Adaptive Label Noise Cleaning with Meta-Supervision for Deep Face Recognition

Yaobin Zhang, Weihong Deng,[*] Yaoyao Zhong, Jiani Hu
Beijing University of Posts and Telecommunications
{zhangyaobin, whdeng, zhongyaoyao, jnhu}@bupt.edu.cn

Xian Li, Dongyue Zhao, Dongchao Wen
Canon Innovative Solution (Beijing) Co., Ltd
{lixian, zhaodongyue, wendongchao}@canon-is.com.cn

Abstract

The training of a deep face recognition system usually faces the interference of label noise in the training data. However, it is difficult to obtain a high-precision cleaning model to remove these noises. In this paper, we propose an adaptive label noise cleaning algorithm based on meta-learning for face recognition datasets, which can learn the distribution of the data to be cleaned and make automatic adjustments based on class differences. It first learns reliable cleaning knowledge from well-labeled noisy data, then gradually transfers it to the target data with meta-supervision to improve performance. A threshold adapter module is also proposed to address the drift problem in transfer learning methods. Extensive experiments clean two noisy in-the-wild face recognition datasets and show the effectiveness of the proposed method to reach state-of-the-art performance on the IJB-C face recognition benchmark.

1. Introduction

Deep face recognition depends heavily on the training data [57, 58, 59]. Due to the deficiencies in data collection and preprocessing, there is usually label noise in the dataset. For the face datasets, it refers to the existence of one to multiple faces of different people in one class. In recent years, increasing the data scale of face recognition datasets is proved essential for training deep models [6, 20, 24, 56, 60], but the label noise rate also inevitably improved [47]. Some studies [4, 9, 47, 48] reveal the heavy harm of label noise in the training sets to face recognition accuracy. This leads to a contradiction between data size and cleanliness, which gives birth to the data cleaning task. It goals to keep the face images of one person (noted as “signals”), delete the face images of other people (noted as “noise”), and keep as many images as possible in one class.

Many data cleaning solutions [1, 6, 47] are proposed to eliminate label noise. For example, FaceGraph [56] deploys a global-local Graph Convolutional Network (GCN) [21, 27] as a binary classifier to classify signal and noise on a k-NN graph. The biggest contradiction in these kinds of methods is that the target data to be cleaned is generally unlabeled, so the cleaning model is usually trained on additional labeled data. Assuming that the additional labeled data is the source domain, and the unlabeled target data is the target domain, due to the domain gap, the trained model is difficult to adapt to the distribution of the target data. In Figure 1(a), red triangles and rectangles represent the real signal-noise distribution of the target domain. When a cleaning model trained on the source domain is deployed to clean them, the signal and noise may not be separated well as the blue triangles and rectangles. To solve this problem, many transfer learning methods [18, 31, 32, 39, 44] are

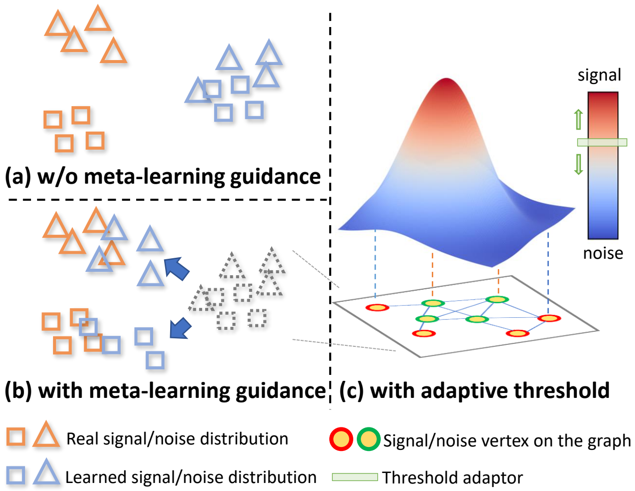

signal
noise
(a) w/o meta-learning guidance
(b) with meta-learning guidance
(c) with adaptive threshold
Real signal/noise distribution
Signal/noise vertex on the graph
Learned signal/noise distribution
Threshold adaptor

Figure 1: Main idea of AMC for face data cleaning. (a)(b) With meta-learning guidance, the learned signal-noise distribution is closer to the real distribution. (c) Learn adaptive threshold of one class on the signal-noise graph manifold.

proposed to eliminate the domain gap [49]. In this paper, we propose the Adaptive Meta Cleaner (AMC) framework, which is a novel transferring method for face data cleaning based on meta-learning [23, 45]. AMC treats the source domain as the meta-train set and the target domain as the meta-test set. Since the target domain is unlabeled, a graph-based unsupervised method is proposed to pseudo-label the target data inspired by some related work [35, 50, 55]. Noted that the signal-noise distribution of the pseudo label is also biased, it is only used for transferring cleaning knowledge instead of directly used for training the cleaning model. In this way, the model learns reliable knowledge from the source domain, and gradually transfers it to the target domain.

This meta-learning-based transferring approach will raise a new problem, *i.e.*, the drift of the decision boundary. The optimization target only measures the upper limit of the data distribution, which aims to predict the signals as close to 1 as possible and the noise as close to 0 as possible. Then it is a common practice to take an empirical boundary threshold value such as 0.5 [7]. Samples with a predicted value greater than the threshold are judged positive, and the ones smaller than the threshold are judged negative. However, in the transfer learning tasks, the model tries to learn decision boundary distribution that is fit for both the source and target set, but it is only expected to perform well on the target set. In this case, an empirical threshold may experience drift between different domains and cannot completely describe the boundary of the target domain. To solve this problem, an adaptive threshold learning method is proposed in AMC along with the meta-learning procedure to dynamically adjust boundary thresholds for different classes.

To verify AMC on real data, we clean two in-the-wild noisy face datasets CASIA-WebFace [54] and Million-Celebs [56], guided by the MS-Celeb-1M [20] dataset with a high-quality signal-noise label. The effectiveness is assessed in terms of the comparative recognition performance of Arcface [10] trained on the cleaned datasets. Results show that AMC effectively improves the face recognition performance compared with previous cleaning methods on face verification tests and the IJB-C benchmark [33]. The subsequent discussion also specifically analyzes the reasons for the performance improvement of the proposed method.

The main contributions can be summarized as follows:

- We explore the data cleaning task from a new perspective of domain gap, and provide one of the possible solutions for signals and noise distribution transferring for deep face recognition datasets, which can inspire more related discussions.
- We design the meta-learning-based AMC framework to clean label noise in face recognition datasets.
- Multiple datasets are cleaned and compared to test their performance limitations.

2. Related Work

Label Noise Cleaning. Label noise cleaning [3, 13, 14, 52] algorithms are widely used to address the label noise problem [17]. In the face recognition community, a lot of self-supervised cleaning methods are proposed by mining the inner correlation of data. CASIA-WebFace [54] cleans every subject guided by its "main photo". MegaFace2 [25] clusters images according to their pairwise distances. VGGFace [38] and VGGFace2 [7] train SVMs as cleaner. Celeb500k [6] and MCSM [53] train CNN-based label predictors to select samples in a bootstrapping manner. Some other work adopts more supervision to enhance cleaning accuracy. Some introduce human labors [1, 7, 38, 47], while others try to introduce cleaning knowledge from external data. FaceGraph [56] deploys a GCN model trained on a simulation set, and WebFace260M [61] deploys MS1M pretrained model as the first teacher to guide self-training. This paper introduces external data to cooperate with the target data to develop a high-quality label noise cleaner.

Meta Learning. Meta-learning [23, 28, 45, 46] is an efficient approach for the models to learn the learning ability, and is widely used for transferring knowledge across domains. MAML [15] and its variances [16, 37, 40] learn a good weight initialization for fast adaptation on a new task for the few-shot learning problem. MTL [43] learns scaling and shifting functions in the meta-learning process for transferring. MLNT [29] and MW-Net [42] learn from noisy labeled data guided by meta-learning. Some work [5, 8, 30, 62] applies meta-learning on the graph-based tasks. MFR [19] introduces meta-learning to the face recognition community to improve generalization ability from different domains. In this paper, the idea of meta-learning is used to transfer the cleaning knowledge learned on the source domain to the target domain.

3. Methodology

In this paper, we propose an automatic learning approach AMC to clean label noise in face recognition datasets. Consider an unlabeled face dataset, for instance, the images of celebrities returned by searching their names on the web search engine. These images are naturally divided into different classes by the searching results, but there may be not only the target celebrity in one class but also some other identities that are related to that celebrity. This leads to multiple identities in one class to make the dataset noisy. The cleaning task aims to select images belonging to one of the identities for each class to build up a noise-free dataset for downstream tasks. We assume that different classes of the original dataset are independent so that we can apply the proposed method to clean label noise for each class separately. The biggest difficulty is that there is usually no ground-truth label supervision for the signals and noise, and the manual labeling is time-consuming and inaccurate.

Let $\mathbb{T}$ represent the unlabeled target face dataset to be cleaned, in where there are n face samples in one class $\{(\boldsymbol{x}_i^t) \mid i \in \{1, 2, \cdots, n\}\}$. The cleaning task predicts labels $\{(\boldsymbol{y}_i^t) \mid i \in \{1, 2, \cdots, n\}\}$ for all n instances, where $\boldsymbol{y}_i^t \in \{0, 1\}$, 1 representing signals and 0 representing noise. Besides, a fully-labeled dataset $\mathbb{S}$ is introduced to help the cleaning of $\mathbb{T}$. In one class of $\mathbb{S}$ there are m face samples $\{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s) \mid i \in \{1, 2, \cdots, m\}\}$, where $\boldsymbol{y}_i^s \in \{0, 1\}$. As shown in Figure 2, AMC is proposed to address the label noise problem with three main modules: pseudo label generation (Section 3.1), meta-optimization (Section 3.2) and adaptive threshold adjustment (Section 3.3).

3.1. Unsupervised Pseudo Label Generation

This step is designed to generate feature embeddings for all data and then pseudo-label the target data. First, a CNN-based face recognition model $\mathbf{E}$ with parameters ν is trained with all $\boldsymbol{y}^s = 1$ labeled data in $\mathbb{S}$ to get the optimal ν^*. Then $\mathbf{E}$ is deployed as a feature extractor to extract d-dimensional feature embeddings for all images in $\mathbb{S}$ and $\mathbb{T}$, which are taken as the input of all subsequent steps. So the images can be represented as d-dimensional $l2$-normalized features, and one class in $\mathbb{T}$ is represented as a matrix

$$X^t \triangleq \left[\mathbf{E}\left(\boldsymbol{x}_1^t; \nu^*\right), \mathbf{E}\left(\boldsymbol{x}_2^t; \nu^*\right), \cdots, \mathbf{E}\left(\boldsymbol{x}_n^t; \nu^*\right)\right]^T \in \mathbb{R}^{n \times d} \tag{1}$$

and so is one class $X^s \in \mathbb{R}^{m \times d}$ in $\mathbb{S}$.

Based on the feature embeddings, an unsupervised module $\mathbf{G}$ is designed to pseudo-label the target data. $\mathbf{G}$ builds a graph $\mathcal{G} = (\mathcal{V}_\mathcal{G}, \mathcal{E}_\mathcal{G})$ for each class as input, where vertices

$$\mathcal{V}_\mathcal{G} = \{1, 2, \cdots, n\} \tag{2}$$

represent n image samples of the class, and edges

$$\mathcal{E}_\mathcal{G} = \{(i, j) \mid \forall i, j \in \mathcal{V}_\mathcal{G},\ S_{ij} > \lambda\} \tag{3}$$

where λ is a threshold hyper-parameter, and $S = X^t {X^t}^T$ is the $n \times n$ pairwise cosine similarity of feature matrix X^t. Then module $\mathbf{G}$ divides $\mathcal{G}$ into multiple connected subgraphs $\{\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \cdots, \mathcal{G}^{(K)}\}$ that satisfy

$$\begin{cases} \mathcal{V}_{\mathcal{G}^{(p)}} \cap \mathcal{V}_{\mathcal{G}^{(q)}} = \varnothing,\ \forall p, q \in \{1, 2, \cdots, K\} \text{ and } p \neq q \\ \displaystyle\bigcup_{p=1}^{K} \mathcal{V}_{\mathcal{G}^{(p)}} = \mathcal{V}_\mathcal{G} \end{cases} \tag{4}$$

All samples in the subgraph that contains the most vertices are pseudo-labeled as signals, while other samples are pseudo-labeled as noise. So the pseudo label of one class is represented as

$$\hat{Y}^t = \left[\hat{\boldsymbol{y}}_1^t, \hat{\boldsymbol{y}}_2^t, \cdots, \hat{\boldsymbol{y}}_n^t\right]^T \in \mathbb{R}^{n \times 1} \tag{5}$$

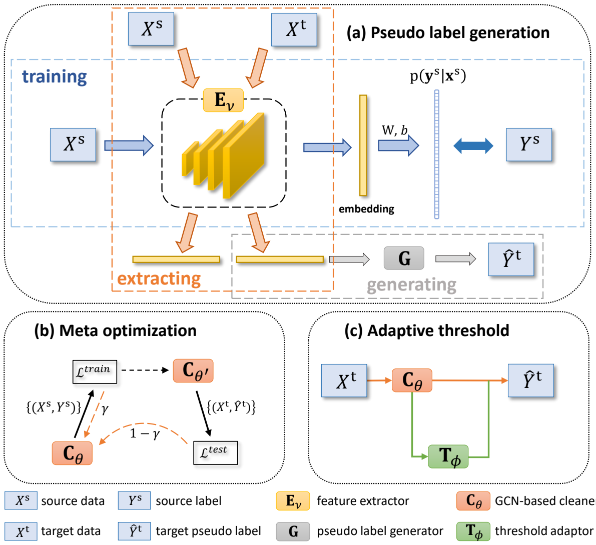


Figure 2: Overview of AMC. (a) Pre-train a face recognition model $\mathbf{E}$ with source data $\mathbb{S}$ as feature extractor, and pseudo-label the target data $\mathbb{T}$ with unsupervised label generator $\mathbf{G}$. (b) Meta-learning: using source data as meta-train set and pseudo-labeled target data as meta-test set to train the GCN cleaner $\mathbf{C}$. (c) A threshold adapter $\mathbf{T}$ helps clean target data to solve the problem of boundary drift.

$$\hat{\boldsymbol{y}}_i^t = \begin{cases} 1, & \text{if } i \in \mathcal{V}_{\mathcal{G}^{(p^*)}},\ p^* = \arg\max\limits_p \|\mathcal{V}_{\mathcal{G}^{(p)}}\| \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where $\|\mathcal{V}_{\mathcal{G}^{(p)}}\|$ means the number of vertices in graph $\mathcal{G}^{(p)}$.

3.2. Meta-Optimization

To solve the problem that a model trained with full supervision has a poor effect on unlabeled target data, and to make full use of the knowledge of the source data to clean the target data, we propose to train the cleaning model by meta-learning to bridge the difference in the data distribution of the source and target. The cleaner, denoted as $\mathbf{C}$, is a GCN-based multiple-layer binary vertex classification network, which takes the feature matrix X of a class and a k-NN graph $\mathcal{G}_X$ built by X as input. Following a recent GCN-based cleaning method [56], we use the same forward propagation function to implement cleaner $\mathbf{C}$. Please refer to the supplementary materials for the detailed network structure. Symbolic, the cleaner can be expressed as

$$P = \sigma\left(\mathbf{C}\left(X, \mathcal{G}_X; \theta\right)\right) \tag{7}$$

where θ is the parameters of $\mathbf{C}$, and σ is the sigmoid activation function that outputs prediction scores $P = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_n]^T \in \mathbb{R}^{n \times 1}$ for all n samples of the class, where each element $\boldsymbol{p}_i \in (0, 1)$.

Meta-Train. In meta-train phase, the labeled dataset $\mathbb{S}$ is used to train $\mathbf{C}$ with full supervision. For a class of m samples, the meta-train loss function is formulated as the mean of the binary cross-entropy loss of all samples:

$$\mathcal{L}^{\text{train}} = -\frac{1}{m}\sum_{i=1}^{m}\left[\boldsymbol{y}_i^s \cdot \log \boldsymbol{p}_i^s + (1-\boldsymbol{y}_i^s) \cdot \log\left(1-\boldsymbol{p}_i^s\right)\right] \tag{8}$$

where $\boldsymbol{p}_i^s$ is the network output score of the i-th sample between 0 and 1, and $\boldsymbol{y}_i^s \in \{0,1\}$ is the label of the i-th sample. In back-propagation, Stochastic Gradient Descent (SGD) is used to update the network parameters θ. For a graph mini-batch of batch size B, the updating principle is formulated as:

$$\theta' = \theta - \alpha\frac{1}{B}\sum_{b=1}^{B}\nabla_\theta \mathcal{L}_b^{\text{train}}(\theta) \tag{9}$$

where α is the meta-learning rate.

Meta-Test. In the meta-test phase, the performance of cleaner $\mathbf{C}$ with updated parameters θ' is tested on the pseudo-labeled data $\mathbb{T}$ with meta-test loss function in the same form as the meta-train loss. For a class with n samples, the meta-test binary cross-entropy loss is

$$\mathcal{L}^{\text{test}} = -\frac{1}{n}\sum_{i=1}^{n}\left[\hat{\boldsymbol{y}}_i^t \cdot \log \boldsymbol{p}_i^t + \left(1-\hat{\boldsymbol{y}}_i^t\right) \cdot \log\left(1-\boldsymbol{p}_i^t\right)\right] \tag{10}$$

where $\boldsymbol{p}_i^t$ is the network output score of the i-th samples between 0 and 1, and $\hat{\boldsymbol{y}}_i^t \in \{0,1\}$ is the pseudo label of the i-th sample. Especially, in order to avoid model overfitting to the biased pseudo label, the pseudo label of each sample is randomly dropped out with the probability of p.

Meta-Update. Combining the meta-train and meta-test loss, the final meta-learning loss function is designed as

$$\mathcal{L}^{\text{meta}} = \gamma\mathcal{L}^{\text{train}}(\theta) + (1-\gamma)\,\mathcal{L}^{\text{test}}(\theta') \tag{11}$$

where γ balances meta-train and meta-test. Therefore, in one step, parameter θ is updated by

$$\begin{aligned}\theta \leftarrow \quad & \theta - \tfrac{\gamma\cdot\alpha}{B}\sum_{b=1}^{B}\frac{\partial\mathcal{L}_b^{\text{train}}(\theta)}{\partial\theta} - \tfrac{(1-\gamma)\cdot\alpha}{B}\sum_{b=1}^{B}\frac{\partial\mathcal{L}_b^{\text{test}}(\theta')}{\partial\theta'} \\ & + \tfrac{(1-\gamma)\cdot\alpha^2}{B^2}\sum_{b=1}^{B}\frac{\partial^2\mathcal{L}_b^{\text{train}}(\theta)}{\partial\theta^2}\sum_{b=1}^{B}\frac{\partial\mathcal{L}_b^{\text{test}}(\theta')}{\partial\theta'}\end{aligned} \tag{12}$$

The meta-learning method is equivalent to gradient descent on the meta-train and meta-test sets, and applying high-order regularities to correct the two domains. In this way, the model tries to optimize the source and target data simultaneously to perform well on the two domains. Please refer to the supplementary materials for detailed derivation.

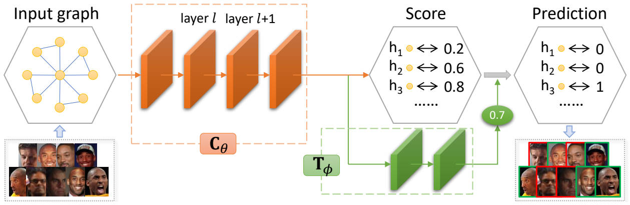


Figure 3: The forward propagation framework with adaptive threshold. $\mathbf{C}$ takes a graph built from a class as input and predicts scores for all samples. $\mathbf{T}$ takes the unprobabilized output of $\mathbf{C}$ as input and outputs the threshold $\boldsymbol{t}$.

3.3. Adaptive Threshold

There is still the boundary drift problem in the cleaner as illustrated in Section 1. In fact, as a discriminating model, the cleaner learns posterior probability $p\left(Y^t|X^t,\boldsymbol{t}\right)$ for class X^t, and the prediction label Y^t is determined as

$$Y^t = \left[\boldsymbol{y}_1^t, \boldsymbol{y}_2^t, \cdots, \boldsymbol{y}_n^t\right]^T \in \mathbb{R}^{n\times 1} \tag{13}$$

$$\boldsymbol{y}_i^t = \begin{cases} 1, & \text{if } \boldsymbol{p}_i^t > \boldsymbol{t} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where $\boldsymbol{t} \in (0,1)$ is the decision boundary distribution. Instead of fixing $\boldsymbol{t}$ to 0.5 as in many binary classification tasks, we propose an adaptive threshold learning method to explicitly learn the decision boundary for different classes. As shown in Figure 3, a threshold adapter network $\mathbf{T}$ with parameters ϕ is designed along with the cleaner $\mathbf{C}$. In forward propagation, $\mathbf{T}$ takes the unprobabilized output $\mathbf{C}\left(X^t, \mathcal{G}_{X^t};\theta\right) \in \mathbb{R}^{n\times 1}$ of the cleaner for a graph in the target domain as input, and outputs the threshold $\boldsymbol{t}$ which is normalized by the sigmoid function:

$$\boldsymbol{t} = \mathbf{T}\left(\mathbf{C}\left(X^t, \mathcal{G}_{X^t};\theta\right);\phi\right) \tag{15}$$

To efficiently update parameters ϕ, a threshold-aware loss function is designed. For a graph with n vertices, the adaptive threshold loss is formulated as

$$\begin{aligned}\mathcal{L}^{\text{th}} = & -\frac{1}{n}\sum_{i=1}^{n}\Big[\hat{\boldsymbol{y}}_i^t \cdot \log\Big(1 - \left[1 - \boldsymbol{p}_i^t - \left[1 - \boldsymbol{t} - m_{\text{fn}}\right]_+\right]_+\Big) \\ & + \left(1-\hat{\boldsymbol{y}}_i^t\right) \cdot \log\Big(1 - \Big[\boldsymbol{p}_i^t - \left[\boldsymbol{t} - m_{\text{fp}}\right]_+\Big]_+\Big)\Big]\end{aligned} \tag{16}$$

where $[\cdot]_+$ means $\max(\cdot, 0)$, m_{fn} and m_{fp} are the margins for positive and negative samples. Compared with the Mean Square Error (MSE) loss, this form of cross-entropy provides more effective gradients. In back-propagation, gradients propagate through $\boldsymbol{t}$ to ϕ. The implementation of $\mathbf{T}$ is to first average the input, and then pass through a fully

Algorithm 1 Adaptive Meta Cleaner.

```
Require: labeled data $\mathbb{S} = \{(X^s, Y^s)\}$, unlabeled data $\mathbb{T} = \{(X^t)\}$, feature extractor $\mathbf{E}_\nu$, GCN cleaner $\mathbf{C}_\theta$, threshold adaptor $\mathbf{T}_\phi$, unsupervised pseudo label generator $\mathbf{G}$, number of iterations $I$, batch size $B$, hyper-parameters $\lambda$, $p$, $\gamma$, $m_{\text{fp}}$, $m_{\text{fn}}$
Ensure: optimal parameters $\nu$, $\theta$, $\phi$, predicted label $Y^t$.
  • Initialize $\nu$, $\theta$ and $\phi$.
  • Find optimized parameters $\nu^*$ on data $\mathbb{S}$.
  • Generate pseudo label $\hat{Y}^t$ for $\mathbb{T}$ with $\mathbf{G}$ by Eq.5 and Eq.6.
  for $i = 1, \cdots, I$ do
    if $i \bmod 2$ then
      • Randomly select $B$ samples from set $\mathbb{S}$ to get the input meta-train mini-batch $\mathbf{B}^s$.
      • Randomly select $B$ samples from set $\mathbb{T}$ to get the input meta-test mini-batch $\mathbf{B}^t$.
      • Meta-train: calculate $\theta'$ by Eq.9.
      • Meta-update: update $\theta$ by Eq.12.
    else
      • Randomly select $B$ samples from set $\mathbb{T}$ to get the input mini-batch $\mathbf{B}^t$.
      • Update $\phi$ by the adaptive threshold loss $\mathcal{L}^{\text{th}}$.
    end if
  end for
  • Predict label $Y_t$ by Eq.13 and Eq.14.
```

connected layer activated by the sigmoid function to get the predicted threshold between 0 and 1.

3.4. Summary

The whole training procedure for AMC is summarized in Algorithm 1. A face feature extractor $\mathbf{E}$ is trained on $\mathbb{S}$, and $\mathbb{T}$ is pseudo-labeled by $\mathbf{G}$. Then the meta-optimization step and adaptive threshold learning step are carried out in turn until convergence. Finally, the optimized cleaner $\mathbf{C}$ and threshold adapter $\mathbf{T}$ are used to predict labels for $\mathbb{T}$.

4. Experiments

4.1. Experimental Setup

Evaluation Metrics Data cleaning performance is evaluated by training deep face recognition models with the cleaned datasets. The verification set CFP-CP [41] is used to test cross-pose recognition accuracy, and AgeDB [34] is used to test cross-age recognition accuracy. IJB-B [51] and IJB-C [33] benchmarks are used to evaluate template-wise face recognition performance by measuring True Positive Rate (TPR) at given False Positive Rate (FPR) and Rank-1 retrieval accuracy. MegaFace Challenge 1 [25] tests large-scale face recognition performance under 1M distractors.

Datasets	# photos	# subjects	Noise-Free
MS-Retina [11]	5.2M	93K	✓
MS-Celeb-1M [20]	7.5M	100K	×
CASIA-WebFace [54]	0.5M	10K	×
MillionCelebs [56]	87.0M	1M	×

Table 1: Face recognition datasets used in the experiments.

Implementation Details Table 1 shows face training sets used in the experiments. We randomly select 1,000 classes from each dataset to train the cleaner. To guarantee the representation reliability, feature extractor $\mathbf{E}$ is implemented as a ResNet-100 [22] Arcface [10] model that outputs $d = 512$ dimensional feature embeddings. 3-NN graphs are built with self-loop on all nodes as the input graph. The cleaner $\mathbf{C}$ is designed as a 5-layer GCN with 256-dimensional hidden features. Adam [26] is used as the meta-optimizer with the learning rate 0.001, weight decay 0.0005 and graph-batch size $B = 50$. Hyper-parameters $\lambda = 0.6$, $p = 0.9$, $\gamma = 0.6$, and the margins m_{fn} and m_{fp} are set 0.3 and 0.0, respectively. Since the difference in the number of classes can significantly affect recognition accuracy, for a fair comparison, we treat the pseudo label as output label if no signal is output by the cleaner. For face recognition training, SGD is used as the optimizer with the initial learning rate 0.1, weight decay 0.0005 and batch size 512. The learning rate is divided three times by 0.1 when the loss value does not decrease. Input images are aligned, resized to 112×112, and normalized by subtracting 127.5 and divided by 128.

4.2. Experiments on CASIA-WebFace

In this section, we clean the widely-used CASIA-WebFace [54] dataset and compare face recognition accuracy of ResNet-34 [22] ArcFace [10] model trained on different cleaned data. The noise rate of CASIA-WebFace is estimated 9.3-13.0% [47]. Lines 1 to 5 in Table 2 show the baseline performance of different cleaning methods. Comparative methods include the original dataset, a manually cleaned version [2], VGG [7] cleaning method that 1-vs-n SVMs are trained as classifiers, MF2 [35] cleaning method that selects signals in a graph according to the average pair-wise distance, and FaceGraph [56] that cleans one class in a graph with a global-local GCN. An ablation study is made with four setups: trained on source data (Source), pre-trained on source data and fine-tuned on target data (Fine-Tune), meta-learning method with threshold 0.5 (Meta), and meta-learning method with the threshold adapter $\mathbf{T}$ (AMC).

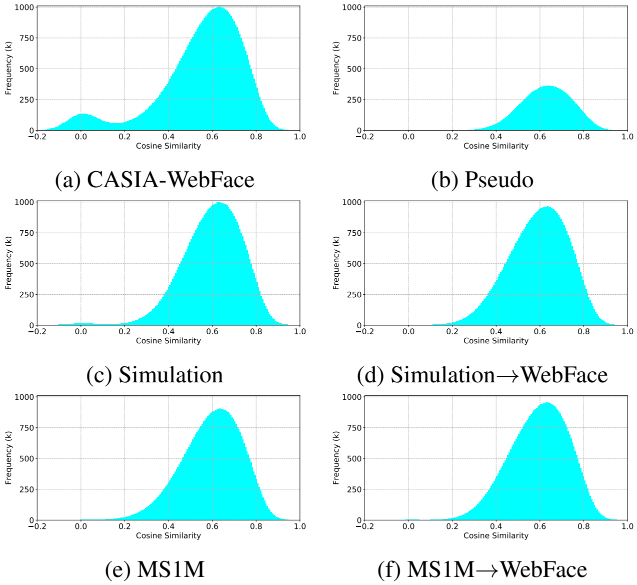
We take the noisy WebFace dataset as the target domain and select a labeled set as the source domain. Two kinds of source data are compared, one is simulated and the other is real. First, a simulation set is built based on the MS-Retina [11] dataset, which is a cleaned version of MS-Celeb-1M [20]: Assuming that it is a noise-free set, we

Domain	Methods	#	IJB-B (%)				IJB-C (%)				CFP-FP (%)	AgeDB (%)
			1e-5	1e-4	1e-3	Rank1	1e-5	1e-4	1e-3	Rank1		
WebFace [54]	-	1	58.88	75.96	86.41	86.68	68.04	80.71	89.34	88.21	94.73	93.83
	Manual [2]	2	59.90	75.36	86.01	86.79	69.00	80.79	89.08	88.17	94.11	93.63
	MF2 [35]	3	64.09	77.21	87.05	87.57	70.91	81.40	90.03	89.38	94.33	94.00
	VGG [7]	4	57.54	76.08	86.49	86.82	67.32	80.50	89.54	88.29	94.66	94.03
	FaceGraph [56]	5	61.93	77.58	87.95	88.40	72.74	82.71	90.74	90.19	95.20	94.23
Simulation ↓ WebFace	Source	6	63.27	77.14	86.81	87.55	71.63	81.70	90.12	88.99	94.59	93.90
	Fine-Tune	7	64.64	77.59	87.17	87.83	72.49	82.15	90.15	89.51	94.72	94.07
	Meta	8	64.28	77.99	87.49	88.27	71.93	82.77	90.36	89.95	94.88	**94.09**
	AMC	9	**65.98**	**78.74**	**87.76**	**88.55**	**73.59**	**82.94**	**90.40**	**90.06**	**95.03**	93.85
MS1M ↓ WebFace	Source	10	62.84	77.59	87.53	88.22	71.63	81.95	90.10	89.47	94.77	94.08
	Fine-Tune	11	64.27	77.66	87.26	88.27	72.69	82.40	90.30	89.59	94.71	94.27
	Meta	12	64.82	78.30	87.73	88.25	72.42	82.76	90.56	89.76	94.73	94.40
	AMC	13	**65.88**	**79.04**	**88.07**	**89.05**	**73.78**	**83.02**	**90.87**	**90.52**	**94.77**	**94.42**

Table 2: Train ResNet-34 [22] deep face recognition models by Arcface [10] with cleaned CASIA-WebFace [54] dataset.

select half of its classes as the base set, then gradually replace its images with randomly selected images from the other half as noise until noise rate reaches the same level of the real set MS-Celeb-1M. Lines 6 to 9 in Table 2 show that using simulation set to clean WebFace [54] reaches significantly higher face recognition accuracy than baseline results. For instance, on the IJB-C benchmark [33], the dataset cleaned by AMC outperforms the origin WebFace [54] by 5.55% and the previous state-of-the-art FaceGraph [56] by 0.85% to reach 73.59% TPR at 1e-5 FPR. From the ablation experiment, the cleaner trained on source data or fine-tuned on target data has achieved remarkable cleaning performance. The proposed meta-learning (#8) and adaptive threshold (#9) algorithms further enhance the recognition accuracy step by step, which proves that meta-learning-based transferring methods can effectively deal with the noisy label issue, and the proposed adapter is an effective solution to the boundary drift problem.

Noting the possible underrepresentation problem in the simulation set, we also propose to deploy a real in-the-wild set as the source domain for training. MS-Celeb-1M [20] is used because it contains abundant information about a variety of noise conditions, and there are a lot of cleaned versions on the web [1, 10, 11, 12]. The MS-Retina cleaned version is selected to label the signal and noise: for any image in MS-Celeb-1M, if it is also contained in MS-Retina, it is label as a signal, otherwise, it is labeled as noise. Lines 10 to 13 in Table 2 show the cleaning performance. It is observed that the real set significantly surpasses the simulation set on training the cleaner with AMC, especially on the IJB-C benchmark. It reaches 73.78% TPR at 1e-5 FPR to outperform the simulation set by 0.19%. This illustrates that the signal-noise distribution of the real set can provide more cleaning knowledge and the proposed AMC method successfully transferring it to the target data.



(a) CASIA-WebFace

(b) Pseudo

(c) Simulation

(d) Simulation→WebFace

(e) MS1M

(f) MS1M→WebFace

Figure 4: Histogram of pairwise intra-class similarity.

4.3. Experiments on MillionCelebs

The previous experiment shows that well-labeled data has good potential to transfer cleaning knowledge to an unlabeled one by the proposed AMC method. In this section, we continue to use MS-Celeb-1M [20] as the source data to clean a more challenging dataset, MillionCelebs [56], which is larger and dirtier. Due to the large scale of it, we randomly select 100,000 classes and remove obvious noise to build a subset named MC-mini for quick comparison. Noted that MillionCelebs [56] concludes the identities in MS1M [20], for a fair comparison, the identities selected to build MC-mini are excluded from MS1M. This dataset to be

Domain	Methods	#	IJB-B (%)				IJB-C (%)				CFP-FP (%)	AgeDB (%)
			1e-5	1e-4	1e-3	Rank1	1e-5	1e-4	1e-3	Rank1		
MC-mini [56]	-	1	84.06	92.03	95.51	94.24	90.57	94.31	96.85	95.75	96.27	96.77
	MF2 [35]	2	87.23	92.93	95.95	94.37	92.27	94.96	97.08	95.98	96.20	97.08
	VGG [7]	3	86.34	92.66	95.88	94.47	91.86	94.93	97.06	96.02	96.64	96.90
	FaceGraph [56]	4	87.04	92.78	95.82	94.43	91.92	95.04	97.05	96.00	96.34	96.97
MS1M ↓ MC-mini	Source	5	87.43	92.87	95.84	94.47	92.33	95.10	97.04	96.06	96.23	97.08
	Fine-Tune	6	87.20	92.71	95.91	94.42	91.95	94.94	97.04	95.96	96.40	97.20
	Meta	7	87.43	93.04	95.83	94.40	92.14	94.98	96.95	95.87	96.39	96.75
	AMC	8	**87.47**	**93.13**	**95.96**	**94.63**	**92.36**	**95.27**	**97.13**	**96.16**	**96.53**	**97.25**

Table 3: Train ResNet-50 [22] deep face recognition models by Arcface [10] with cleaned MC-mini [54] dataset.

Method	Id.(%)	Ver.(%)
CASIA-WebFace [54]	89.84	91.59
VGGFace2 [7]	88.69	92.72
MS1M-IBUG [12]	95.56	96.33
MC-mini [35]	95.95	97.19
WebFace - FaceGraph [56]	90.04	92.50
MC-mini -FaceGraph [56]	96.16	97.76
WebFace - AMC	90.34	92.83
MC-mini - AMC	**96.22**	**98.15**

Table 4: Verification TPR (@FPR=1e-6) and identification Rank-1 on the MegaFace Challenge 1 [25]. “MC-mini - X” means MC-mini dataset cleaned by method “X”.

cleaned has many more classes than CASIA-WebFace [54] and is noisier as well, which can test the robustness of the proposed method in large-scale cleaning.

Table 3 compares face recognition performances of ResNet-50 [22] ArcFace [10] model trained on the original and cleaned datasets. Like cleaning WebFace, the proposed method achieves a significant recognition performance improvement on the larger-scale MC-mini. Using MS1M [20] as source data, AMC fully surpasses the previous state-of-the-art FaceGraph and all other comparative methods to reach 92.36% and 95.27% TPR at 1e-5 and 1e-4 FPR and 96.16% Rank-1 performance on the IJB-C benchmark. Table 4 shows TPR at 1e-6 FPR verification and Rank-1 identification performance of ResNet-50 [22] ArcFace [10] model on the MegaFace Challenge 1 [25] adopting FaceScrub [36] as probe set and using the wash list provided by DeepInsight [10]. The CASIA-WebFace and MC-mini datasets cleaned by AMC reach the highest accuracy, outperforming other methods by a large margin. When cleaning the MC-mini dataset, AMC is 0.39% higher than the previous state-of-the-art FaceGraph [56] to reach 98.15% verification accuracy. It also performs better than many public datasets like VGGFace2 [7] and MS1M-IBUG [12], which fully proves the effectiveness of the proposed AMC method on large-scale recognition.

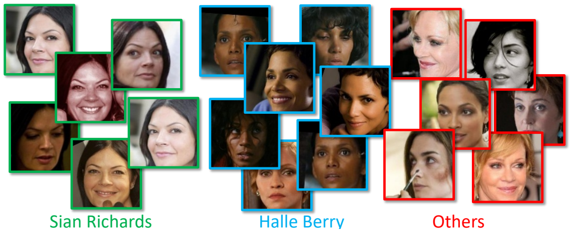
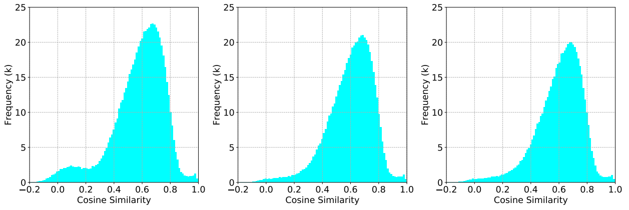


Figure 5: There are two main groups in class “1075644” of CASIA-WebFace dataset. Cleaner trained on real set manages to select one, but the one trained on simulation set fails.

4.4. Discussion

Comparing with the GCN-based state-of-the-art cleaning method FaceGraph [56], there are three main developments in AMC: 1) A real noisy set is deployed as source data instead of a simulation one. 2) Meta-learning is used to transfer cleaning knowledge. 3) A threshold adapter is proposed to deal with the boundary drift problem. These three aspects are analyzed and discussed in detail below.

Source Data. Figure 4 compares the intra-class pairwise cosine similarity histogram of CASIA-WebFace [54]. There are two main peaks before cleaning (4a), which are at similarity 0.0 and 0.6. The former is obviously caused by the label noise. The pseudo label (4b) can accurately refuse noise, however, many signals are also deleted. Comparing the simulation set and real set cleaned versions, it is observed that the simulation cleaned dataset still has a small noise peak at around similarity 0.0, but the real set cleaned version almost eliminates all noise conditions to reserve one main signal peak while keeping the recall. We track some classes and find that the “multi-modal” phenomenon is the main reason that causes the fail of cleaning: in one noisy class, there may be multiple main face groups. As shown in Figure 5, the images belong to one class “1075644” in CASIA-WebFace [54]. The cleaner trained on the simulation set selects images with both green and blue rectangles as signals, but they actually belong to two identities. The reason for this error is that noise data in the simulation set



(a) $\gamma = 0$ (b) $\gamma = 0.6$ (c) $\gamma = 1$

Figure 6: Intra-class similarity histogram with parameter γ.

are randomly added so that no multi-modal knowledge is contained for training, and the cleaner tends to accept all the main modes. Inversely, the cleaning model trained on the real noisy data manages to pick the green rectangles as signals. This shows that a real noise set contains more noise information that cannot be learned from a simulated one.

Balance Term γ. γ is used for balancing meta-train and meta-test loss in the training procedure. Figure 6 shows the change of intra-class similarity histogram of cleaned MC-mini with γ. In Figure 6a, $\gamma = 0$ means the meta-update of the cleaner only relies on the gradients from the pseudo-labeled target set, which leads to two peaks in the histogram near 0.1 and 0.7. This shows that the pseudo label cannot correctly guide the training of the cleaner. The model is overfitted to the biased pseudo distribution, so the precision of cleaning decreases. In Figure 6c, $\gamma = 1$ is equivalent to pre-training the cleaner with the meta-train set, and no meta-information is used. It is observed that the precision of cleaning is greatly improved, which once again proves the authenticity and reliability of real source data. However, the peak value of the histogram drops significantly, which means a lot of signal samples are also deleted, leading to a lower cleaning recall. After comparison experiments, we set γ to 0.6 to clean MC-mini as shown in Figure 6b. On the one hand, the cleaning result maintains the same accuracy as directly training with the source data. On the other hand, it recalls more signals under the guidance of the pseudo-labeled target data. Therefore, with the γ-reconciled meta-learning method, the cleaner completes the transferring task while maintaining excellent cleaning performance by high precision and recall.

Threshold Adapter. In the proposed method, a threshold adapter $\mathbf{T}$ is deployed in AMC to judge the threshold boundary for the given class. To explore how the adapter is better than the hand-designed threshold, Figure 7 visualizes the mapping between the mean of its input values and its output threshold value with curves. The scattered points around the curves record the mapping between the mean value of randomly selected classes from the target domain and the ideal threshold that makes the class reach the minimum misclassification rate. Red and green represent experiment #9 and #13 in Table 2, and blue represents experiment #8 in Table 3. It is observed from the scatter points that the mean input values and the ideal threshold values are positively correlated, and the adaptor manages to fit this correlation in the form of Sigmoid as the three curves. For instance, the mean input of MC-mini is mostly lower than that of WebFace due to its big noise, and its distribution is more concentrated, so the adapter learns a smaller effective input interval in both value and range. Table 5 compares the Pearson correlation coefficient between the mean input and ideal threshold, and the variance of the differences between the ideal and output predicted threshold. Training with the real set significantly improves the correlation coefficient. The stronger the correlation, the more obvious the effect of the adapter to reduce the variance, which means the model better fits the distribution of the signal-noise boundary of the target domain to deal with the drift problem.

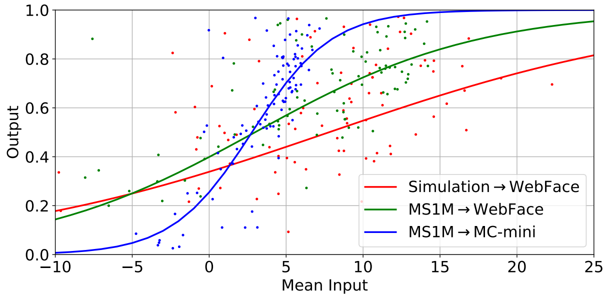


Figure 7: Map of the mean value of the input of adapter and its corresponding output threshold.

Methods	Correlation	Variance ($\times$1e−2)	
		with $\mathbf{T}$	without $\mathbf{T}$
Simulation→WebFace	0.42	3.635	4.458
MS1M→WebFace	0.60	2.315	4.028
MS1M→MC-mini	0.77	2.609	6.204

Table 5: The Pearson correlation coefficient between the mean input and ideal threshold, and the variance of the differences between the ideal and output threshold.

5. Conclusion

In this paper, we propose a novel meta-learning-based label noise cleaning method AMC, which can effectively learn useful cleaning knowledge from source data, and transfer it to the target data. In the experiments, AMC is deployed to clean two noisy face recognition datasets to show that training face recognition models with the dataset cleaned by AMC can reach better recognition performance on many benchmarks than existing cleaning methods.

Acknowledgments This work was supported by Canon Innovation Solution (Beijing) Co., Ltd. under Grant No. OLA21011.

References

[1] Challenge 3: Face feature test/trillion pairs. trillionpairs.deepglint.com.

[2] Github: happynear/faceverification. http://github.com/happynear/FaceVerification/.

[3] Anelia Angelova, Yaser Abu-Mostafam, and Pietro Perona. Pruning training sets for learning of object categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 494–501. IEEE, 2005.

[4] Ankan Bansal, Carlos Castillo, Rajeev Ranjan, and Rama Chellappa. The do's and don'ts for cnn-based face verification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2545–2554, 2017.

[5] Avishek Joey Bose, Ankit Jain, Piero Molino, and William L Hamilton. Meta-graph: Few shot link prediction via meta learning. *arXiv preprint arXiv:1912.09867*, 2019.

[6] Jiajiong Cao, Yingming Li, and Zhongfei Zhang. Celeb-500k: A large training dataset for face recognition. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2406–2410. IEEE, 2018.

[7] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *Automatic Face &amp; Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 67–74. IEEE, 2018.

[8] Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. Meta relational learning for few-shot link prediction in knowledge graphs. *arXiv preprint arXiv:1909.01515*, 2019.

[9] Jiankang Deng, Jia Guo, Tongliang Liu, Mingming Gong, and Stefanos Zafeiriou. Sub-center arcface: Boosting face recognition by large-scale noisy web faces.

[10] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[11] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*, 2019.

[12] Jiankang Deng, Yuxiang Zhou, and Stefanos Zafeiriou. Marginal loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 60–68, 2017.

[13] Weiwei Du and Kiichi Urahama. Error-correcting semi-supervised learning with mode-filter on graphs. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 2095–2100. IEEE, 2009.

[14] Weiwei Du and Kiichi Urahama. Error-correcting semi-supervised pattern recognition with mode filter on graphs. In *2010 2nd International Symposium on Aware Computing*, pages 6–11. IEEE, 2010.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[16] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9516–9527, 2018.

[17] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[18] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[19] Jianzhu Guo, Xiangyu Zhu, Chenxu Zhao, Dong Cao, Zhen Lei, and Stan Z Li. Learning meta face recognition in unseen domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6172, 2020.

[20] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pages 87–102. Springer, 2016.

[21] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[23] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

[24] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[25] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016.

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[27] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[28] Christiane Lemke, Marcin Budka, and Bogdan Gabrys. Metalearning: a survey of trends and technologies. *Artificial intelligence review*, 44(1):117–130, 2015.

[29] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5051–5059, 2019.

[30] Lu Liu, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. Learning to propagate for graph meta-

learning. In *Advances in Neural Information Processing Systems*, pages 1039–1050, 2019.

[31] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[32] Zimeng Luo, Jiani Hu, Weihong Deng, and Haifeng Shen. Deep unsupervised domain adaptation for face recognition. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 453–457. IEEE, 2018.

[33] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, pages 158–165. IEEE, 2018.

[34] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: The first manually collected, in-the-wild age database. In *Computer Vision and Pattern Recognition Workshops*, pages 1997–2005, 2017.

[35] Aaron Nech and Ira Kemelmacher-Shlizerman. Level playing field for million scale face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3406–3415. IEEE, 2017.

[36] Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *2014 IEEE international conference on image processing (ICIP)*, pages 343–347. IEEE, 2014.

[37] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

[38] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *bmvc*, volume 1, page 6, 2015.

[39] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766, 2007.

[40] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, pages 113–124, 2019.

[41] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[42] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, pages 1919–1930, 2019.

[43] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2019.

[44] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[45] Joaquin Vanschoren. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*, 2018.

[46] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

[47] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. In *European Conference on Computer Vision*, pages 780–795. Springer, 2018.

[48] Mei Wang and Weihong Deng. Deep face recognition: A survey. *arXiv preprint arXiv:1804.06655*, 2018.

[49] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[50] Mei Wang, Weihong Deng, Jiani Hu, Xunqiang Tao, and Yaohai Huang. Racial faces in the wild: Reducing racial bias by information maximization adaptation network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 692–702, 2019.

[51] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. Iarpa janus benchmark-b face dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 90–98, 2017.

[52] D Randall Wilson and Tony R Martinez. Instance pruning techniques. In *ICML*, volume 97, pages 400–411, 1997.

[53] Yan Xu, Yu Cheng, Jian Zhao, Zhecan Wang, Lin Xiong, Karlekar Jayashree, Hajime Tamura, Tomoyuki Kagaya, Shengmei Shen, Sugiri Pranata, et al. High performance large scale face recognition with multi-cognition softmax and feature retrieval. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1898–1906, 2017.

[54] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[55] Xiaohang Zhan, Ziwei Liu, Junjie Yan, Dahua Lin, and Chen Change Loy. Consensus-driven propagation in massive unlabeled data for face recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 568–583, 2018.

[56] Yaobin Zhang, Weihong Deng, Mei Wang, Jiani Hu, Xian Li, Dongyue Zhao, and Dongchao Wen. Global-local gcn: Large-scale label noise cleansing for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7731–7740, 2020.

[57] Jian Zhao, Jianshu Li, Xiaoguang Tu, Fang Zhao, Yuan Xin, Junliang Xing, Hengzhu Liu, Shuicheng Yan, and Jiashi Feng. Multi-prototype networks for unconstrained set-based face recognition. *arXiv preprint arXiv:1902.04755*, 2019.

[58] Jian Zhao, Lin Xiong, Yu Cheng, Yi Cheng, Jianshu Li, Li Zhou, Yan Xu, Jayashree Karlekar, Sugiri Pranata, Shengmei Shen, et al. 3d-aided deep pose-invariant face recognition. In *IJCAI*, volume 2, page 11, 2018.

[59] Jian Zhao, Lin Xiong, Jianshu Li, Junliang Xing, Shuicheng Yan, and Jiashi Feng. 3d-aided dual-agent gans for unconstrained face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2380–2394, 2018.

[60] Erjin Zhou, Zhimin Cao, and Qi Yin. Naive-deep face recognition: Touching the limit of lfw benchmark or not? *arXiv preprint arXiv:1501.04690*, 2015.

[61] Zheng Zhu, Guan Huang, Jiankang Deng, Yun Ye, Junjie Huang, Xinze Chen, Jiagang Zhu, Tian Yang, Jiwen Lu, Dalong Du, et al. Webface260m: A benchmark unveiling the power of million-scale deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10492–10502, 2021.

[62] Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. *arXiv preprint arXiv:1902.08412*, 2019.