Unsupervised Domain Adaptation for Fine-grained Deepfake ~~Classification~~

IEEE Publication Technology, *Staff, IEEE,*

***Abstract*—Deepfake represented by face swapping and face reenactment can transfer the appearance and behavioral expressions of a face in one video image to another face in a different video. In recent years, with the advancement of deep learning techniques ~~such as convolutional neural networks and generative adversarial models~~, deepfake technology has developed rapidly, achieving increasingly realistic effects. Therefore, many researchers have begun to focus on deepfake detection research. Currently, research on deepfake detection is mainly limited to binary classification of real and fake images, rather than identifying different ~~fake~~ methods~~. However, the recognition of different fake methods is of great significance for improving the accuracy, robustness, and privacy protection of deepfake detection in open-scene face image applications~~. In this paper, we propose an unsupervised ~~domain-adaptive fine-grained deepfake recognition~~ method for unknown deepfake types ~~that are commonly present in open-scene environments.~~ ~~This~~ method first uses labeled data from the source domain for model pre-training to establish the ability to recognize different fake methods in the source domain. Then, the method uses adaptive clustering ~~based on network memory effects~~ to cluster unlabeled images in the target domain, and designs a pseudo-label generator ~~for the target domain~~ to match the ~~adaptive~~ clustering results with the known deepfake types in the source domain. Finally, ~~in the model retraining stage, the pre-trained model is retrained~~ using labeled data ~~from~~ the source domain and pseudo-labeled data ~~from~~ the target domain to effectively identify unknown deepfake methods ~~in the target domain~~. We validate the effectiveness of ~~our~~ proposed method ~~on~~ three deepfake datasets~~:~~ ForgerNet [1], FaceForensics++ [2], and FakeAVCeleb [3]~~, and compare it with state-of-the-art methods~~. Experimental results show that our method has better domain generalization ability than the state-of-the-art methods~~.~~**

***Index Terms*—Deepfake detection, ~~D~~omain adaptation, ~~S~~emi-supervised learning**

I. Introduction

DEEPFAKE technology has has witnessed a notable surge in recent years. This particular technology possesses the capability to fabricate convincing images and videos that are virtually indistinguishable to the human eyes. In 2017,an individual known as Deepfakes posted a video on social media wherein the face of a person was seamlessly replaced with that of a movie star, thus attracting significant attention. Concurrently, the advancement of deepfake technologyhas accentuated the risks associated with network information security. The generation of falsified facial videos and images generated by deepfake technology can inflict harm upon an individual's reputation, property, and privacy. Moreover, the employment of deepfake technology in manipulating public opinion, compromising public safety, and influencing international relations poses a substantial threat to national security. Consequently, in recent years researchers have increasingly directed their focus towards deepface detection, resulting in the proposal of numerous deepface detection methods.

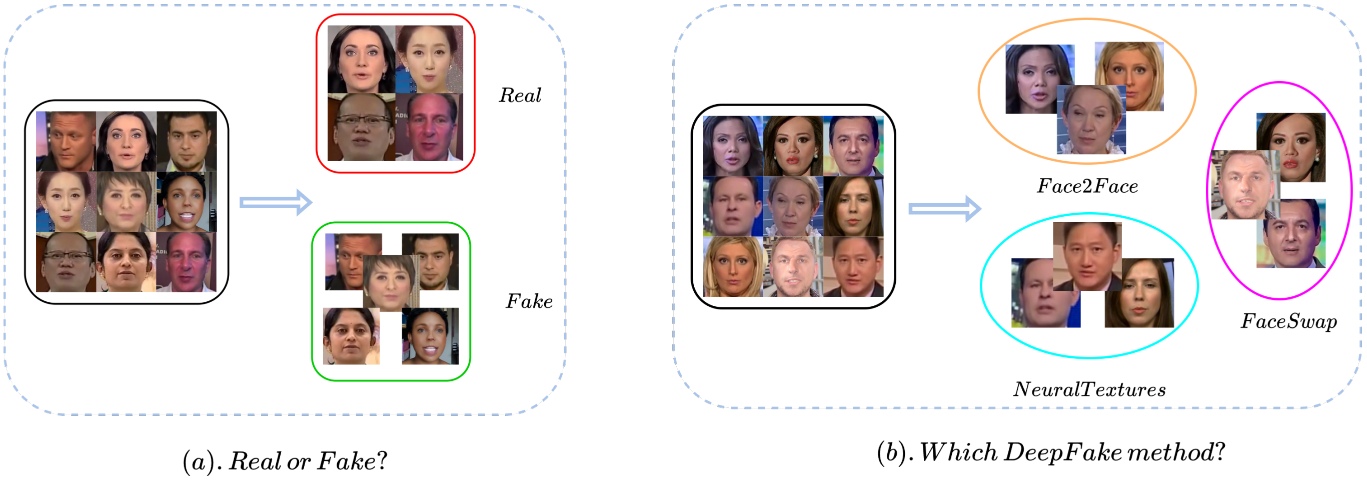
Based on the level of clues employed for detection, the existing deepface detection methods can be primarily classified into two categories: detection methods based on indicators of manipulated image details and detection methods based on inconsistencies arising from image tampering. The first category relies predominantly on abnormal key points, facial texture information and frequency domain information as primary indicators for discerning the authenticity of an image [4]–[9]. Conversely, the second category of methods centers on inconsistencies that arise during image tampering.Manipulating and editing solely the local area of the original image often lead to the inconsistency between the artificially generated part and the remaining part, and thus the second category of methods has a broader applicability in deepface detection [10]–[13]

As shown in Figure 1(a), existing methods for deepfake detection mainly focus on binary classification of real or fake images. However, in real-world scenarios, unknown deepfake types are frequently encountered, necessitating the need for fine-grained deepfake detection, as shown in Figure 1(b), to achieve robust cross-domain deepfake detection.Unfortunately, there is very limited research on cross-domain deepfake detection. ForgeryNet [1], which constructed a dataset containing 15 types of deepfake methods and suggested that it can be used for fine-grained deepfake detection. However, fine-grained deepfake detection in open scenarios faces several challenges: (1) the differences between false samples generated by different deepfake methods are small, making it difficult for the human eyes to distinguish. (2) The existing deepfake detection datasets are relatively small and imbalanced, making it difficult to perform fine-trained deepface detection in a supervised learning manner. (3) in open scenarios, models often face unknown deepfakes leading to a significant decrease in performance.

This paper proposes an unsupervised domain adaptive method for fine-grained deepfake detection in open scenarios. We first learn a multi-class deepface classification model using the labeled data in source domain, establishing initial ability to recognize different deepfake methods in the source domain. Then, the method uses a proposed Network Memory Effect-based Adaptive Clustering (NMEAC) to cluster unlabeled images in the target domain, and designs a Pseudo Label Generator (PLG) to recognize known deepfake types in the source

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received April 19, 2021; revised August 16, 2021.



(a). *Real or Fake*?

(b). *Which DeepFake method*?

Fig. 1. The difference between binary classification of real and fake images and fine-grained deepfake detection task. Figure (a) describes the task of using a model to identify the authenticity of given unknown images and classify them as real or fake. The task in Figure (b) is to identify the deepfake generation method of given fake images and classify them accordingly.

domain. Finally, the method uses a Model Re-Training (MR) step, to retrain the initial multi-class deepface classification network using both labeled source data and pseudo-labeled target domain data.

This main contributions of this work are as follows.

(1) To the best of our knowledge, this is the first study that investigates the fine-grained recognition of different deepfake methods, enabling the identification of unknown deepfake methods in open scenarios. (2) We propose an novel adaptive clustering method based on network memory effect, enabling the network to obtain reliable pseudo labels for the unlabeled target domain data. The pseudo-labels enable the source-domain multi-class deepfake detection network to classify unseen deepfake attacks correctly. (3) We construct multiple cross-domain fine-grained deepfake detection tasks based on several public datasets, enabling comprehensive evaluations of our method and the state-of-the-art (SOTA) methods.

II. Related Work

A. Deepfake Detection

~~With the emergence of GAN, the research for deepfake continues to deepen. The images generated by existing forgery methods are difficult to identify by human eyes, which has brought a certain impact on people's lives. Therefore, the task of deepfake detection has attracted many researchers to explore, and them have continued to conduct in depth research on the topic of continuously optimizing the deepfake detection network. Although there are many SOTA methods for deepfake detection, they can be roughly divided into the following two categories from the clue level of deepfake detection: The first type of work believes that deepfake detection should authenticate images from image detail clues. The second type of work argues that deepfake detection should focus on the inconsistencies generated from image tampering.~~

The first type of ~~work~~ is based on ~~indicators of manipulated image details, which mainly focus on detail clues such as facial landmark, texture features and frequency domain information~~. ~~[4] The authors argue that there is a huge difference between the deepfake generation process and the real-world imaging process~~. ~~The last step in the deepfake generation process is usually the decoding of the facial feature map into a three-channel RGB image. In this process, some internal correlation is introduced between the various channels of the generated image.~~ ~~However, the various channels of images collected in the real world are completely decoupled, and~~ ~~the author uses the correlation of the three channels of the generated image as a clue for image authentication.~~ Yang et al. [5] observed that each part of ~~the~~ face ~~in the fake video image generated by deepfake~~ can be generated ~~effectively and~~ with high quality, but there is no explicit constraint between ~~each part~~ during the ~~model training~~ process, ~~so that the facial layout of different parts is inconsistent with the real image~~. For ~~example,~~ the ~~positional relationship between~~ eyes and eyelashes ~~and the positional relationship between~~ nose and mouth. This ~~method~~ detects ~~whether~~ ~~the image is forged by detecting the key points of the face of the image and detecting the relative positional relationship~~ of each part. ~~[8]~~ Li et al. ~~used~~ local discrete cosine transform (DCT) to extract frequency domain features~~; integrated~~ metric learning and ~~used~~ single-center loss to ensure the similarity ~~within the~~ real face class. ~~[9]~~ Zhao et al. used a texture enhancer to extract shallow features, ~~used deeper features to generate attention maps~~, ~~and multiplied the attention maps point-by-point with shallow texture features after bilinear interpolation, and then used the obtained feature matrix For counterfeiting.~~

~~The second type of method is mainly based on the inconsistency caused in the process of image tampering, and this type of deepfake detection method has stronger robustness. Li et al. [10] observed that the forgery of editing and tampering with the original will include the step of fusing the forged region with the background region in the source image, so there will be an image stitching boundary. At this boundary there will be artifacts of the underlying features of the image. This method first finds the position of the stitching boundary~~

~~in the image, and judges whether it is forged according to the result of image segmentation. [11] defined a new face-changing method I2G (inconsistency image generator), using I2G for data enhancement, according to the pixel level labels of I2G, by predicting the consistency between patches in the feature space to learn a good feature representation. [12] Hope to be able to judge the authenticity of the picture by detecting whether the inside and outside of the face belong to the same ID, and then use the Transformer to output an inner identity and outer identity, and use the classification loss to learn. SBI [13] obtains two different forged images by performing different augmentations on different training samples during the model training process. The two newly generated forged images are obtained by fusing the facial mask formed by the key points of the face of the initial image. New forged images to improve the generalization ability of the model for deepfake detection.~~

~~The existing deep forgery detection meathods are mainly aimed at the two classifications of true and false.~~ They h~~ave not considered the problem of~~ fine-grained ~~recognition of forgery methods~~. ~~Although~~ Forgerynet [1] considers three ~~categories of classification, these three categories are respectively image authenticity, identity replacement, and do not explore the model's ability to recognize forgery methods.~~ ~~The main focus of this paper is to~~ study ~~the~~ fine-grained ~~classification of different~~ deepfake ~~methods~~. ~~Given~~ a set of ~~forged~~ images, ~~We want~~ ~~the~~ ~~model to be able to classify as accurately as possible which forgery method each fake image was generated by.~~

B. Domain Adaptation

~~With the development of deep learning and the advent of the era of big data~~, ~~many fields and tasks have been rapidly advanced by using a large amount of labeled data to assist model training. However, obtaining data annotations is cumbersome and difficult~~. ~~Therefore, researchers consider whether the information of one or more source domains can be transferred to the target domain, so as to improve the effectiveness of the model in the target domain.~~ ~~But due to many factors such as illumination, pose and image quality, there is always a distribution difference or domain shift between the two domains, which may lead to lower model performance.~~ Domain adaptation ~~is~~ to solve ~~the above problems, so that~~ the model trained in the source domain ~~can~~ be ~~directly~~ transferred to the target domain without ~~causing~~ a significant decline ~~in model performance~~. Existing domain adaptation methods can be ~~divided~~ into supervised domain adaptation and unsupervised domain adaptation ~~from the perspective of~~ whether the target domain data has labels or not. Supervised domain adaptation assumes that ~~the data in the~~ source domain and ~~the~~ target domain have ~~labels~~ [14]–[17],while unsupervised domain adaptation assumes that the source domain ~~data~~ has ~~labels~~ and the target domain data has no labels~~.~~ [18]–[22].

The supervised domain adaptation target domain data has clean labels to participate in training, and the knowledge of the source domain in domain adaptation can be transferred to the target domain more easily. [14] Constructed semantic alignment loss and separation loss based on the idea of embedding metric learning in deep learning, so that samples with the same label in the target domain and source domain are closer to each other, and samples with different labels are farther apart. [15] proposed a weight regularizer r_w to compensate the difference between the source domain and the target domain by minimizing the parameters of the jth layer of the source domain model and the target domain model. Chopra et al. [16] proposed a model for domain adaptation via inter-domain interpolation (DLID). DLID starts with all source data samples and gradually replaces source data with target data to generate intermediate datasets. After the intermediate dataset is generated, it is trained in an unsupervised manner using a deep nonlinear feature extractor that predicts a sparse decomposition.

Unsupervised domain adaptation uses labeled source domain data and unlabeled target domain data in the training process, hoping that the model can still be effective in unlabeled target domain samples during testing. CoGAN [18] proposes a dual-GAN structure to generate synthetic target data paired with synthetic source data. GAN1 is used to generate source data, and GAN2 is used to generate target data. The parameters of GAN1 and GAN2 in the first few layers in the generative model and the last few layers in the discriminative model are shared. The trained CoGAN can generate target domain data and assign corresponding labels to it based on source domain data. Therefore, the generated labeled target domain data can be leveraged to train the target model. Domain Adversarial Neural Network (DANN) [19] accidentally introduces an additional domain discriminator between the feature extractor and category classifier of the classification network to distinguish whether the input is from the source domain or the target domain. And between the domain discriminator and the feature extraction network, a gradient inversion layer GRL is set. DANN minimizes domain confusion loss (for all samples) and label prediction loss (for source samples), while maximizing domain confusion loss by using GRL. Shen et al. [20] propose to minimize the distance between the source and target domains by replacing the domain classifier with a network that learns to approximate the Wasserstein distance. Xavier and Bengio [21] proposed a stacked denoising autoencoder (SDA) to extract high-level representations of source and target domain data, and then jointly reconstruct the source and target domain data by using the same network.

C. ~~Open World Semi Supervised Learning~~

Traditionally speaking, the existing supervised learning and semi-supervised learning are designed for the closed world,and the unlabeled data types are consistent with the labeled data types. In the closed world [23], semi-supervised learning is more competitive than supervised learning. However,there are categories that do not exist in labeled data during testing in the real open scene, and the performance of the model will decrease. has declined. Therefore, some researchers began to study the task of semi-supervised learning in the open world. DS3L [24] does not fully exploit all unlabeled data like traditional semi-supervised learning work, but selectively

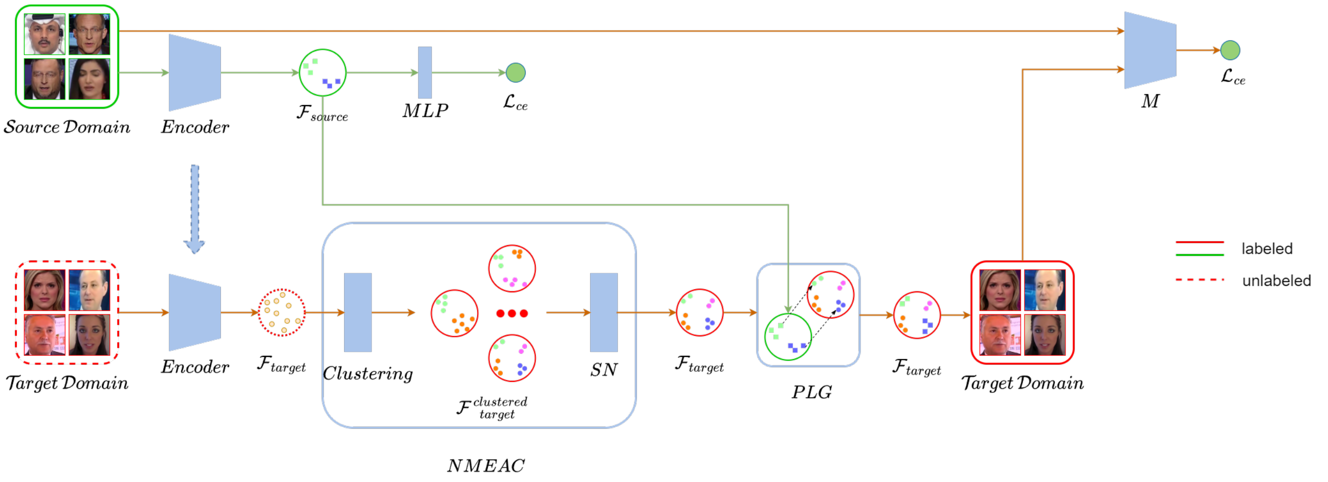


Fig. 2. First, use labeled data from the source domain to train a feature extraction network Encoder. Then, use Encoder to extract features F_T from unlabeled data X_T in the target domain. The extracted features F_T of the target domain are then used for clustering the unlabeled target domain images X_T using Network Memory Effect-based Adaptive Clustering (NMEAC) to convert the previously unclassified source domain features into labeled source domain features. Next, through the Target Domain Pseudo Label Generator (PLG), the adaptive clustering results of the target domain are associated with the known depth-forging methods of the source domain to obtain the optimal pseudo label $Y_T^{'}$ for the target domain data, resulting in Y_T^{*}. Finally, the pre-trained M in the first step is retrained using labeled data from the source domain and target domain data optimized by PLG to obtain the desired forging method through Model Retraining (MR).

chooses to use unlabeled data for model training. When unseen classes appear in the unlabeled dataset, samples from new classes are adaptively assigned low weights during training. While DS3L weakens unlabeled data with unknown categories to improve distribution matching, it also enhances labeled data to prevent performance degradation. ORCA [23] computes a supervised target loss for labeled data and a pairwise target loss for unlabeled data. Directly combining the labeled target loss with the paired unlabeled target loss, the generalization ability of the model to unknown categories has declined. The author proposes an adaptive margin based on uncertainty for this problem. During the training process Gradually reduces the plasticity of the model, increases the discriminability of the model, and reduces the gap between the known class and the new class

For the task of fine-grained classification of different deepfake methods, what we need to solve is the problem of performance degradation of classification models when faced with open-world unknown class forgery methods. These open-world semi-supervised learning works have given us some inspiration. As a baseline, the effect of the specific task of deep forgery fine-grained classification is not so obvious.

III. Proposed Approach

A. Method Overview

Our goal is to establish a fine-grained classification model for deepfake detection with better generalization performance, which can not only identify known deepfake methods in the source domain, but also better recognize unknown deepfake methods in the target domain. We propose an unsupervised domain adaptive deepfake detection model based on the model network memory effect. The main framework of our model is shown in Figure 2. Given labeled data X_S in the source domain and unlabeled data X_T in the target domain, there are N_S types of deepfake in X_S, $X_S = \{X_{s1}, X_{s2}, \dots, X_{sn}\}$, and each type of X_S has its corresponding label, $Y_S = \{Y_{s1}, Y_{s2}, \dots, Y_{sn}\}$. First, we pretrain a classification model M on X_S using the cross-entropy loss, which can effectively identify different deepfake methods. We freeze the Encoder in front of the pre-trained M classification head and use Encoder to extract features F_T from unlabeled data X_T in the target domain. Then, we perform adaptive clustering based on the network memory effect (NMEAC) on the extracted target domain features F_T to cluster the unlabeled images in the target domain, thereby assigning optimal pseudo labels $Y_T^{'}$ to the target domain data X_T. Next, we associate the adaptive clustering results in the target domain with the known deepfake methods in the source domain using a pseudo label generator (PLG) for the target domain, and optimize $Y_T^{'}$ to obtain the pseudo labels Y_T^{*}, $Y_T^{*} = \{Y_{t1}^{*}, Y_{t2}^{*}, \dots, Y_{tm}^{*}\}$. Finally, we retrain our pre-trained M using the labeled data from the source domain and the target domain data with optimized pseudo labels generated by PLG through model retraining (MR) to obtain our final fine-grained classification model for different deepfake methods. We will describe these steps in detail in the following chapters.

B. Source-Domain ~~Based Multi-Classification Model Learning~~

Similar to many works on adaptive learning [18]–[20] and open-world semi-supervised learning [23], [24], we need to maximize the use of labeled data to learn useful knowledge and transfer it to data in the target domain. We train a source domain classifier M using the pre-training loss L_{mp} on data with various known forgery methods in the source domain, and expect the model to effectively classify the known forgery methods. The pre-trained M has two main functions: 1) extracting features F_T from unlabeled data in the target domain. We cannot directly train the unlabeled target domain data X_T using the supervised objective loss, so we need to assign the most effective pseudo labels. To do this, we need to classify the target domain data, and before doing so, we need an effective encoder Encoder to extract features from unlabeled data. We hope that the Encoder only focuses on features that can distinguish different forgery methods. Therefore, we freeze the pre-trained model M before the classification head and use it as an encoder to focus on fine-grained features that distinguish different forgery methods. 2) The core of our method is to assign optimal pseudo labels to the unlabeled data in the target domain, and then use the labeled data in the source domain and the pseudo-labeled data in the target domain to train a deepfake method recognition model. Therefore, at the end of our method, we need to retrain the trained M model for this part.

$$L_{mp} = -\sum_{i=1}^{N_S} Y_i \ln \frac{e^{Y_i}}{\sum_{i=1}^{N_S} e^{Y_i}}. \tag{1}$$

C. Adaptive Clustering Based on Network Memory Effect

For the unlabeled data in the target domain, use the pre-trained Encode to obtain the features that are inclined to the fine-grained classification of the deepfake method. We hope to use these target domain data features to assign the optimal pseudo-label to these unlabeled data. For these target domains The features are clustered to place the most effective pseudo-labels for the target domain data. In [25], the network memory effect is proposed, that is, model training in supervised learning tends to be more biased towards clean data annotation. If incorrect data annotations (i.e. noise labels) are used to assist model training, the performance of the model will degrade significantly. Based on this principle, we propose an adaptive clustering NMEAC based on network memory effects.

In machine learning, K-Means [26] is the most commonly used clustering technique. For unlabeled data, K-Means can divide the data according to its data characteristics. K-means first sets the unlabeled data to k center points $a = \{a_1, a_2, ...a_k\}$, calculates the distance to the k cluster centers for each sample in the data set and divides it into the cluster center with the smallest distance In the corresponding class, recalculate its clustering center a_j according to Formula 2 for each class, and repeat the above work until the optimal clustering is obtained. However, in the process of clustering, there are two problems that need to be solved. One is how to divide the different types so that the different types can be separated according to the division criteria we want. The other is to determine the final number of categories for clustering. The problem. In our adaptive clustering NMEAC process, for the first question, we first pre-trained a classifier model that can identify different deepfake methods, and used the feature encoder Encoder of this model to extract other Features, and use K-Means clustering to divide them, so that we can divide the target domain data according to different levels of fake fine-grained classification in NMEAC. As for the problem we need to solve in unlabeled data clustering, we should use K-Means to divide the target domain unlabeled data into several categories. For this problem, in traditional machine learning, researchers [26] have used the following method to obtain the optimal K. Manually set a maximum category number N of K-Means, set an objective function L, and use K-Means to cluster the data from 2 to N. Find the optimal solution of the objective function L. The effect of this clustering method is acceptable, but there are obvious defects: how to set the maximum number of categories, and whether the maximum threshold set is less than the actual number of categories of unlabeled data. In addition, when the number of sample data is small, this is achievable. However, when the data size increases to a large enough size, the workload of clustering the data according to the number of all cluster types from beginning to end is huge.

$$a_j = \frac{1}{c_j} \sum_{x \in c_j} x. \tag{2}$$

In NMEAC, we first artificially set a maximum number of species N according to the previous work. Use K-Means to cluster the target domain features extracted by the Encoder from 2 to N in sequence. Afterwards, we sequentially input the target domain data of different clustering methods into the selection network SN, and obtain the final output of NMEAC by observing the training situation of the SN network. Here, the first improvement we make is no longer simply to choose an objective function to find the optimal number of clusters. We use the principle of the model to focus on clean labels (network memory effect), design a selection network SN, and use different Different types of target domain data clustered by K value train SN through a small number of iterations. We observe the change of SN loss and the change of classification accuracy during the training process. The loss during SN training is cross-entropy loss. We conducted the NMEAC test according to the above method. After a large number of experimental results, it was proved that, as shown in Figure 3, when the number of categories from 2 clustering to the actual number of unlabeled data, the loss of SN training has been showing a downward trend, and the accuracy of SN It has been an upward trend. However, when the number of clusters exceeds the ideal optimal number of clusters, the loss of SN starts to increase and the accuracy rate begins to decline. In addition, according to the data comparison, the adaptive clustering under the ideal optimal clustering number K value obtained the target domain data with pseudo-labels, and the convergence trend of the model in the SN training process is much faster than other clustering methods. The second important improvement is to start clustering when the number of clusters is 2. When the acc of SN is observed for the Nth time, the loss starts to rise, and N-1 is the optimal number of clusters for our NMEAC.

Therefore, our final design of NMEAC clusters the target domain feature F_T input to EMEAC from the initial two categories, and the clustering result assigns a pseudo-label $Y_n^{'}$ to the target domain data X_T, and then these targets with pseudo-labels The domain data is sent to the SN network to train a small number of iterations emax, and the final loss (cross entry loss) and accuracy acc are recorded. Repeat the above steps until the loss of SN starts to rise and the accuracy

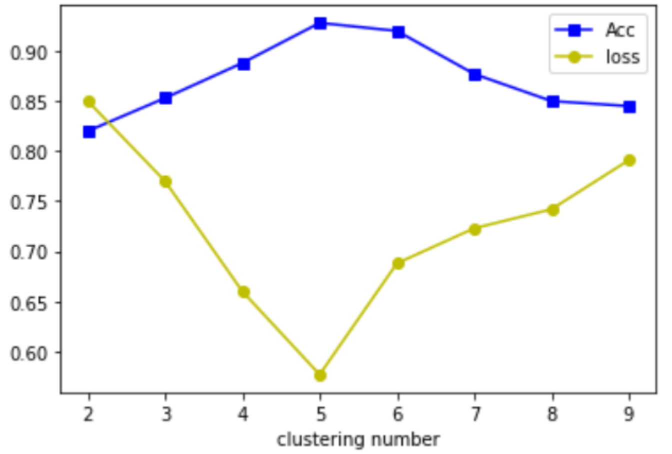


Fig. 3. the impact of different cluster numbers on the loss and accuracy of SN network training in EMNAC

rate starts to drop, then this time the value of K before this time is the category number N_T of our optimal final target domain deepfake method. Based on this, the optimal pseudo-label $Y_T^{'}$ is assigned to the target domain data, and the output (X_T, $Y_T^{'}$) of EMEAC is obtained. The adaptive clustering algorithm based on network memory effect is as follows.

Algorithm 1 Algorithm for Adaptive Clustering based on Network Memory EFect

```
Input: $X_T$: Target domain images; $\mathcal{E}$: Feature Extractor; $\mathcal{C}$: K-Means clustering; $\mathcal{S}$: Selecting Net; $e_{max}$: the maximum epochs for training selecting net; $N$: K-Means clustering theoretical maximum,adaptive clustering stops before clustering $X_T$ into N classes;
Output: $(X_T, Y_T^{'})$: Target domain images with the clustered pseudo-labels,$X_T = \{X_{t1}, X_{t2}, ..., X_{tm}\}$,$Y_T^{'} = \{Y_{t1}^{'}, Y_{t2}^{'}, ..., Y_{tm}^{'}\}$; $N_T$: Final K value for NMEAC
set $Acc^* \leftarrow 0, Loss^* \leftarrow 1e3, N \leftarrow 1e5$;
$F_T \leftarrow \mathcal{E}(X_T)$;
for $n = 2$; $n < N$; $n++$ do
    $(X_T, Y_n^{'}) \leftarrow \mathcal{C}(F_T, n)$;
    for $i = 0; i < e_{max}; i++$ do
        Training $S(X_T, Y_n^{'}, i)$:
    end for
    $(Acc_n, Loss_n) \leftarrow S(X_T, Y_n^{'}, e_{max})$;
    if $Acc_n > Acc^*$ and $Loss_n < Loss^*$ then
        $Acc^* \leftarrow Acc_n, Loss^* \leftarrow Loss_n$;
    else if $Acc_n < Acc^*$ and $Loss_n > Loss^*$ then
        $m \leftarrow n - 1$;
        $(X_T, Y_T^{'}) \leftarrow \mathcal{C}(F_T, N_T)$;
        Return $(X_T, Y_T^{'})$;
    end if
end for
```

D. Target Domain Pseudo-label Generation

Through NMEAC, we have assigned the most effective pseudo-label $Y_T^{'}$ to the target domain unlabeled data X_T. Our ultimate goal is to combine the target domain unlabeled data X_T and the source domain labeled data X_S for model retraining, so that the model can Identify unknown deepfake methods in the target domain, and better improve the fine-grained classification ability of model deepfake methods. However, there will be a problem to be solved urgently: when the source domain data and the target domain data have the same forgery type, the source domain data will have its own accurate label. For the forgery type data in the target domain, we have assigned them in NMEAC new pseudo-label. If the source domain data and the target domain data are directly trained jointly, it is easy to make the model ambiguous, resulting in a decline in the ability to identify forgery methods. Therefore, it is very important to find the type of forged method already in the source domain in the target domain data marked with pseudo-labels, and to give it the same category label as the source domain data. As for other methods not found in the source domain in the target domain The matching data is an unknown type of deepfake in the open world.

Therefore, we designed a target domain data pseudo-label generator PLG after NMEAC to optimize the optimal pseudo-label $Y_T^{'}$ assigned by NMEAC for the target domain data X_T. We sample the same number of samples for each category of n-type source domain data and m-type target domain data to obtain X_{si},X_{tj} , (i=1,2,n, j=1,2,. . . ,m), using The deepfake method fine-grained classifier Encoder extracts the source domain data X_{si}, target domain data X_{tj} features to obtain source domain data features F_{si}, target domain data features F_{tj}. We set ϕ to be used to calculate the mean mu and covariance matrix Σ of each category feature matrix in the source domain and the target domain, and we calculate μ_{si}, Σ_{si} and μ_{tj}, Σ_{tj} of the respective mean and covariance matrix for F_{si}, F_{tj}. In this way, we have separate statistics mu, Σ for each type of forged data distribution in the source domain and target domain. In order to find out the part of data in the target domain that has the same type of forgery as the source domain, we propose a forgery method category feature distance FID [27]. For each forgery type in the source domain data, we calculate the forgery method category feature distance FID to each forgery type data in the target domain data. We determine the two sets of deepfake data with the closest statistical distribution based on the minimum value of the FID. In this way, for each known forgery type in the source domain data, the data with the closest deepfake type is found in the target domain data. We optimize the existing pseudo-labels of the target domain data we find, and the remaining target domain data still use the pseudo-labels assigned by NMEAC as unknown forgery types. Finally, after PLG optimization, the target domain data X_T is given a new pseudo-label Y_T^*

$$\begin{aligned} FID = argmin(|\mu_{si} - \mu_{tj}|^2 \\ + Tr(\Sigma_{si} + \Sigma_{tj} - 2(\Sigma_{si}\Sigma_{tj})^{\frac{1}{2}})). \end{aligned} \tag{3}$$

E. Model Retraining

After NMEAC divides the target domain data into different categories of deepfake methods, and then uses PLG to optimize the pseudo-labels for these divided target domain data,

TABLE I
THE COMMONLY USED DATA SETS FOR DEEPFAKE DETECTION INTRODUCE THE NUMBER OF REAL/FALSE/TOTAL VIDEOS CONTAINED IN THE EXISTING DEEPFAKE DATA SETS, THE NUMBER OF VIDEO INTERVIEWERS IN THE DATA SET, THE TYPES OF DEEPFAKE METHODS, AND THE NUMBER OF AVAILABLE DEEPFAKE METHODS.

Dataset	**Vidoes(R/F/T)**	**Subjects**	**Methods(Numbers)**
UADFV	49/49/98	49	FakeApp(1)
DeepfakeTIMIT	640/320/960	32	Faceswap-GAN(1)
FaceForensics++	1000/4000/5000	N/A	Deepfakes,Face2Face,FaceSwap, NeuralTextures,FaceShifter(5)
Celeb-DF	590/5639/6229	59	DeepFake(1)
DeepFakeDetection	363/3000/3363	28	Deepfakes,Face2Face,FaceSwap, NeuralTextures,FaceShifter(5)
DFDC	23654/104500/128154	960	8 unknown methods
DFDC(preview)	1131/4119/5250	N/A	2 unknown methods
DeepForensics-1.0	50000/10000/60000	100	DF-VAE(1)
ForgeryNet	99630/121617/221247	5400	FaceShifter,Deepfakes,FSGAN,Blendface, MMReplace,DSS,SBS,Talking-headVideo, ATVGNet,FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG(15)
FakeAVCeleb	500/19500/20000	500	FaceSwap,FSGAN,Wav2Lip(3)

now our source domain data X_S has a clean label Y_S, and the target domain data X_T has the PLG Optimize the pseudo-label Y_T^*. We hope to use source domain data (X_S, Y_S) and target domain data (X_T, Y_T^*) to retrain our previously pre-trained M , so that our deep fake fine-grained classification network is not only in the source domain data It can effectively identify different deepfake methods, and the performance of the test on the target domain data will not decrease significantly. In the part of model retraining, we treat the source domain data and the target domain data equally, and take the same number of samples for each category of data in the two domains as the training set $X_S \cup X_T$ of MR. There are a total of $N_{S\cup T}$ types in the training set For the deepfake type, we use the retraining loss L_{mr} to retrain the model on the training set for the previously pretrained M. In the next experimental chapter, we will introduce the effectiveness of our method in detail from a large number of experimental results, and it will also be verified from the ablation experiment that the various method steps we introduce in this chapter are essential.

$$L_{mr} = -\sum_{i=1}^{N_{S\cup T}} Y_i \ln \frac{e^{Y_i}}{\sum_{i=1}^{N_{S\cup T}} e^{Y_i}}. \tag{4}$$

IV. EXPERIMENTS

A. Datasets

Table 1 is the commonly used data sets for deepfake detection. We selected three data sets with more forgery methods for our experimental verification, including Forgerynet [1], FaceForensics++ [2],FakeAVceleb [3]

Forgerynet: the largest publicly available face forgery dataset with unified annotations in image and video-level data, covering four tasks: 1) Image forgery classification. 2) Spatial forgery positioning. 3) Video forgery classification 4) Time forgery location. ForgeryNet has a total of 2.9 million images, 221,247 videos, 15 deepfake operations, and 36 independent and more mixed perturbations. Deepfake methods include FaceShifter [28], Deepfakes [29], FSGAN [30], Blendface, MMReplace, Deepfakes-StarGAN2-Stack, StarGAN2-Blendface-Stack, Talking-headVideo [31], ATVG-Net [32], FOMM [33], StyleGAN [34], MaskGAN [35], StarGAN [36], SC-FEG [37], DFG [38].

FaceForensics++: the commonly used deepfake detection dataset. FF++ has a total of 1000 original video sequences, and these original videos come from 977 youtube videos. In addition, there are a total of 5000 fake videos in the data set, all of which are generated from 1000 original videos through the following five deepfake methods: Deepfakes [29], Face2Face [39], FaceSwap [40] and NeuralTextures [41], FaceShifter [28]]. In FF++, each video contains a trackable unoccluded frontal face. In addition, the dataset provides face masks for image and video classification and segmentation.

FakeAVCeleb : the deepfake dataset generated by selecting 490 videos from VoxCeleb2 [42] as original videos. Unlike the gender-biased VoxCeleb2 dataset, FakeAVCeleb selects 400 raw videos from four different races, Caucasian, Black, Asian (Southern) and Asian (Eastern), each containing 100 celebrities Real videos, 50 of these 100 videos are male and 50 are female, in addition, 40 are male and 50 are female in East Asia, a total of 90 original videos. FakeAVCeleb uses state-of-the-art deepfake and synthetic speech generation methods to generate our FakeAVCeleb dataset. We use face swapping methods FaceSwap [40], FSGAN [30] and Wav2Lip [43] to generate deepfake videos.

B. Experimental Protocols

All our experiments are divided into cross-dataset experiments and within-dataset experiments. According to the research task, whether the deepfake method of the target domain data is consistent with the source domain data and whether the source domain target domain data distribution is consistent, we divide the cross-dataset experiment into three experimental protocols: 1) The deepfake method is the same, but the data distribution is different , a cross-dataset experiment where the source domain is labeled and the target domain is unlabeled. In this experimental protocol, we mainly investigate whether our method is better than the existing state-of-the-art methods for isomorphic domain-adaptive domain shifting. 2) The deepfake methods are not completely consistent, the data distribution is inconsistent, and the source domain has labels and the target domain has no labels for cross-dataset experiments. In this experimental protocol, the source domain and the target domain have no intersection, and the data distribution is inconsistent. 3) The deepfake method is not completely consistent, the data distribution is partially consistent, and the source domain has labels and the target domain has no labels for cross-dataset experiments. In this experimental protocol, the target domain data will include source domain data and other data, and the distribution of source domain data and target domain data is not completely consistent. In the two experimental protocols of 2 and 3, the target domain data contains forgery methods that have not appeared in the active domain data. We use these two sets of experimental protocols to simulate the scene of unknown deepfake methods that appear in reality, and hope to use this To prove that our method can effectively identify unknown deepfake methods in addition to identifying known deepfake methods. The details of the three sets of cross-dataset experimental protocols are shown in Table 2-5. Each table introduces the datasets used in the source and target domains and the deepfake methods contained in the data in the source and target domains.

All of our cross-domain experiments are carried out on the three deepfake detection data sets of ForgeryNet, FaceForensics++, and FakeAVceleb. In order to explain the two cross-domain experimental protocols in detail, we will record ForgeryNet as N, FaceForensics++ as F, and FakeAVceleb as C. In the first experimental protocol, we need to do cross-domain experiments on the same deepfake method category in different data sets. By observing that F and N have two identical methods FaceShifter and Deepfakes, C and F have an identical fake method FaceSwap, C and N have the same forgery method FSGAN. Therefore, according to the method of (source domain → target domain), the first experimental protocol is divided into 6 groups of experiments, and the details of the specific experimental protocol are shown in Table 3. In addition, we hope that the target domain contains deepfake methods that have not been seen in the source domain data to prove the effectiveness of our method for identifying unknown deepfake methods, and conduct cross-domain experiments on different data sets and their combinations. . The second and third experimental protocols are formulated according to whether the source domain data and the target domain data are completely inconsistent. In the second experimental protocol, the data in the source domain and the target domain have no intersection, the data distribution is inconsistent, and the target domain contains unknown forgery methods in the source domain. The specific protocol details are shown in Table 4. In the third experimental protocol, the target domain data contains source domain data, and the data distribution is not completely consistent. The target domain contains unknown forgery methods in the source domain. The specific protocol details are shown in Table 5.

For the first experimental protocol, because the deepfake types of the source domain and the target domain data are exactly the same, the model in our method can always use the classification head of N in the model pre-training during training and testing. In order to show the performance difference between our method and the SOTA method in cross-domain experiments, we calculate the classification accuracy Acc of the model of each sub-protocol on the target domain test set in the first experimental protocol. For the second and third experimental protocols, because the target domain contains deepfake methods whose source domain data is unknown, the classification header of N in the model pre-training cannot be used all the time, and the accuracy of the model in the test set cannot be directly tested. Therefore, we use the division method of query set and reference set in DA to divide the target domain test set according to probe and gallery. Use the feature extractor retrained by our model to extract its features from the data in the probe and gallery, and find the reference sample that finds the top5 similarity in the reference set for each test query sample. And calculate the top5 classification accuracy ACC of the model based on the real labels of the query and reference samples.

In the cross-dataset experiment, we selected SBI [13], DS3L [24], ORCA [23], SDAT [44] as the baseline of the experiment, and compared our method with these baselines in the various experimental protocols that have been set. The test results in to verify the effectiveness of our proposed method. (SBI is a recent representative SOTA method for deepfake detection. This method obtains two different forged images by performing different augmentations on different training samples during the model training process. The two newly generated forged images are obtained from the initial image. The facial mask formed by the key points of the face is fused to obtain a new forged image, so as to improve the generalization ability of the deepfake detection of the model. In our cross-dataset experiment, we added SBI when training the forgery method classifier in the source domain into the data augmentation approach as an experimental baseline) DS3L is one of the existing SOTA methods in open-world semi-supervised learning, and DS3L [24] adaptively Samples of the new class are assigned low weights. ORCA [23] proposed an uncertainty-based adaptive margin to combine the supervised target loss of labeled data and the paired target loss of unlabeled data, thereby improving the generalization ability of the model to unknown categories. SDAT is one of the domain confrontation training (DAT) SOTA methods in the existing DA. SDAT proposes a simple and novel DAT formula, which improves the smoothness of the near-optimal task loss in the DAT algorithm and thus improves the model on the target domain. Generalization. In addition, there is a baseline where the model is trained on source domain data and tested directly on target domain data.

As for the experiments in the dataset, we divided the experimental protocols into (F→F), (N→N), and (C→C) on the three deepfake datasets F, N, and C. The source and target domain data in each sub-protocol are equally sampled for each forgery method in the modified experimental protocol dataset. The experimental baseline in the dataset is consistent with the baseline of the cross-dataset experiment, and the evaluation index of the experimental results uses the classification accuracy Acc of the model on the target domain data.

TABLE II
Experimental Protocol 1:Intra-dataset Experiment with the same deepfake method, the same data distribution, and the source domain is labeled and the target domain is unlabeled

S→T	Deepfakes in Source Domain	Deepfakes in Target Domain
F→F	Deepfakes,Face2Face,FaceSwap, NeuralTextures,FaceShifter	Deepfakes,Face2Face,FaceSwap, NeuralTextures,FaceShifter
N→N	FaceShifter,Deepfakes,FSGAN, Blendface,MMReplace,DSS,SBS, Talking-headVideo,ATVGNet, FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG	FaceShifter,Deepfakes,FSGAN, Blendface,MMReplace,DSS,SBS, Talking-headVideo,ATVGNet, FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG
C→C	FaceSwap,FSGAN,Wav2Lip	FaceSwap,FSGAN,Wav2Lip

TABLE III
Experimental Protocol 2:Cross-dataset experiments with the same deepfake method, different data distributions, source domains with labels and target domains without labels

S→T	Deepfakes in Source Domain	Deepfakes in Target Domain
F→N	FaceShifter,Deepfakes	FaceShifter,Deepfakes
N→F	FaceShifter,Deepfakes	FaceShifter,Deepfakes
F→N&C	FaceShifter,Deepfakes,FaceSwap	FaceShifter,Deepfakes,FaceSwap
N&C→F	FaceShifter,Deepfakes,FaceSwap	FaceShifter,Deepfakes,FaceSwap
N→F&C	FaceShifter,Deepfakes,FSGAN	FaceShifter,Deepfakes,FSGAN
F&C→N	FaceShifter,Deepfakes,FSGAN	FaceShifter,Deepfakes,FSGAN

TABLE IV
Experimental Protocol 3: Cross-dataset experiment with incompletely consistent deepfake methods, inconsistent data distribution, source domains with labels and target domains without labels

S→T	Deepfakes in Source Domain	Deepfakes in Target Domain
F→N	Deepfakes,Face2Face,FaceSwap, NeuralTextures,FaceShifter	FaceShifter,Deepfakes,FSGAN, Blendface, MMReplace,DSS,SBS,Talking-headVideo, ATVGNet,FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG
N→F	FaceShifter,Deepfakes,FSGAN, Blendface, MMReplace,DSS,SBS,Talking-headVideo, ATVGNet,FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG	Deepfakes,Face2Face,FaceSwap, NeuralTextures,FaceShifter
F→C	Deepfakes,Face2Face,FaceSwap, NeuralTextures,FaceShifter	FaceSwap,FSGAN,Wav2Lip
C→F	FaceSwap,FSGAN,Wav2Lip	Deepfakes,Face2Face,FaceSwap, NeuralTextures,FaceShifter
N→C	FaceShifter,Deepfakes,FSGAN, Blendface, MMReplace,DSS,SBS,Talking-headVideo, ATVGNet,FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG	FaceSwap,FSGAN,Wav2Lip
C→N	FaceSwap,FSGAN,Wav2Lip	FaceShifter,Deepfakes,FSGAN, Blendface, MMReplace,DSS,SBS,Talking-headVideo, ATVGNet,FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG
F&C→N	Deepfakes,Face2Face,FaceShifter,FaceSwap, NeuralTextures,FSGAN,Wav2Lip	FaceShifter,Deepfakes,FSGAN, Blendface, MMReplace,DSS,SBS,Talking-headVideo, ATVGNet,FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG
N→F&C	FaceShifter,Deepfakes,FSGAN, Blendface, MMReplace,DSS,SBS,Talking-headVideo, ATVGNet,FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG	Deepfakes,Face2Face,FaceShifter,FaceSwap, NeuralTextures,FSGAN,Wav2Lip
C&N→F	FaceShifter,Deepfakes,FSGAN, Blendface, MMReplace,DSS, SBS,Talking-headVideo, ATVGNet,FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG,FaceSwap,Wav2Lip	Deepfakes,Face2Face,FaceSwap, NeuralTextures,FaceShifter
F→C&N	Deepfakes,Face2Face,FaceSwap, NeuralTextures,FaceShifter	FaceShifter,Deepfakes,FSGAN, Blendface, MMReplace,DSS, SBS,Talking-headVideo, ATVGNet,FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG,FaceSwap,Wav2Lip
F&N→C	FaceShifter,Deepfakes,FSGAN, Blendface, MMReplace,DSS,SBS,Talking-headVideo, ATVGNet,FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG,NeuralTextures, Face2Face,FaceSwap	FaceSwap,FSGAN,Wav2Lip
C→F&N	FaceSwap,FSGAN,Wav2Lip	FaceShifter,Deepfakes,FSGAN, Blendface, MMReplace,DSS,SBS,Talking-headVideo, ATVGNet,FOMM,StyleGAN,MaskGAN, StarGAN,SCFEG,DFG,NeuralTextures, Face2Face,FaceSwap

C. Implementation Details

1) Network Structure: The pre-trained deepfake fine-grained classifier N and the model-retrained classifier N share the same network structure, and N is composed of a feature

TABLE V
EXPERIMENTAL PROTOCOL 4: CROSS-DATASET EXPERIMENT WITH INCOMPLETELY CONSISTENT DEEPFAKE METHODS, CONSISTENT DATA DISTRIBUTION, SOURCE DOMAINS WITH LABELS AND TARGET DOMAINS WITHOUT LABELS

S→T	Deepfakes in Source Domain	Deepfakes in Target Domain
F→F&N	Deepfakes,Face2Face,FaceSwap,NeuralTextures,FaceShifter	Deepfakes,Face2Face,FaceSwap,NeuralTextures,FaceShifter,FSGAN,Blendface,MMReplace,DSS,SBS,Talking-headVideo,ATVGNet,FOMM,StyleGAN,MaskGAN,StarGAN,SCFEG,DFG
N→F&N	FaceShifter,Deepfakes,FSGAN,Blendface,MMReplace,DSS,SBS,Talking-headVideo,ATVGNet,FOMM,StyleGAN,MaskGAN,StarGAN,SCFEG,DFG	Deepfakes,Face2Face,FaceSwap,NeuralTextures,FaceShifter,FSGAN,Blendface,MMReplace,DSS,SBS,Talking-headVideo,ATVGNet,FOMM,StyleGAN,MaskGAN,StarGAN,SCFEG,DFG
F→C&F	Deepfakes,Face2Face,FaceSwap,NeuralTextures,FaceShifter	Deepfakes,Face2Face,FaceSwap,NeuralTextures,FaceShifter,FSGAN,Wav2Lip
C→F&C	FaceSwap,FSGAN,Wav2Lip	Deepfakes,Face2Face,FaceSwap,NeuralTextures,FaceShifter,FSGAN,Wav2Lip
N→N&C	FaceShifter,Deepfakes,FSGAN,Blendface,MMReplace,DSS,SBS,Talking-headVideo,ATVGNet,FOMM,StyleGAN,MaskGAN,StarGAN,SCFEG,DFG	FaceShifter,Deepfakes,FSGAN,Blendface,MMReplace,DSS,SBS,Talking-headVideo,ATVGNet,FOMM,StyleGAN,MaskGAN,StarGAN,SCFEG,DFG,FaceSwap,Wav2Lip
C→N&C	FaceSwap,FSGAN,Wav2Lip	FaceShifter,Deepfakes,FSGAN,Blendface,MMReplace,DSS,SBS,Talking-headVideo,ATVGNet,FOMM,StyleGAN,MaskGAN,StarGAN,SCFEG,DFG,FaceSwap,Wav2Lip

encoder Encoder and a classification head Mlp. There are four residual blocks in the Encoder, and each residual block has four convolutional layers, and the settings of the convolutional layers are the same as the convolutional part of Convnext [45]. The Encoder maps the (3*224*224) input image of the source domain or the target domain to a 1024-D feature vector in the deepfake fine-grained feature space. MLP is composed of two layers of fully connected network. In the model pre-training stage, the 1024-dimensional source domain feature vector is mapped to the scores of samples belonging to each of the n categories of the source domain. The selector network SN in NMEAC adopts the same structure as Resnet-18 [46], and selects the optimal clustering scheme by observing the convergence during SN training.

2) Training Details: We use the open source face-alignment for face detection and face keypoint location. All detected faces are cropped to 256 × 256 images based on 5 facial keypoints of eyes center, nose and mouth. We scale the cropped face area to a size of 224*224. In the model pre-training stage MP, we use random cropping, random horizontal flipping, color dithering and other methods for data augmentation. In this stage we train end-to-end by minimizing the loss in Equation 1. We do not use any data augmentation when training SN in NMEAC, and the training loss uses cross-entropy loss. In the process of model retraining MR, we only perform data enhancement of color dithering, and the loss adopts formula 4. Both MP and MR use the ADAM [47] optimization method to update the parameters of N. The initial learning rate of MP is set to 4e-3 to train 100 epochs, and the initial learning rate of MR is set to 4e-3 to train 50 epochs. The SN in NMEAC chooses SGD [48] to optimize network parameters. The initial learning rate is set to 0.01, and only 10 epochs are trained to observe the final acc and loss to determine the optimal result of NMEAC.

TABLE VI
RESULTS FOR INTRA-DATASET EXPERIMENT WITH THE SAME DEEPFAKE METHOD, THE SAME DATA DISTRIBUTION, AND THE SOURCE DOMAIN IS LABELED AND THE TARGET DOMAIN IS UNLABELED

Method	F→F	N→N	C→C
SBI	96.7	97.3	97.8
DS3L	97.3	99.1	98.4
SDAT	98.2	97.7	99.1
Ours	**98.3**	**99.3**	**99.5**

D. Intra-Dataset Experiment Results

In Experimental Protocol 1, we conducted intra-dataset experiments in each of the three data sets FaceForenics++, ForgeryNet, and FakeAVCeleb. In each data set, we selected 12,000 images from the data of each forgery method, according to 5/5/2 is divided into source domain training set, target domain training set, and test set. The selected baseline experiments are the deepfake detection SOTA method SBI, the semi-supervised learning SOTA method DS3L and the unsupervised domain adaptive method SDAT. The experimental results are shown in Table 6. The results of our method on the test set of the model under the three experimental protocols are all SOTA. As shown in the table, although our method has the highest model recognition accuracy compared with other methods, the gap with other methods is not obvious after comparison. The reason is mainly because the cross-domain experimental protocol in the data set is carried out in each deepfake data set, the source of the original data of the forged image is the same, the data of the source domain and the target domain are identically distributed, and the method of deepfake is consistent. The individual experiments in are straightforward. Experiments in the data set prove that our method can well identify known deepfake methods, and our method is SOTA in the closed world

TABLE VII
RESULTS FOR CROSS-DATASET EXPERIMENTS WITH THE SAME DEEPFAKE METHOD, DIFFERENT DATA DISTRIBUTIONS, SOURCE DOMAINS WITH LABELS AND TARGET DOMAINS WITHOUT LABELS

Method	**F→N**	**N→F**	**F→N&C**	**N&C→F**	**N→F&C**	**F&C→N**
SBI	92.3	91.6	90.1	90.8	93.5	92.3
DS3L	93.1	92.3	91.2	91.7	92.1	91.8
SDAT	91.5	90.4	89.5	90.1	91.6	92.5
Ours	**95.2**	**96.3**	**94.8**	**95.9**	**96.3**	**97.7**

E. Cross-dataset Experiment Results

1) Experimental Analysis for Experimental Protocol 2: Compared with the experiment in the dataset, the source domain and target domain deepfake methods of this experimental protocol are the same, but the source domain target domain data distribution is different, hoping to verify our ability to identify known deepfake methods for deepfakes in the open world. In this experimental protocol, according to the protocol details shown in Table 3 of the experimental protocol, we select 10,000 data for each forgery method in the source domain and target domain data sets of each experiment to form a training set, and then select 2,000 data as a test. set. The baseline model compared with the experimental protocol is the same as the experiment in the dataset.

Although SBI is currently the most effective deepfake recognition method, it essentially sets a data augmentation method to fit a new deepfake method, and it is effective in face-changing and other deepfake recognition methods. However, the recognition performance drops when the forgery method is not based on the facial contour change or even the forgery method is unknown (Table 7, row 2, column 3). DS3L weakens the loss of unlabeled data in the target domain so that the model can still achieve better recognition ability in the target domain. (Table 7, row 2) DS3L is sub-SOTA in single-dataset cross-domain testing. SDAT adds GRL to the domain confrontation training framework and proposes a new target domain loss. In the baseline experiment, we carried out comparative experiments in our experimental protocol according to the way the author mentioned SDAT+MCC++CDAN in the paper.

After the analysis of the results of this cross-domain experiment, as shown in Table 7, in each experiment under the experimental protocol, our method exceeds the accuracy rate of the existing SOTA method by 3%-5%. The horizontal comparison in row 5 of Table 7 can also well explain the fact that when faced with too many unknown forgery methods, the recognition ability of the model will decline. The experimental results can prove that when our method is faced with known deepfake methods in the open world, the model recognition ability is effective, and the performance of the model will not be significantly reduced due to the inconsistency of the source image of the forged image (different data distribution).

2) Experimental Analysis for Experimental Protocol 3: Compared with the previous experimental protocol, the distribution of data in the source domain and target domain of this experimental protocol is different, but the target domain data may contain existing forgery methods in the source domain as well as unknown forgery methods. We hope to validate our ability to identify known deepfake methods and unknown deepfake methods in the open world. In this experimental protocol, according to the protocol details shown in Table 3 of the experimental protocol, we select 10,000 data for each falsification method in the source domain and target domain data sets of each experiment to form a training set, and then select 2,000 data respectively according to 1 The ratio of :1 is used as the query set and reference set of the test set. The baseline models compared with this experimental protocol include SBI, SDAT, ORCA.

Although SBI is currently the most effective deepfake identification method, its recognition performance declines in the face of unknown forgery methods. SDAT adds GRL to the domain confrontation training framework and proposes a new target domain loss. In the baseline experiment, we carried out comparative experiments in our experimental protocol according to the way the author mentioned SDAT+MCC++CDAN in the paper. Comparing Table 7 and Table 8, we can observe that the model recognition ability of SBI and SDAT drops significantly when the target domain contains unknown deepfake methods in the source domain. ORCA classifies samples in an unlabeled dataset into previously seen classes, or forms a new class by clustering similar samples together. And extracting an adaptive threshold to solve the inter-class differences between known classes and unknown classes, it can be observed from Table 8 that ORCA is effective in identifying unknown forgery methods to a certain extent. Among the three experimental baselines in this experiment, only ORCA pays attention to the effective identification of open world unknown category data, and proposes corresponding methods for identifying open world unknown categories, so ORCA is the optimal baseline under this experimental protocol.

After the analysis of the results of this cross-domain experiment, as shown in Table 8, in each experiment under the experimental protocol, our method exceeds the accuracy rate of the existing SOTA method by 4%-6%. The experimental results can prove that our method can not only effectively identify known deepfake methods, but also has no significant decline in model recognition ability when faced with unknown deepfake methods in the open world. Also, the performance of the model will not be significantly degraded due to the inconsistent data distribution of the source domain and the target domain.

3) Experimental Analysis for Experimental Protocol 4: Compared with Experimental Protocol 3, the source domain and target domain deepfake methods of this experimental protocol are not exactly the same, but the distribution of source domain and target domain data is incomplete, that is, part of the target domain data is composed of source domain data. We hope to further verify our model in the open The ability to

TABLE VIII
RESULTS FOR CROSS-DATASET EXPERIMENT WITH INCOMPLETELY CONSISTENT DEEPFAKE METHODS, INCONSISTENT DATA DISTRIBUTION, SOURCE DOMAINS WITH LABELS AND TARGET DOMAINS WITHOUT LABELS

Method	F→N	N→F	F→C	C→F	N→C	C→N	F&C→N	N→F&C	C&N→F	F→C&N	F&N→C	C→F&N
SBI	61.7	65.4	63.5	62.6	67.3	60.7	65.9	63.6	67.4	60.3	68.9	59.3
SDAT	67.1	72.3	70.3	64.2	74.7	62.8	70.6	71.2	73.4	66.7	75.4	61.7
ORCA	86.5	90.2	88.6	87.3	91.0	85.4	88.3	89.9	91.5	84.6	92.3	83.5
Ours	**91.2**	**94.5**	**92.8**	**93.0**	**95.7**	**91.6**	**92.3**	**93.7**	**95.1**	**90.5**	**96.7**	**90.8**

TABLE IX
RESULTS FOR CROSS-DATASET EXPERIMENT WITH INCOMPLETELY CONSISTENT DEEPFAKE METHODS, CONSISTENT DATA DISTRIBUTION, SOURCE DOMAINS WITH LABELS AND TARGET DOMAINS WITHOUT LABELS

Method	F→F&N	N→F&N	F→C&F	C→F&C	N→N&C	C→N&C
SBI	63.5	66.1	65.9	65.1	68.9	61.6
SDAT	69.3	73.5	72.4	68.2	75.7	67.4
ORCA	88.6	91.4	89.1	88.7	91.8	86.1
Ours	**92.3**	**97.7**	**94.8**	**93.3**	**96.4**	**92.5**

identify known and unknown deepfake methods in the world. In the common data set of the source domain and the target domain, we select 12,000 images from the data of each forgery method and divide them into source domain training set, target domain training set, test reference set, and test set according to the ratio of 5/5/1/1. queryset. For the data set that only exists in the target domain, 7000 images are sampled from each new forgery method (except the common forgery method of the two data sets), and supplemented to the target domain training set and test reference set according to the ratio of 5/1/1. , test queryset. In this experimental protocol, the choice of our experimental baseline is the same as in Experimental Protocol 3.

As shown in Table 9, the ability of SBI to identify unknown forgery methods is the weakest among several baselines. The main reason is that SBI focuses on the robust detection of deepfake true-fake binary classification, and it is difficult to identify unknown forgery methods. While domain adaptation methods for SDAT do not focus on the detection of new classes. For the ORCA author to propose a corresponding solution for the new class recognition in the open world, but for our task our performance exceeds it, the main reason is that we more effectively deal with the unlabeled known and unknown class data in the target domain, our method is more effective for the difficult task of deepfake method identification.

After the analysis of the results of this cross-domain experiment, as shown in Table 9, in each experiment under the experimental protocol, our method exceeds the accuracy rate of the existing SOTA method by 4%-6%. For the real open world, forged images can contain known and unknown forgery methods, and can also come from the same or different original data distributions. This experiment fits this scenario. The experimental results can demonstrate that our method can effectively identify known and unknown deepfake methods in the open world.

F. Ablation

In order to verify that the adaptive clustering based on the network memory effect proposed in this paper is effective for the identification of deepfake methods, we conduct ablation experiments on NMEAC and PLG in this part of the ablation experiment. First, we cancel the feature clustering of the target domain data, and directly use the deepfake method recognition model M learned in the source domain to test and verify on the target domain data. We call this the ablation baseline. The second comparison method is to replace EMNAC and PLG in our method with DBSCAN to cluster and divide the target domain features and assign them corresponding pseudo-labels, so as to combine the source domain data and target domain data for model retraining. The protocol of all ablation experiments is not completely consistent according to the deepfake method, and the cross-dataset experiments with inconsistent data distribution are carried out from F→C

TABLE X
RESULTS FOR ABLATION EXPERIMENT

Method	Result
Baseline(w/o MR)	55.7
DBSCAN(w/o NMEAC+PLG)	70.2
Ours	**92.8**

The quantitative analysis of the ablation test results is shown in Table 10, and the qualitative analysis is shown in Figure 5. The experimental results show that the performance of the recognition model trained in the source domain directly drops sharply on the target domain (baseline). For the comparative experiment DBSCAN used for the ablation experiment, based on the density-based noise space clustering, we determined its hyperparameters Eps and MinPts through experiments, and clustered and divided the target domain data in the experiment. In general, we can see its clustering from the results The results are somewhat helpful to the model's ability to identify unknown forgery methods (compared to the baseline), but compared to our method, it is slightly insufficient to improve the model's ability to identify unknown forgery methods. As for the analysis of the above results, we can also get evidence from the T-SNE visualization results of the target domain features.

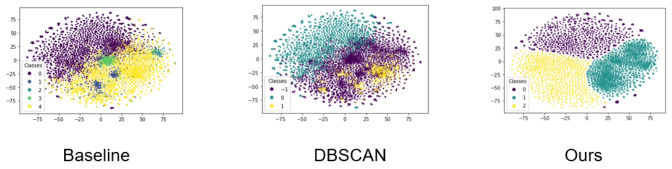


Fig. 4. T-SNE results for ablation experiment.The baseline model is directly tested in the target domain after pre-training in the source domain. DBSCAN replaces NMEAC and PLG in our method with DBSCAN.

V. Conlusion

In this paper, we propose a fine-grained deepfake recognition method based on unsupervised domain adaptation for unknown deepfake types prevalent in open scenes. The model-based network memory effect assigns the optimal pseudo-label to the unlabeled target domain data, and then retrains the pre-trained model in the source domain, so as to realize the effective recognition of the target domain data in the source domain. We verified the effectiveness of the proposed method on Forgertnet [1], FaceForensics++ [2], and FakeAVceleb [3] three deep fake datasets, and compared it with the SOTA method. Experimental results show that our method has better domain generalization ability than SOTA methods. Our method can effectively identify unknown deepfake methods in open world scenarios.

References

[1] He, Y., Gan, B., Chen, S., Zhou, Y., Yin, G., Song, L., ... & Liu, Z. (2021). Forgerynet: A versatile benchmark for comprehensive forgery analysis. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4360-4369).

[2] Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). Faceforensics++: Learning to detect manipulated facial images. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 1-11).

[3] Khalid, H., Tariq, S., Kim, M., & Woo, S. S. (2021). FakeAVCeleb: A novel audio-video multimodal deepfake dataset. arXiv preprint arXiv:2108.05080.

[4] Li, H., Li, B., Tan, S., & Huang, J. (2020). Identification of deep network generated images using disparities in color components. Signal Processing, 174, 107616.

[5] Yang, X., Li, Y., Qi, H., & Lyu, S. (2019, July). Exposing GAN-synthesized faces using landmark locations. In Proceedings of the ACM workshop on information hiding and multimedia security (pp. 113-118).

[6] Das, S., Seferbekov, S., Datta, A., Islam, M., & Amin, M. (2021). Towards solving the deepfake problem: An analysis on improving deepfake detection using dynamic face augmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3776-3785).

[7] Sun, Z., Han, Y., Hua, Z., Ruan, N., & Jia, W. (2021). Improving the efficiency and robustness of deepfakes detection through precise geometric features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3609-3618).

[8] Li, J., Xie, H., Li, J., Wang, Z., & Zhang, Y. (2021). Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6458-6467).

[9] Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., & Yu, N. (2021). Multi-attentional deepfake detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2185-2194).

[10] Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., & Guo, B. (2020). Face x-ray for more general face forgery detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5001-5010).

[11] Zhao, T., Xu, X., Xu, M., Ding, H., Xiong, Y., & Xia, W. (2021). Learning self-consistency for deepfake detection. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 15023-15033).

[12] Dong, X., Bao, J., Chen, D., Zhang, T., Zhang, W., Yu, N., ... & Guo, B. (2022). Protecting celebrities with identity consistency transformer. arXiv preprint arXiv:2203.01318.

[13] Shiohara, K., & Yamasaki, T. (2022). Detecting deepfakes with self-blended images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 18720-18729).

[14] Motiian, S., Piccirilli, M., Adjeroh, D. A., & Doretto, G. (2017). Unified deep supervised domain adaptation and generalization. In Proceedings of the IEEE international conference on computer vision (pp. 5715-5725).

[15] Rozantsev, A., Salzmann, M., & Fua, P. (2018). Beyond sharing weights for deep domain adaptation. IEEE transactions on pattern analysis and machine intelligence, 41(4), 801-814.

[16] Chopra, S., Balakrishnan, S., & Gopalan, R. (2013, June). Dlid: Deep learning for domain adaptation by interpolating between domains. In ICML workshop on challenges in representation learning (Vol. 2, No. 6). Citeseer.

[17] Li, Y., Wang, N., Shi, J., Liu, J., & Hou, X. (2016). Revisiting batch normalization for practical domain adaptation. arXiv preprint arXiv:1603.04779.

[18] Liu, M. Y., & Tuzel, O. (2016). Coupled generative adversarial networks. Advances in neural information processing systems, 29.

[19] Ganin, Y., & Lempitsky, V. (2015, June). Unsupervised domain adaptation by backpropagation. In International conference on machine learning (pp. 1180-1189). PMLR.

[20] Shen, J., Qu, Y., Zhang, W., & Yu, Y. (2018, April). Wasserstein distance guided representation learning for domain adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

[21] Glorot, X., Bordes, A., & Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In Proceedings of the 28th international conference on machine learning (ICML-11) (pp. 513-520).

[22] Sankaranarayanan, S., Balaji, Y., Castillo, C. D., & Chellappa, R. (2018). Generate to adapt: Aligning domains using generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8503-8512).

[23] Cao, K., Brbic, M., & Leskovec, J. (2021). Open-world semi-supervised learning. arXiv preprint arXiv:2102.03526.

[24] Guo, L. Z., Zhang, Z. Y., Jiang, Y., Li, Y. F., & Zhou, Z. H. (2020, November). Safe deep semi-supervised learning for unseen-class unlabeled data. In International Conference on Machine Learning (pp. 3897-3906). PMLR.

[25] Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., & Yu, N. (2021). Multi-attentional deepfake detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2185-2194).

[26] Pelleg, D., & Moore, A. (1999, August). Accelerating exact k-means algorithms with geometric reasoning. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 277-281).

[27] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30.

[28] Li, L., Bao, J., Yang, H., Chen, D., & Wen, F. (2019). Faceshifter: Towards high fidelity and occlusion aware face swapping. arXiv preprint arXiv:1912.13457.

[29] Deepfakes github. https://github.com/ deepfakes/faceswap. Accessed: 2018-10-29.

[30] Nirkin, Y., Keller, Y., & Hassner, T. (2019). Fsgan: Subject agnostic face swapping and reenactment. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 7184-7193).

[31] Fried, O., Tewari, A., Zollhöfer, M., Finkelstein, A., Shechtman, E., Goldman, D. B., ... & Agrawala, M. (2019). Text-based editing of talking-head video. ACM Transactions on Graphics (TOG), 38(4), 1-14.

[32] Chen, L., Maddox, R. K., Duan, Z., & Xu, C. (2019). Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 7832-7841).

[33] Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., & Sebe, N. (2019). First order motion model for image animation. Advances in Neural Information Processing Systems, 32.

[34] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8110-8119).

[35] Lee, C. H., Liu, Z., Wu, L., & Luo, P. (2020). Maskgan: Towards diverse and interactive facial image manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5549-5558).

[36] Choi, Y., Uh, Y., Yoo, J., & Ha, J. W. (2020). Stargan v2: Diverse image synthesis for multiple domains. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8188-8197).

[37] Jo, Y., & Park, J. (2019). Sc-fegan: Face editing generative adversarial network with user's sketch and color. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 1745-1753).

[38] Deng, Y., Yang, J., Chen, D., Wen, F., & Tong, X. (2020). Disentangled and controllable face image generation via 3d imitative-contrastive learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5154-5163).

[39] Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., & Nießner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2387-2395).

[40] Faceswap. https://github.com/ MarekKowalski/FaceSwap/. Accessed: 2018-10-29.

[41] Thies, J., Zollhöfer, M., & Nießner, M. (2019). Deferred neural rendering: Image synthesis using neural textures. Acm Transactions on Graphics (TOG), 38(4), 1-12.

[42] Chung, J. S., Nagrani, A., & Zisserman, A. (2018). Voxceleb2: Deep speaker recognition. arXiv preprint arXiv:1806.05622.

[43] Prajwal, K. R., Mukhopadhyay, R., Namboodiri, V. P., & Jawahar, C. V. (2020, October). A lip sync expert is all you need for speech to lip generation in the wild. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 484-492).

[44] Rangwani, H., Aithal, S. K., Mishra, M., Jain, A., Radhakrishnan, V. B. (2022, June). A closer look at smoothness in domain adversarial training. In International Conference on Machine Learning (pp. 18378-18399). PMLR.

[45] Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., Xie, S. (2022). A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11976-11986).

[46] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[47] Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[48] Robbins, H., Monro, S. (1951). A stochastic approximation method. The annals of mathematical statistics, 400-407.