectors should lead to linear interpolations of the associated targets. It will produce large magnitudes for the training procedure of metric learning, also could be regarded as a form of data augmentation, that encourages the model to behave linearly in-between training examples. For the maximum of $L_{PC}((x_i^+, \hat{x}_i); \phi)$, it means adversarial samples aims to confuse the learned metric as much as possible, and encourage the distance between the negative and positive samples smaller than the margin value. The procedure of adversarial training manner enhances both the discriminated power and robustness of the learned metrics to solve the potential "hard negative" instances.

D. Adversarial Metric Learning

The AMDAML framework will simultaneously train the adversarial negative generator and the adaptive margin distance metric with the following objective function:

$$\min_{\phi, \theta_g} L = L_{AM} + \epsilon L_{adv} \tag{9}$$

where $\epsilon \in [0, 1]$ is the parameter to balance the weights of different terms. AMDAML consists of the adaptive metric objective and adversarial negative generator, they form an adversarial learning scheme by optimizing the opposite objective functions. During the training period, when the adversarial negatives participate into the learning function Eq. 9, the D_n will go down while the value of D_p may increase. We will prove that under such dynamic situation of D_n and D_p value, the up or down margin of our approach (in Eq. 3) could still be bounded into a certain range of value, and further avoid the under-fitting or over-fitting during the this step.

1) Theorem 1: : $\mathcal{T}_{up}$ **and** $\mathcal{T}_{down}$ **have upper bounds respectively.**

2) Proof of Theorem 1: : From $\frac{\partial \mathcal{T}_{up}}{\partial D_n} = \frac{\alpha}{\alpha+1}\exp(-\alpha D_n)$, it is easy to see if D_n goes up, $\mathcal{T}_{up}$ will increase. When $D_n \geq 0$, we could get: $\lim_{D_n \to \infty} \mathcal{T}_{up} = \frac{1}{\alpha+1}$. When D_n increased $\mathcal{T}_{up}$ has an upper bound, and D_p also has the same upper bound since $D_p \leq \mathcal{T}_{up}$.Then, from

$$\frac{\partial \mathcal{T}_{down}}{\partial D_p} = \frac{\beta}{(\beta-1)\ln 2}\frac{exp(\beta D_p)}{exp(\beta D_p)+1} \tag{10}$$

it is obviously that $\mathcal{T}_{down}$ is increasing when $\beta > 1$. Since D_p has the upper bound $\frac{1}{\alpha+1}$, $\mathcal{T}_{down}$ also could be bounded at the upper value:

$$\lim_{D_p \to \frac{1}{\alpha+1}} \mathcal{T}_{down} = \frac{1}{\beta-1}\log\left(1+\exp\left(\frac{\beta}{\alpha+1}\right)\right) \tag{11}$$

3) Theorem 2: : $\mathcal{T}_{up}$ **and** $\mathcal{T}_{down}$ **have lower bounds respectively.**

4) Proof of Theorem 2: : $\mathcal{T}_{down}$ has a lower bound:

$$\lim_{D_p \to 0} \mathcal{T}_{down} = \frac{1}{\beta-1} \tag{12}$$

Since $D_n \geq \mathcal{T}_{down}$, D_n has the same lower bound. Then we show that $\mathcal{T}_{up}$ has a lower bound:

$$\lim_{D_n \to \frac{1}{\beta-1}} \mathcal{T}_{up} = \frac{1}{\alpha+1}(1-\exp\frac{-\alpha}{\beta-1}) \tag{13}$$

Finally, when bounded, $D_n > \frac{1}{\beta-1}, D_p < \frac{1}{\alpha+1}$, and:

$$D_n - D_p > \frac{\alpha-\beta+2}{(\beta-1)(\alpha+1)} \tag{14}$$

it prove that adaptive margin loss preserve a proper margin between negative (both original and adversarial) and positive pairs when $\alpha - \beta + 2 > 0$ finally, and the upper or lower bound of margin value is existed during the training step.

For the optimization, we apply the Stochastic Gradient Descent (SGD) back-propagation method to optimize the parameters of the L.

We show the overall process in Algorithm. 1.

Algorithm 1 AMDAML

Require: X - Training sample pairs; I - iterations times; $\tau, \gamma, \lambda, \lambda_1, \lambda_2, \epsilon$ - hyper parameters for the approach;

Ensure: Parameter of adversarial generator θ_g and parameters of the deep metric learning function ϕ.

```
Pre-train φ under the sampling pairs from D without the generated adversarial negatives through Eq. 5
Initialize θ_g, making i = 0;
while i < I: do
    Random sampling mini-batch of M pairs of instances
    Do forward propagation to get representations for M
    Compute the T_up and T_down through Eq. 3
    Apply the hyper parameters and jointly optimize θ_g and φ in Eq.9, through back-propagate.
end while
return θ_g and φ
```

IV. Experiment

In this section, empirical investigations are conducted to validate the effectiveness of AMDAML. In detail, we first compare the performance of the proposed approach with four state-of-the-art metric learning methods (**GB-LMNN** [34], **HDML** [10], **GMML** [35], **AML** [19]), and two deep metric learning methods: **Contrastive** [12], **Triplet** [3] on four benchmark classification datasets. Next, all the methods are compared on two image datasets related to face verification and matching. Finally, the parametric sensitivity of AMDAML is studied. More experiment result could be seen in the

1) Datasets: For classification task, we evaluated all the methods on four widely used benchmark datasets named **MNIST** [36][1], **FASHION-MNIST** [37][2], **SVHN** [38][3], **CIFAR-10** [39][4]. In this task, We compare all approaches over 10 random trials, and in each trial, 20% of examples are randomly selected as the training set, and the rest are used for testing. The training pairs ($\{X_i\}_{i=1}^N$) are generated through randomly picked 10,000 constraint pairs among all the training examples.

We also applied four datasets to evaluate the capabilities of all methods on face verification and model generalization test. For the face verification, we select **PubFig** [40][5], which consists of 2*10^4 pairs of images belonging to 140 people; another one is **LFW** [41][6], which includes 13,233 unconstrained face images of 5,749 samples. For both of them, we extracted the adopted features following [19], then select first 60% pairs for face training and the rest are used for verification testing. Next, for the model generalization test, we select **CAR196** [42] for the experiment. The Cars196 dataset contains 16,185 images from 196 car models, we used the first 98 models with 8,054 images for training and the remaining for testing. Here, we also adopt the same data prepossessing steps for the to make fair comparisons with other baselines.

2) Implementation: AMDAML is implemented using *Pytorch* 0.4.0 library[7], for the deep metric network, it is trained through a AlexNet deep network [43], and randomly initialized the weight parameter on each layer. This network structure used for Contrastive and Triplet methods for fair comparisons. The learning rate is 0.01 for the deep metric

[1] http://yann.lecun.com/exdb/mnist/
[2] https://github.com/zalandoresearch/fashion-mnist
[3] http://ufldl.stanford.edu/housenumbers/
[4] https://www.cs.toronto.edu/ kriz/cifar.html
[5] http://www.cs.columbia.edu/CAVE/databases/pubfig/
[6] http://vis-www.cs.umass.edu/lfw/
[7] https://pytorch.org/

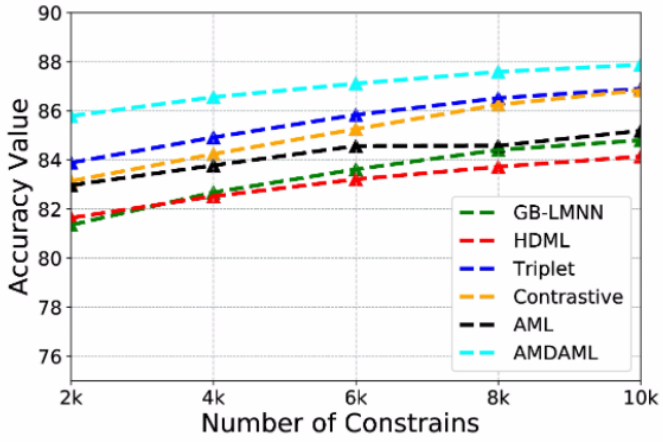
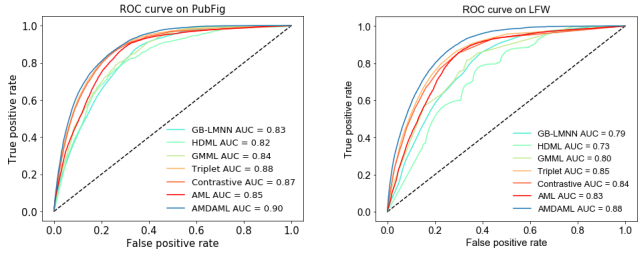


Fig. 3: Accuracy rates of competing methods with increasing number of constraint pairs on **FASHION-MNIST** dataset.

network, and 0.002 for the generator net. We fixed the maximum training iteration to 5,000 and set the batch size as 128, at last, we empirically fixed the parameters $\lambda, \lambda_1, \lambda_2$ and ϵ as 1, 0.1, 50, 0.1 to balance the weights of different terms. For other methods, we test all of them based on code released by corresponding authors, and initialize the hyper parameters in each method based on the author's recommendation, then fine-tune via a validation set. For triplet and contrastive, we use the same network architecture and corresponding parameters with our approach. Overall, we repeat each experiment 10 times independently and report the average result.

3) Experiments on Classification: We adopted the classification accuracy and macro F1 score as the evaluation criteria. Moreover, we perform the t-test (significance level 0.05) to investigate the superiority of our method to the best baseline method on all datasets. The results are shown in Table. I. We could see GB-LMNN and HDML are performed poorly on most of the datasets, AML is better than these methods, it proved the effectiveness of the adversarial learning manner. However, AML still applies the Mahalanobis distance, so it could not perform well on complex image sets such as CIFAR-10. Benefit from DNN, Contrastive, Triplet and AMDAML have significantly better performance. Compared with Contrastive and Triplet, our approach could achieve not only significantly higher classification accuracy, but also higher F1 score. It proved that generated adversarial is more effective as a complement pairwise constraint, the learned distance metric presents better performance with synthetic negative samples.

To illustrate the effectiveness of different numbers of constraints on the learned distance metric, we use various input constraint amounts for the competing methods. Figure. 3 shows the comparison of accuracy on FASHION-MNIST dataset. The higher variance of GB-LMNN, HDML indicates the strong dependency on the quality of input constraints. Compared with Triplet and Contrastive, AMDAML provides



(a) (b)

Fig. 4: ROC curve with AUC value of each method on PubFig and LFW datasets

the best and relative stable classification performance, indicating the learned distance metric presents strong robustness with synthetic hard negative samples.

4) Experiments on Verification: The goal of face verification in the two datasets is to determine whether a pair of face images belongs to the same person. We plot the Receiving Operator Characteristic (ROC) curve by changing the thresholds of different distance metrics. Then the values of Area Under Curve (AUC) are calculated to evaluate the performances quantitatively. From the ROC curves and AUC values in Fig. 4, it is obvious that for both of the PubFig and LFW datasets, AMDAML outperforms all baseline approaches by providing higher AUC value. Compared with the existing methods which only exploit the observed negative samples in their form, our approach generates adversarial samples for a full description of the negative distributions, which helps for learning more robust distance metric model.

5) Experiments on generalization: The evaluation for generalization task is different with other works, we apply Recall@K metric [44], for every test sample in the CAR196 data set. We first retrieve K most similar images from the test set, if an instance of the same class is retrieved among these K samples, a score of 1 would return, otherwise it would be 0. We evaluate the generalization effective of L_{adv} for AMDAML (in Eq. 9) through explicitly comparing the performance of the recall value (R@1) retrieval result curves, mainly on the training(seen) and testing(unseen) in Fig. 5. Specifically, the training curve of the contrastive/triplet method rises quickly, however, they drop to quite a low level after several iterations in the test set. It shows that existing approaches are more likely to generalize worse on the unseen class. After employing the L_{adv} on such loss functions,the training result curve rises much slower than the original ones; but on the testing set, they steadily increase to a relatively high value. It implies that our adversarial loss could act as a regularization term to improve the generalization ability of the learned metric.

Moreover, comparing with the contrastive/triplet loss, we observe that our adaptive margin loss makes the learned metric could be adjusted more quickly to the final level, and perform more stable comparing with others. It shows that the

TABLE I: **Comparison of classification performance on competing methods under totally 10k constraints of sample pairs. ✓ indicates that AMDAML performs statistically better than the baseline method according to the p-values.**

Methods	FASHION-MNIST ✓		MNIST ✓		CIFAR-10 ✓		SVHN ✓	
	Accuracy (%)	F1 Scores	Accuracy (%)	F1 Scores	Accuracy (%)	F1 Scores	Accuracy (%)	F1 Scores
GB-LMNN	84.78 ± 0.07	84.63 ± 0.01	94.97 ± 0.04	94.89 ± 0.00	45.81 ± 0.10	45.71 ± 0.08	54.47 ± 0.09	54.35 ± 0.06
HDML	84.16 ± 0.09	84.02 ± 0.02	94.13 ± 0.05	94.04 ± 0.01	45.65 ± 0.08	45.53 ± 0.09	54.62±0.10	54.51 ± 0.03
GMML	83.28 ± 0.08	83.16 ± 0.05	94.69 ± 0.05	94.60 ± 0.01	40.79 ± 0.10	40.70 ± 0.06	50.23 ± 0.06	50.14 ± 0.04
AML	85.27 ± 0.08	85.14 ± 0.05	95.05 ± 0.05	94.94 ± 0.01	49.58 ± 0.12	49.47 ± 0.06	55.28 ± 0.14	55.17 ± 0.08
Triplet	90.91 ± 0.09	90.79 ± 0.04	98.11 ± 0.05	98.02 ± 0.01	63.67 ± 0.12	63.58 ± 0.05	78.41 ± 0.11	78.35 ± 0.04
Contrastive	89.51 ± 0.10	86.41 ± 0.03	97.93 ± 0.03	97.78 ± 0.02	62.58 ± 0.13	50.97 ± 0.06	77.49 ± 0.13	77.37 ± 0.05
AMDAML	**91.95± 0.10**	**87.86 ± 0.05**	**98.86 ± 0.04**	**98.25 ± 0.02**	**65.12 ± 0.12**	**53.88 ± 0.07**	**80.03 ± 0.12**	**79.92 ± 0.07**

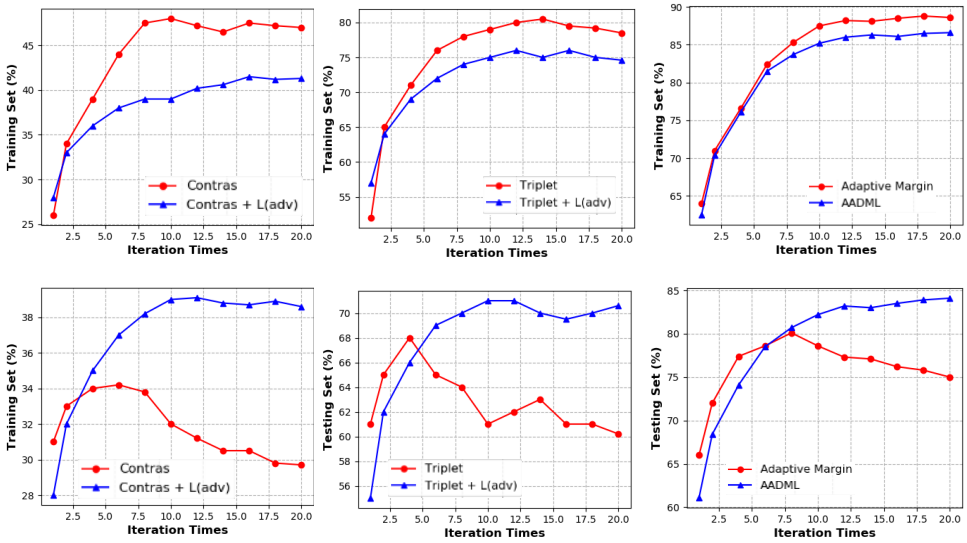


Fig. 5: Top figs show the Recall@1 curves on seen training classes, and the bottom figs describe the unseen testing classes, experiments on the CARS dataset.

adaptive property of our metric learning approach.

6) Parametric Sensitivity Analysis: There are two parameters which might influence the model performance: α and β in Eq. 3. As shown in Sec. III-D, $\alpha - \beta + 2 > 0$. Here we will analyze the influence by changing one parameter while fixing the other one.

The influence of α on $\mathcal{T}_{up}$: We plot different $\mathcal{T}_{up}$ values under various α in Fig. 6(a). It is obvious that $\mathcal{T}_{up}$ would be decreased when α goes up (from blue, orange to green line). If α is too large, $\mathcal{T}_{up}$ value could be extremely small and lead to overfitting; on the other hands, too small α would weakean the constraint of intra-class distance. Figure. 6(c) showes the correspond result on CIFAR-10 and SVHN dataset with a fixed β and varying α, we can see that when α is relative low, constrain would be weaken, and accuracy would go down. The performance is relative optimal when α is around 5.0.

The influence of β on $\mathcal{T}_{down}$: From Fig. 6(b), we could find that when β value increaes(from blue, orange to green line), $\mathcal{T}_{down}$ diminishes. Similarly, a large β value would cause small down-margin $\mathcal{T}_{down}$ and under-fitting might happened; small β will lead to huge $\mathcal{T}_{down}$ and over-fitting. Results with a certain α value and varying β are shown in Figure. 6(d). The results indicate that performance reaches best when β = 3.0.

V. Conclusion and Future Work

We propose an adversarial learning based metric learning framework, named AMDAML. This framework contains two essential components including adaptive margin based distance metric learning term and adversarial negative generator part. Unlike existing metric learning approaches that focus on sampling strategies to maintain the quality of the metric, we exploit existing instances to generate adversarial negatives to the training step. Then, combined with the adaptive margin metric objective function, AMDAML could intensify the discriminable within the learned embedding. Our empirical evaluation of real-world image data sets shows that our approach effectively improves the performance of the existing methods in an adversarial manner.

In the future work, we aim to extent our current work to more domain applications, such as the graph embedding [45], stream based mining in the cyber-security [46]. We could make AMDAML focus on tapping the potential of finging more informative samples, it is an interesting work to apply it as an efficient data augmentation approach in such area.

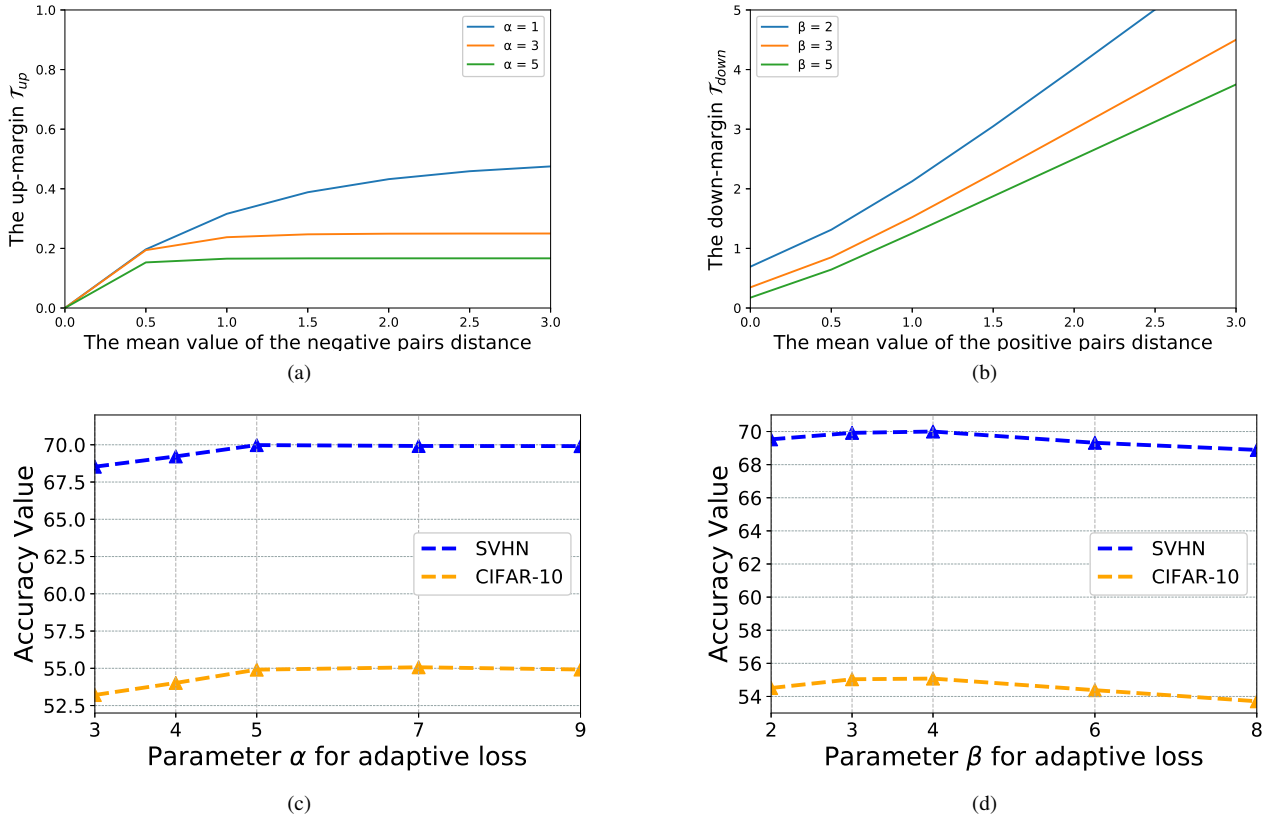


Fig. 6: The illustration of two different nonlinear mapping strategies for the adaptive margin

ACKNOWLEDGEMENT

The research reported herein was supported in part by NIH 1R21TW010991-01A1; NSA award H98230-15-1-0271, ONR award 12879034; NSF FAIN awards DGE-1931800, OAC 1931541, and DGE-1723602; NSF awards DMS-1737978 and MRI-1828467; an IBM faculty award (Research); and an HP grant. Any opinions, recom-mendations, or conclusions expressed are those of the authorsand not necessarily of the aforementioned supporters.

REFERENCES

[1] J. Wang, S. Zhou, J. Wang, and Q. Hou, "Deep ranking model by large adaptive margin learning for person re-identification," *Pattern Recognition*, vol. 74, pp. 241–252, 2018.

[2] Z. Liu, D. Wang, and H. Lu, "Stepwise metric promotion for unsupervised video person re-identification," in *ICCV*, 2017, pp. 2429–2438.

[3] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *CVPR*, 2016, pp. 4004–4012.

[4] Y. Li, X. Tian, X. Shen, and D. Tao, "Classification and representation joint learning via deep networks," in *IJCAI*, 2017, pp. 2215–2221. [Online]. Available: https://doi.org/10.24963/ijcai.2017/308

[5] W. Liu, D. Xu, I. W. Tsang, and W. Zhang, "Metric learning for multi-output tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 408–422, Feb 2019.

[6] Z. Wang, B. Dong, Y. Lin, Y. Wang, M. S. Islam, and L. Khan, "Co-representation learning framework for the open-set data classification," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 239–244.

[7] M. M. Masud, T. M. Al-Khateeb, K. W. Hamlen, J. Gao, L. Khan, J. Han, and B. Thuraisingham, "Cloud-based malware detection for evolving data streams," *ACM transactions on management information systems (TMIS)*, vol. 2, no. 3, pp. 1–27, 2008.

[8] A. Haque, L. Khan, M. Baron, B. Thuraisingham, and C. Aggarwal, "Efficient handling of concept drift and concept evolution over stream data," in *IEEE 32nd International Conference on Data Engineering*. IEEE, 2016, pp. 481–492.

[9] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *NIPS*, 2003, pp. 521–528.

[10] D. Kedem, S. Tyree, F. Sha, G. R. Lanckriet, and K. Q. Weinberger, "Non-linear metric learning," in *NIPS*, 2012, pp. 2573–2581.

[11] J. Wang, H. T. Do, A. Woznica, and A. Kalousis, "Metric learning with multiple kernels," in *NIPS*, 2011, pp. 1170–1178.

[12] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *CVPR*, 2015, pp. 4353–4361.

[13] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *arXiv preprint arXiv:1502.00873*, 2015.

[14] Z. Wang, Z. Kong, S. Changra, H. Tao, and L. Khan, "Robust high dimensional stream classification with novel class detection," in *ICDE*, 2019, pp. 1418–1429.

[15] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *ICCV*, 2015, pp. 118–126.

[16] J. Lu, J. Hu, and Y.-P. Tan, "Discriminative deep metric learning for face and kinship verification," *TIP*, vol. 26, no. 9, pp. 4269–4282, 2017.

[17] I. Masi, A. T. Trn, T. Hassner, J. T. Leksut, and G. Medioni, "Do we really need to collect millions of faces for effective face recognition ?" in *ECCV*. Springer, 2016, pp. 579–596.

[18] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *ICCV*, 2017, pp. 2840–2848.

[19] S. Chen, C. Gong, J. Yang, X. Li, Y. Wei, and J. Li, "Adversarial metric learning," in *IJCAI-18*, 7 2018, pp. 2021–2027. [Online]. Available: https://doi.org/10.24963/ijcai.2018/279

[20] Z. Li, Y. Zhang, Y. Wei, Y. Wu, and Q. Yang, "End-to-end adversarial memory network for cross-domain sentiment classification." in *IJCAI*, 2017, pp. 2237–2243.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.

[22] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *null*. IEEE, 2006, pp. 1735–1742.

[23] M. T. Law, R. Urtasun, and R. S. Zemel, "Deep spectral clustering learning," in *ICML*. JMLR. org, 2017, pp. 1985–1994.

[24] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015, pp. 815–823.

[25] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, 2016, pp. 1857–1865.

[26] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, "Deep metric learning with angular loss," in *ICCV*, 2017, pp. 2593–2601.

[27] B. Harwood, B. Kumar, G. Carneiro, I. Reid, T. Drummond *et al.*, "Smart mining for deep metric learning," in *ICCV*, 2017, pp. 2821–2829.

[28] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *ICCV*, 2017, pp. 360–368.

[29] Y. Yuan, K. Yang, and C. Zhang, "Hard-aware deeply cascaded embedding," in *ICCV*, 2017, pp. 814–823.

[30] Y. Zhao, Z. Jin, G.-j. Qi, H. Lu, and X.-s. Hua, "An adversarial approach to hard triplet generation," in *ECCV*, 2018, pp. 501–517.

[31] Y. Duan, W. Zheng, X. Lin, J. Lu, and J. Zhou, "Deep adversarial metric learning," in *CVPR*, 2018, pp. 2780–2789.

[32] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *AISTATS*, 2011, pp. 315–323.

[33] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *ICLR*, 2018.

[34] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[35] P. Zadeh, R. Hosseini, and S. Sra, "Geometric mean metric learning," in *ICML*, 2016, pp. 2464–2471.

[36] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition." in *the IEEE, 86*, 1998, p. 2278–2324. [Online]. Available: http://ieeexplore.ieee.org/document/726791/

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

[38] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop*, vol. 2011, no. 2, 2011, p. 5.

[39] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[40] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010, pp. 807–814.

[41] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, 2007.

[42] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 554–561.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[44] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *TPAMI*, vol. 33, no. 1, pp. 117–128, 2011.

[45] P. Parveen, J. Evans, B. Thuraisingham, K. W. Hamlen, and L. Khan, "Insider threat detection using stream mining and graph mining," in *IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. IEEE, 2011, pp. 1102–1110.

[46] T. Al-Khateeb, M. M. Masud, L. Khan, C. Aggarwal, J. Han, and B. Thuraisingham, "Stream classification with recurring and novel class detection using class-based ensemble," in *IEEE 12th International Conference on Data Mining*. IEEE, 2012, pp. 31–40.