A New Heterogeneous Mixture of Experts Model for Deepfake Detection

Qichang Wang, Ruixia Liu*
School of Mathematics and Statistics,
Shandong Artificial Intelligence Institute,
Qilu University of Technology (Shandong Academy of Sciences)
Jinan, China
10431220629@stu.qlu.edu.cn, liurx@sdas.org

Abstract

Deepfake detection remains a challenging task due to the diversity of forgery techniques and the distributional shifts between training and testing data. Existing methods, often framed as binary classification tasks, struggle with generalization, particularly in real-world scenarios involving unknown forgery types. To address this challenge, we propose DFMoE, a novel deepfake detection framework based on a Heterogeneous Mixture of Experts (HMoE) model. Our approach incorporates dynamic gating networks that adaptively select expert networks of varying capacities and scales according to the input characteristics, enabling precise identification of different types of forgeries. Leveraging a pre-trained face recognition model for multi-scale feature extraction, DFMoE combines expert specialization with adaptive data augmentation to enhance the detection of both known and unknown deepfake types. Experimental results show that our method significantly improves detection accuracy and robustness, offering a highly effective solution for deepfake detection across diverse scenarios.

Keywords: Deepfakes, Forgery Detection, Mixture of Experts, Gating Network, Robustness, Data Augmentation

1. Introduction

With the rapid advancement of deepfake technology, AI-generated highly realistic visual content has garnered widespread attention. Although Generative Adversarial Networks (GANs) have brought significant economic and entertainment value to the field of computer vision, their misuse poses serious concerns, such as violations of personal privacy, manipulation of public opinion, and threats to public safety. To address these challenges, developing deepfake detection methods with broad applicability has become a pressing need.

*Corresponding author.
E-mail address: liurx@sdas.org (R. Liu).

Initially, most detection methods treated deepfake detection as a simple binary classification task. While these methods perform well in detecting specific types of forgeries, their generalization ability in real-world scenarios is limited. In real-life applications, the training and test data often come from different distributions, and the feature distributions of various types of forgeries differ, causing these methods to exhibit significant performance drops in practical use, limiting their wide applicability.

In recent years, some methods have attempted to enhance generalization capabilities through pre-trained models, data augmentation, and more efficient model designs. Although these methods have made some progress in cross-domain detection, they often overfit certain forgery types. Moreover, many methods rely on a single encoder to parse different types of forgery data, but in real-world scenarios, numerous complex and unknown forgery types exist. The limitations of a single model make it difficult to effectively handle these complexities. Although some studies have designed multiple networks to handle different forgery types using strategies like knowledge distillation, these methods fail to dynamically select the most suitable network based on the differences in forgery types.

Another line of research has used data augmentation to generate more diverse and representative forgery samples, training more generalizable detectors. For example, some studies have generated new fake faces through pixel-level mixing, increasing the diversity of training samples and enhancing the robustness of detection models. Other studies have proposed semantic-based data augmentation strategies, manipulating semantic content such as lighting, facial expressions, and angles to generate forgery samples. Compared to simple pixel operations, semantic-level augmentation methods better simulate complex forgery scenarios.

While data augmentation methods show good potential for improving model generalization, their limitations are becoming apparent. Since current data augmentation methods mainly rely on combinations and transformations of existing forgery types, they cannot fully cover all possible unknown

forgery types.

To address the aforementioned issues, this paper proposes a deepfake detection method based on a Heterogeneous Mixture of Experts (HMoE)[36] model–DFMoE. Our design allows the system to adaptively select the most appropriate expert network based on the scale of different forgery features, thus improving detection accuracy and generalization capability.

Inspired by prior work, we also introduced a gating network based on a pre-trained face recognition model to capture rich facial features in real-world scenarios. Through the gating network, our method can adaptively select different expert networks for processing based on the characteristics of the forgery types.

In addition, we made innovations in data augmentation strategies. Existing forgery data features typically exhibit a discrete distribution in latent space, which fails to effectively cover the features of unknown forgery types. This shortcoming severely affects the model's ability to generalize in real-world applications. To overcome this problem, we enhance the features extracted by expert networks through data augmentation, enabling these enhanced features to better cover the feature domain of unknown forgery types. Specifically, we introduced latent space-based feature interpolation and expansion methods, generating more diverse feature representations by mixing and perturbing the features of known forgery types. These feature augmentation strategies significantly improve the classifier's generalization ability, enabling it to accurately handle previously unseen forgery types.

In summary, our method addresses the limitations of traditional deepfake detection approaches in terms of generalization by introducing the HMoE model and advanced data augmentation techniques. By adaptively selecting expert networks of different scales to handle forgery types, our method can not only cope with known forgery types but also possesses strong capabilities in handling unknown forgery types, achieving better performance with fewer activated parameters. Our research provides a novel approach and technical pathway for the field of deepfake detection, with broad application prospects. The contributions of our work are as follows:

1. We propose a deep false detection method based on the Heterogeneous Mixture of Experts model, named DFMoE, which incorporates a dynamic gating network that adaptively selects an appropriately sized expert network based on the characteristics of the input false samples.

2. Our method adaptively selects expert networks of different scales based on the differences in forgery characteristics, breaking the limitations of relying on a single network structure and enhancing the model's flexibility and adaptability.

3. We introduce adaptive data augmentation operations, where expert networks adaptively select appropriate augmentation methods, significantly improving detection accuracy and generalization capability.

4.Through extensive experimental validation, our method outperforms existing methods on multiple datasets, demonstrating significant improvements.

2. Related Work

In this section, we briefly review deepfake detection methods, which are predominantly categorized into two areas: image spatial domain-based detection methods and image frequency domain-based detection methods.

2.1. Image Spatial Domain-Based Detection Methods

Some early studies have achieved relatively good results in the field of binary classification, and some backbones have been proposed with good results, such as Xception [7] and EfficientNet [33]. However, these backbones are not specifically designed for forgery detection tasks. When the model detects false images synthesized by unknown forgery methods, the accuracy of the detection decreases rapidly, and the robustness is insufficient when facing common perturbation methods such as image compression. Zhao [41] believed that the difference between real and fake faces mainly exists in subtle local areas, so he proposed a new multi-attention Deepfake detection Network, which enhances the texture information extracted by the model in the shallow network and fully excavates the subtle texture artifacts. However, the above methods often overfit and produce artifacts of specific forgery methods. In order to avoid overfitting specific forgery methods, Li [20] proposed the Face X-ray model to make judgments by detecting the edge splicing area between the forged face and the background. Relying on the detection of artifact traces of specific forgery methods, it has good generalization ability across data sets. Unlike Face X-Ray, which needs to rely on other faces with similar key points to generate fake faces, Shiohara [32] learned feature-representations with stronger generalization capabilities, and propose self-blended images (SBI) as synthetic fake images. It only transforms the key points of the face image itself, and at the same time cooperates with the data augmentation method to generate realistic Fake face images. These methods do not depend on using fake faces from a specific forgery technique to train the network. As a result, they exhibit strong generalization abilities when confronted with synthetic images generated by unknown forgery methods. However, due to their reliance on the self-forgery process, these approaches prove ineffective in dealing with fake images synthesized through unknown forgery methods. It performs poorly on fake images synthesized by whole face synthesis methods. In addition, Cao [5] proposed a RECCE framework for face forgery detection, which uses reconstruction classification to mine common features of real faces,

and proposed a reconstruction-guided attention module that uses the difference between the reconstructed image and the original image as an attention map for Guide the model toward areas more likely to be tampered with. There is also some work on the interpretability of detection models. For example, Dong [11] assumed that the detection model determines the authenticity of an image by detecting information unrelated to the person's identity in the image. Therefore, the face identity is used as an auxiliary label to design a source feature encoder and The target encoder performs the identity recognition task, and the FST-Matching deep fake detection model is proposed to decouple the feature representations in the image that are relevant and irrelevant to the person identity recognition task, and improve the fake detection performance of compressed videos.

2.2. Image Frequency Domain-Based Detection Methods

Detection methods based on the image frequency domain mainly focus on mining high-frequency signals, phase spectrum, etc. in the image frequency signal, and use frequency domain features or fusion features of the frequency domain and spatial domain to detect deep facial forgery videos. For example, Qian [27] found that the artifact details caused by forgery methods can be well mined in the frequency domain; in order to obtain comprehensive frequency domain information, a frequency domain perceptual decomposition module was designed to adaptively capture the artifacts in the image. Forgery clues, since detection relies on frequency domain information, this method still maintains excellent detection performance in the face of highly compressed forgery images. Liu [23] found that cumulative upsampling will lead to significant changes in the frequency domain, especially the phase spectrum, so they proposed a new spatial-phase shallow learning (SPSL) method, which combines spatial images and phase spectrum to capture the upsampling artifacts of face forgery to improve transferability for face forgery detection. In order to more comprehensively capture artifacts in the frequency domain, Li [23] proposed an adaptive frequency feature generation module to extract differential features from different frequency bands in a learnable manner. At the same time, considering the different feature distributions of different forgery methods, single center loss (SCL) is proposed to improve the intra-class compactness of real faces and increase the inter-class difference between real faces and fake faces.

The above methods mainly use image frequency domain features for deep fake video detection, but ignore the pixel features of the original spatial domain features. Therefore, combining frequency domain and spatial domain features can effectively make up for the shortcomings of both. Therefore, Gu [14] proposed a progressive reinforcement learning framework to utilize RGB and fine-grained frequency cues to perform fine-grained decomposition of RGB images to completely decouple real and false trajectories in frequency space. Wang [38] proposed a multi-modal approach that combines frequency domain and spatial domain to mine robust forgery traces in images that do not change due to different forgery techniques. Chen [6] divided the original image/video frame into several areas. Taking into account the small difference between real areas and the large gap between real areas and fake areas, based on dividing the original image into several areas, The difference between two areas is calculated from both frequency domain features and spatial domain features to determine the authenticity of the video. Recently, Yan [39] proposed a simple yet effective detector to expand the forgery space by constructing and modeling the variations within and between forged features in the latent space, thus achieving a generalizable deepfake detector.

2.3. Mixture of Experts (MoE)

The core concept of the MoE[16] method lies in the introduction of multiple expert networks, enhancing the flexibility and adaptability of the model. Each expert network is specifically trained for a particular forgery type and possesses unique feature extraction capabilities. Through a gating mechanism, the model dynamically selects the most suitable expert network based on the characteristics of the input samples, significantly improving detection accuracy and generalization ability. The concept of MoE was first introduced in natural language processing [30] [18][4]and computer vision[28][24][25][31]. Jacobs[16] initially proposed this supervised learning process, in which a system contains multiple independent networks, each processing a subset of all training samples. Shazeer[30] later discovered that not all expert networks are used—only a few experts participate in inference—thus greatly increasing model scalability with minimal computational overhead. Lepikhin[18] extended MoE to Transformers, and Fedus[13] simplified MoE routing algorithms, designing a more intuitive model improvement scheme that reduces communication and computation costs. Recently, Wang[36]proposed the Heterogeneous Mixture of Experts (HMoE) model, where experts are of different scales. This heterogeneity allows more specialized experts to effectively handle complex features. In our work, we draw inspiration from the concept of HMoE, incorporating a gating network to intelligently and adaptively select expert networks of different scales based on the differences in forgery features, thus achieving efficient deepfake detection.

3. Method

We propose a novel architecture, DFMoE, based on Heterogeneous Mixture of Experts, which significantly differs from traditional training architectures. DFMoE consists of several key components: a fusion gating network, expert networks, and a data augmentation module. We design a

multi-scale feature extraction and dynamic expert selection mechanism based on the MoE model, combined with both within-domain and cross-domain data augmentation strategies, to enhance the detection performance of both known and unknown types of forgeries.

3.1. Overall Architecture

Figure 1 illustrates the overall architecture of the proposed DFMoE model. The input images are processed through a pre-trained Xception network to extract features, serving as a shared feature extractor responsible for generating high-dimensional facial representations. These features are then embedded into an N-dimensional expert space via a multi-layer perceptron (MLP). Following this, the gating network dynamically selects the most suitable combination of expert networks (EfficientNet sub-networks), each learning forgery-specific features at different scales. Finally, after passing through a fusion module and a data augmentation module, the features are fed into a fully connected (FC) layer for binary classification of real vs. fake.

3.2. Gating Network

The gating network is the core mechanism of DFMoE, responsible for dynamically assigning expert networks based on the input sample's features. The gating network consists of two main parts: multi-scale feature selection and dynamic expert assignment. These components ensure that the system efficiently captures different types of deepfake features.

3.2.1 Multi-Scale Feature Selection

To capture multi-scale features from forgery samples, we use the pre-trained Xception network to extract features from various scales. These features are processed by the MLP to generate a weight matrix W , which guides the gating network in weighting and selecting features from different scales.

Given an input image i, we define the feature extractor $f_{\text{Xception}}(I)$, which produces feature maps at N_{scales} different scales:

$$\mathbf{F}_i = f_{\text{Xception}}(\mathbf{I})_i, i = 1, \ldots, N_{\text{scales}} \tag{1}$$

These feature maps represent the input image's representations across different scales. Next, the MLP generates a corresponding weight vector W for each feature map:

$$\mathbf{w}_i = MLP(\mathbf{F}_i), i = 1, \ldots, N_{scales} \tag{2}$$

These weights measure the importance of each scale's features, determining which features are routed to the expert networks.

3.2.2 Dynamic Expert Assignment

We adopt a top-K routing strategy to select the most relevant expert networks, where K is a hyperparameter controlling the number of experts selected by the gating network at each time. Through experimentation, we found that setting K=2 provided the best performance, leading us to use a top-2 routing strategy.

For each input sample, the gating network selects the top K most important features based on the weight matrix w and assigns the sample to the corresponding expert networks. Specifically, the gating network sorts the weights and selects the top-K features with the highest weights, then routes the corresponding features to the appropriate expert networks for processing.

Assuming that the gating network outputs a feature weight matrix $w = \left[w_1, w_2, \ldots, w_{N_{\text{experts}}}\right]$, we select the top K experts with the highest weights.

$$\text{Top} - \text{K experts} = \text{argmax}_K(\mathbf{w}) \tag{3}$$

In this study, the value of K is set to 2 based on experimental results, meaning each sample is assigned to two expert networks, which constitutes our top-2 routing strategy.

3.3. Expert Networks

Expert networks are the core modules in our DFMoE model, designed to handle different types and scales of forgery features. Based on the characteristics of the input sample, these expert networks are dynamically assigned, allowing them to process various types of counterfeit, thus improving detection accuracy and generalization ability.

3.3.1 Heterogeneous Expert Network Architecture

In the DFMoE model, we use the EfficientNet series (EN-B1 to EN-B5) as the heterogeneous expert network architecture. Each expert network processes the selected features from the gating network and extracts deep-level forgery-related features. These expert networks are specifically optimized for different forgery types and feature scales, forming a heterogeneous expert system. Specifically, given the input features $\mathbf{F}_{\text{selected}}$, chosen by the gating network, each expert network E_i extracts its corresponding high-dimensional feature representation:

$$\mathbf{z}_i = E_i(\mathbf{F}_{\text{selected}}), i = 1, \ldots, N_{\text{experts}} \tag{4}$$

where $\mathbf{z}_i$ represents the output feature vector of the i-th expert network.

3.3.2 Fusion of Expert Networks

To effectively integrate the outputs from multiple expert networks, we introduce a feature fusion module. By performing

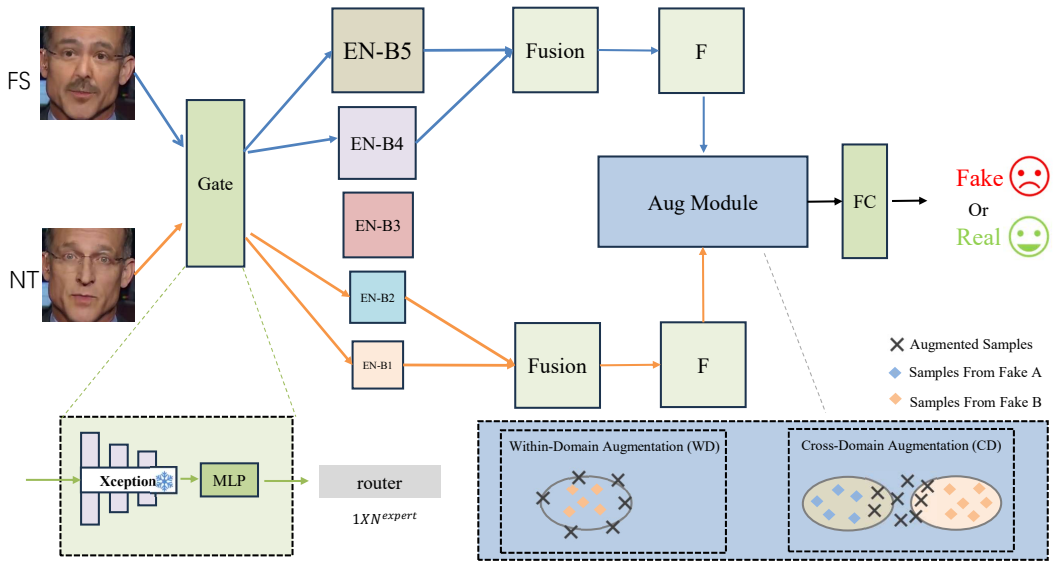

Figure 1. The overall pipeline of our proposed method (two fake types are considered as an example). (1) Multi-scale feature extraction is performed using a pre-trained Xception network, with the early layers of the network frozen. (2) Expert networks of different scales are adaptively selected for training. (3) For the learning of the forgery feature, we apply the within-domain(WD) and cross-domain (CD) augmentation.

weighted summation or concatenation of the output features from the experts, we obtain a more representative global feature representation, thereby improving the overall performance of forgery detection. We adopt a weighted feature fusion strategy, combining the output feature vectors from the selected experts. Suppose the outputs from the top-K experts selected by the gating network are $z_1, z_2, \ldots, \mathbf{z_K}$, then the fused feature is defined as:

$$\mathbf{F}_{\text{fused}} = \sum_{i=1}^{K} \alpha_i \mathbf{z}_i \tag{5}$$

where α_i is the weight assigned to each expert network, calculated from the feature weight matrix W output by the gating network. This fusion approach leverages the expertise of each expert, combining the diversity of forgery features to effectively enhance the robustness and accuracy of forgery detection.

3.4. Adaptive Augmentation Module

In this study, we propose an adaptive augmentation module that dynamically determines the applicability of augmentation strategies based on the feature entropy of the expert network outputs and the network selection mechanism. For within-domain (WD) and cross-domain (CD) augmentation, we define three main augmentation strategies:

Feature Stretching Augmentation (FSA) : This strategy expands the distances between features to generate more challenging samples. Specifically, the operation is as follows:

$$z_{aug} = z + \omega_j \cdot \alpha \cdot (z - \mu_i) \tag{6}$$

where α is a random coefficient, μ_i represents the center of the i-th forgery domain, and ω_j is the adaptive weight.

Feature Perturbation Augmentation (FPA): This strategy adds noise perturbations in feature space to improve robustness. It can be achieved by adding Gaussian noise:

$$z_{aug} = z + \omega_i \cdot \beta \cdot \mathcal{N}(0, \sigma^2) \tag{7}$$

where β is a scaling factor, and $\mathcal{N}$ is Gaussian noise.

Cross-expert Feature Fusion (CFF): This strategy generates cross-expert augmented samples by performing linear interpolation between features from different experts. Let $f_i(x)$ and $f_j(x)$ represent the feature outputs of two different experts, the linear interpolation is computed as:

$$z_{aug} = \lambda f_i(x) + (1 - \lambda) f_j(x) \tag{8}$$

where λ is a randomly selected weight between 0 and 1. During the augmentation process, for each input sample, we first compute its feature output through the expert network f_i, and based on the gating network's selection and the feature entropy, we compute the adaptive weight w.The selection of the augmentation strategy no longer solely depends on feature entropy but is complemented by the output of the gating network, ensuring that the augmentation strategy and expert network selection mechanism work in tandem. The final augmented sample can be represented as:

$$z_{aug} = \text{AdaptiveAug}(f_i(x), \omega_i) \tag{9}$$

where AdaptiveAug represents a combination of the aforementioned three augmentation strategies, dynamically chosen and applied based on the adaptive weight ω_i.

3.5. Loss Function

To optimize the overall performance of the deepfake detection model, we designed a comprehensive loss function. This loss function consists of three components: binary classification loss $\mathcal{L}_{BCE}$, feature entropy-based expert assignment constraint loss $\mathcal{L}_{entropy}$, and augmentation consistency loss $\mathcal{L}_{consistency}$.

3.5.1 Parameter Penalty Loss

To prevent the model from excessively relying on large-scale experts during the expert selection process, which could lead to under-utilization of smaller-scale experts, we introduce a parameter penalty loss. This loss is applied based on the hidden state size of the experts, ensuring that smaller-scale experts are activated for appropriate tasks, thereby avoiding resource waste. The specific loss is defined as follows:

$$\mathcal{L}_{P-Penalty} = \frac{1}{T} \sum_{i=1}^{N} M_i \cdot \widehat{P}_i \tag{10}$$

where $\mathcal{M}_i$ represents the average hidden state dimension of expert i , and i is the activation probability of that expert. By penalizing the use of experts with larger hidden dimensions, the model is encouraged to more efficiently utilize smaller-scale experts.

3.5.2 Expert Assignment Constraint Loss

To optimize the expert network selection and reduce redundancy among expert networks, we introduce an expert assignment constraint loss based on feature entropy. Feature entropy measures the uncertainty or complexity of the features, where high entropy indicates more challenging classification, and low entropy suggests simpler classification. By minimizing feature entropy, the model is encouraged to assign experts appropriately based on the complexity of the input sample. This loss increases the entropy of the expert selection process to prevent the model from over-relying on a particular set of experts. The entropy loss is defined as:

$$\mathcal{L}_{entropy} = -\frac{1}{N} \sum_{i=1}^{N} P_i \cdot \log(P_i) \tag{11}$$

whereP_i is the selection probability of expert. By minimizing low entropy cases in the expert selection process, the model can more evenly distribute tasks among multiple experts, thereby improving generalization capability.

3.5.3 Augmentation Consistency Loss

To ensure consistency in the feature space between the original and augmented samples generated by expert networks, we propose an augmentation consistency loss $\mathcal{L}_{consistency}$. This loss is designed to constrain the augmented samples to retain the characteristics of the original features. Specifically, the consistency loss is calculated by comparing the Euclidean distance between the original sample features and the augmented sample features:

$$\mathcal{L}_{consistency} = \frac{1}{N} \sum_{i=1}^{N} \parallel f(x_i) - f(z_{aug}) \parallel_2^2 \tag{12}$$

where $f\left(x_i\right)$ denotes the feature representation of the original sample, and $f\left(z_{aug}\right)$ represents the feature representation of the augmented sample. By minimizing this loss, we ensure that the augmented samples do not deviate from the original feature distribution, thereby enhancing the effectiveness of the augmentation strategies in the model.

3.5.4 Total Loss

Finally, the total loss function of the model is composed of the weighted sum of the above loss terms:

$$\mathcal{L}_{total} = \mathcal{L}_{BCE} + \lambda_1 \mathcal{L}_{P-Penalty} + \lambda_2 \mathcal{L}_{entropy} + \lambda_3 \mathcal{L}_{consistency} \tag{13}$$

where λ_1, λ_2 and λ_3 are hyperparameters that control the weights of each loss term. By adjusting these weights, the contributions of each loss to the total loss can be balanced, allowing for flexible control over expert assignment, diversity, and augmentation consistency during the optimization process.

4. Experiments

4.1. Settings

Dataset. To evaluate the generalization ability of our proposed framework, we conducted experiments on several commonly used deepfake datasets, including: Faceforensics++ (FF++)[29], Celeb-DF-v1 (CDFv1) [22], Celeb-DF-v2 (CDFv2)[22], Faceshifter (Fsh) [19], DeepfakeDetection (DFD)[12], Deepfake Detection Challenge (DFDC)[9] and its preview version (DFDCP)[10], and UADFV[21]. The FF++ dataset comprises 4,000 fake videos and 1,000 real videos, with the fake images generated using four facial forgery algorithms: DeepFakes (DF)[1], Face2Face (F2F)[35], FaceSwap (FS)[2], and NeuralTexture (NT)[34]. FF++ provides three versions with different compression rates: raw, high quality (c23), and low quality (c40). We selected the high-quality version (c23) for training and testing to ensure that the data's visual quality closely aligns with real-world application scenarios. The CDFv1 dataset contains 795 fake videos and 408 real videos generated from celebrity interview footage using DeepFake technology. CDFv2 is an upgraded version that adds 590 original videos and 5,639 corresponding fake videos sourced from

YouTube, covering diverse ages, races, and genders. CDFv2 is currently one of the most challenging datasets for deepfake detection, with its fake videos being visually more difficult to distinguish. The Fsh dataset consists of 1,000 fake videos generated from real videos in FF++. The DFD dataset includes 363 real videos and 3,068 fake videos, employing various generation techniques that exhibit high data quality and diverse scenes. The DFDC dataset contains over 100,000 fake videos and 20,000 real videos, encompassing a wide range of forgery techniques and different contexts, significantly increasing the generalization requirements for models in real-world applications. DFDCP is an early version of the DFDC challenge, which, although smaller in size, still contains videos generated by multiple forgery techniques. The UADFV dataset consists of 49 real videos and 49 fake videos, sourced from YouTube, and is one of the earlier research datasets in the field of deepfake detection.

Evaluation Metrics. By default, we adopted the area under the ROC curve (AUC) and equal error rate (EER) as evaluation metrics. AUC measures the area under the Receiver Operating Characteristic (ROC) curve, while EER represents the false positive rate (FPR), equal to the true positive rate (TPR).

Implementation Details. We employed a pre-trained Xception model as the feature extraction network for the gating mechanism, while using an Efficient system network as the expert network. During the data preprocessing stage, Dlib[17] was utilized for face detection, cropping, and alignment. All face images were resized to 256x256. For both training and testing phases, we used the Adam optimizer with a learning rate of 0.0002. The batch size was set to 32, and each video was sampled with 32 frames for training and testing purposes. We set β in Eq. 7 as 0.2. In the loss function formulation, the hyperparameters λ_1,λ_2, and λ_3 were set to 0.1, 0.1, and 0.05 in Eq. 13, respectively. All models were implemented using the PyTorch framework and were trained on NVIDIA Tesla A100 GPUs.

4.2. Performance Evaluation

4.2.1 Within-Domain Evaluations

To further assess the performance of our proposed DFMoE model within the same dataset, we conducted within-domain evaluation experiments. The model was trained and tested on different forgery types within the FF++(c23) dataset, covering four forgery methods: FF-DF, FF-F2F, FF-FS, and FF-NT. The experimental results are presented in Table 1, with AUC as the evaluation metric.

In Table 1, our method demonstrates outstanding performance across various subtasks of the FF++(c23) dataset. Specifically, DFMoE achieved AUC values of 99.56%, 99.23%, 99.51%, and 97.62% for FF-DF, FF-F2F, FF-FS, and FF-NT, respectively, surpassing all baseline models. In contrast, traditional models like Meso4 and Capsule performed poorly, with AUC values below 70% in all tasks. More mature models such as Resnet34 and Xception performed well but still showed some gap compared to our DFMoE model. This indicates that DFMoE, through its dynamic expert selection and adaptive enhancement strategies, can more accurately detect different types of forgery videos, demonstrating higher generalization ability and robustness. Overall, our method exhibits a clear advantage in handling complex forgery types, especially in the FF-DF and FF-FS tasks, where AUC exceeded 99.5%, validating the superior performance of DFMoE in deepfake detection tasks.

4.2.2 Cross-Domain Evaluations

To verify the model's generalization capability across datasets, we conducted cross-domain evaluation experiments. All models were trained on the FF++(c23) dataset and tested on multiple public deepfake detection datasets, including CDFv1, CDFv2, Fsh, DFD, DFDC, DFDCP, and UADFV. The evaluation metric was AUC, which measures the model's classification performance across different datasets. The results are shown in Table 2.

Table 2 presents the cross-domain evaluation results of different methods across various datasets. It is evident that performance varies significantly across datasets for different methods. Early models such as MesoNet and Capsule performed poorly in cross-domain tests, with AUC values generally below 70%, indicating that these models struggle to adapt to different types of deepfake data. Resnet34 and Xception, as stronger baseline models, performed reasonably well on several datasets, particularly on DFD and DFDC, with AUC of 72.65% and 81.77%, respectively. EfficientB4 performed notably well on datasets outside of FF++, achieving an AUC of 95.27% on the UADFV dataset, indicating good generalization capability for handling deepfakes from different domains. Among all tested methods, our model demonstrated strong generalization ability across most datasets, especially on CDFv1, CDFv2, DFD, and UADFV, where our method achieved AUC values of 83.25%, 79.32%, 81.92%, and 95.18%, respectively, significantly outperforming other comparison methods. Compared to other models, our DFMoE method is better at handling various types of deepfake data, mainly due to the proposed dynamic expert selection mechanism and adaptive enhancement strategies.

4.2.3 Robustness Experiments

In the robustness experiments, the aim was to evaluate the model's performance under different image perturbation conditions, especially when the input images are subjected to various common attacks or transformations (e.g., Gaussian blur, block perturbation, contrast change, saturation change, and JPEG compression). The results demonstrate how the model's performance changes across these scenarios. As

Table 1. Comparison of within-domain experimental results with state-of-the-art methods regarding the AUC (%) metric. Bold and underlined values correspond to the best and the second-best values.

Method	Backbone	FF++(c23)	FF-DF	FF-F2F	FF-FS	FF-NT
Meso4[3]	MesoNet	62.69	75.06	62.10	58.64	57.91
Capsule[26]	Capsule	68.14	74.82	69.99	65.67	68.81
Resnet34[15]	Resnet	97.80	99.02	98.87	99.16	96.19
Xception[7]	Xception	98.23	99.15	99.14	99.33	96.77
EfficientB4[33]	Efficient	97.62	98.68	98.67	99.24	96.08
Face X-ray[20]	HRNet	95.26	97.98	99.04	99.15	95.59
FFD[8]	Xception	98.24	99.01	98.94	99.29	96.79
Recce[5]	Xception	97.98	99.08	99.05	99.34	96.13
RFM[37]	Xception	97.57	98.89	98.44	99.18	96.03
UCF[40]	Xception	98.50	99.29	<u>99.18</u>	99.40	96.67
LSDA[39]	Efficient	<u>99.05</u>	<u>99.37</u>	99.17	<u>99.47</u>	<u>97.41</u>
Ours	Efficient	**99.15**	**99.56**	**99.23**	**99.51**	**97.62**

Table 2. Comparison of cross-domain experimental results with state-of-the-art methods regarding the AUC (%) metric. Bold and underlined values correspond to the best and the second-best values.

Method	Backbone	CDFv1	CDFv2	Fsh	DFD	DFDC	DFDCP	UADFV
Meso4[3]	MesoNet	65.67	64.33	58.62	54.59	56.51	57.98	71.50
Capsule[26]	Capsule	65.59	67.48	63.06	65.65	63.89	64.89	91.60
Resnet34[15]	Resnet	78.59	76.17	61.47	72.65	71.53	72.97	90.05
Xception[7]	Xception	74.36	74.93	66.73	<u>81.77</u>	70.46	75.32	95.30
EfficientB4[33]	Efficient	81.03	77.60	67.56	69.54	63.51	71.01	<u>95.37</u>
Face X-ray[20]	HRNet	71.62	68.32	67.98	76.65	61.87	69.81	90.11
FFD[8]	Xception	78.49	77.70	65.88	80.12	69.45	76.55	94.32
Recce[5]	Xception	79.78	75.25	66.36	80.15	71.18	73.53	93.65
RFM[37]	Xception	78.79	75.17	61.62	78.55	68.16	72.71	93.70
UCF[40]	Xception	74.51	76.42	71.11	80.22	73.16	77.86	95.22
LSDA[39]	Efficient	<u>82.13</u>	<u>78.04</u>	<u>72.17</u>	81.75	<u>73.74</u>	<u>78.17</u>	95.17
Ours	Efficient	**83.25**	**79.32**	**73.10**	**81.92**	**73.95**	**78.51**	**95.58**

shown in the figure, our proposed method demonstrates remarkable robustness compared to baseline models such as EfficientB4, Xception, Face X-ray, and RECCE.

Gaussian Blur: Even under severe blur levels (Level 3 and Level 4), our model maintains high AUC values, outperforming all baselines.

Block-Wise Perturbations: Our method experiences minimal performance degradation compared to others.

Contrast and Saturation Changes: The model remains consistently resilient across all levels of these perturbations, maintaining stable AUC values.

JPEG Compression: While most baseline methods show significant performance drops at Level 4, our model achieves the highest robustness, effectively addressing real-world compression artifacts.

thanks to our multi-scale feature selection mechanism and data augmentation strategies, our model exhibits superior robustness under all types and intensities of perturbations, particularly under high-level perturbations. This demonstrates the strong adaptability of our model in real-world scenarios where image quality may degrade.

4.3. Ablation Studies

In this section, we discuss various techniques and methods to evaluate the performance and effectiveness of the proposed detection model. Through both quantitative and qualitative assessments, we delve into the underlying mechanisms of the model, focusing on the impact of different loss functions and network architectures on model performance.

4.3.1 Effect of Different Loss Functions

Our proposed method incorporates several key loss functions: parameter penalty loss, expert assignment constraint loss, and data augmentation consistency loss. To verify the impact of each loss, we first construct a baseline model A, which only utilizes the extracted features for detection through a classification head, without any additional loss

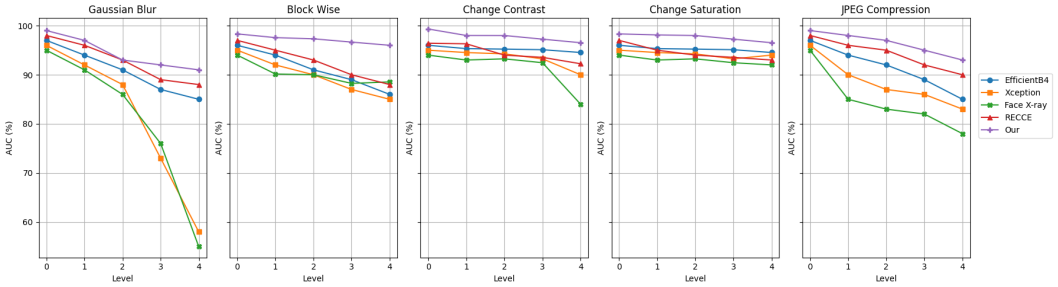

Figure 2. Robustness against unseen perturbations: We report the video-level AUC (%) across five specific types of perturbations at five different degradation levels, comparing our results with four prior methods to demonstrate our robustness.

terms. Then, we develop several variant models to explore the impact of different combinations of loss functions on model performance: (1) Model B: includes only the parameter penalty loss; (2) Model C: includes both the parameter penalty loss and the expert assignment constraint loss; (3) Model D: includes the parameter penalty loss, expert assignment constraint loss, and data augmentation consistency loss. All models are trained on the FF++ (c23) dataset and tested on the FF++ (c23), CDFv1, and CDFv2 datasets. We use AUC(%) as the evaluation metric, and the results are shown in Table 3.

As shown in Table 3, the baseline model A achieves an AUC of 98.66% on the FF++ dataset, but its performance in cross-domain testing (DFDCP and CDFv2) is relatively poor. After incorporating the parameter penalty loss in model B, the performance on FF++ improves slightly, with a significant improvement in cross-domain performance. Model C, which further incorporates the expert assignment constraint loss, shows additional improvements in cross-domain generalization. Finally, Model D, which combines all losses, achieves the best performance across all datasets, especially in cross-domain tasks. The results indicate that introducing the parameter penalty loss, expert assignment constraint loss, and data augmentation consistency loss is crucial for enhancing the model's cross-domain robustness, particularly on the CDFv2 and DFDCP datasets. The combination of these three losses significantly improves the model's generalization ability.

Table 3. We trained various baseline models on the FF++ (c23) dataset and tested them on the FF++, DFDCP, and CDFv2 datasets, with the AUC(%) metric.

Model	$\mathcal{L}_{P-Penalty}$	$\mathcal{L}_{entropy}$	$\mathcal{L}_{consistency}$	FF++	DFDCP	CDFv2
A				98.66	76.58	72.49
B	✓			98.95	77.25	75.62
C	✓	✓		99.06	77.61	78.46
D	✓	✓	✓	**99.15**	**78.51**	**79.32**

4.3.2 Exploring Different Backbones in Gate Network

To further enhance the feature extraction capabilities of our model, we explore the impact of different backbones on model performance. We selected ResNet34, EfficientNet-B4 (EN-b4), and Xception models, trained them on the FF++ (c23) dataset, and evaluated them on both in-domain and cross-domain datasets. The evaluation metrics are AUC (%) and EER (%), and the results are shown in Table 4. The experimental results demonstrate that Xception achieves the best performance in both in-domain and cross-domain tests, particularly excelling in cross-domain generalization. This indicates that using more complex feature extraction networks, such as Xception, can better capture subtle features in deepfake data, thereby enhancing cross-domain detection robustness.

Table 4. Exploring different backbones in Gate Network. All models are evaluated on the in-domain FF++(c23) dataset and cross-domain CDFv2, DFDCP, and DFD datasets regarding AUC(%) and EER(%).

Backbone	FF++		CDFv2		DFDCP		DFD	
	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)
ResNet34	98.32	3.45	74.54	34.81	74.29	32.62	77.54	28.49
EN-b4	99.02	3.28	76.62	31.95	77.15	28.22	80.49	24.63
Xception	**99.15**	**2.56**	**79.32**	**26.56**	**78.51**	**26.46**	**81.92**	**23.55**

4.3.3 The Impact of the K Hyperparameter

To verify the role of the K value in expert network selection, we designed a series of experiments to assess the impact of different K values on model performance. We trained models on the FF++ (c23) dataset and tested them on the CDFv2 dataset, using AUC, AP, and EER as evaluation metrics. All experiments were conducted within the same DFMoE framework, keeping the model architecture and training parameters consistent. The results are shown in Table 5. As shown in Table 5, the model's performance varies with changes in the K value. When K = 2, the model achieves the best AUC, AP, and EER across both in-domain and cross-domain tasks, indicating that expert network selection is most effective

Table 5. Performance Metrics for Different K Values

K Value	FF++(c23)			CDFv2		
	AUC(%)	AP(%)	EER(%)	AUC(%)	AP(%)	EER(%)
1	98.45	99.23	2.67	77.59	86.62	27.62
2	**99.15**	**99.52**	**2.56**	**79.32**	**88.41**	**26.56**
3	99.04	99.46	2.95	78.62	88.16	27.49
4	98.89	99.30	3.15	78.15	87.26	28.10
5	98.40	98.85	3.16	77.34	86.14	28.12

with K = 2, allowing the model to better handle diverse feature distributions. When K becomes too large (e.g., K = 4 or K = 5), model performance declines slightly, likely due to an increase in model complexity, leading to overfitting. Moreover, in in-domain evaluations, performance fluctuations across different K values are minor, suggesting that the K value has less impact on relatively simple in-domain tasks. However, in cross-domain tasks, the influence of K is more significant, highlighting the importance of selecting the appropriate number of expert networks in cross-domain tasks. Considering both model performance and computational cost, we ultimately select K = 2 as the hyperparameter for expert network selection.

4.3.4 Impact of the total number of experts on performance and computational cost.

We further systematically analyze the impact of the total number of experts on performance and computational cost. We train on the FF++ dataset and test on the CDFv1 dataset. The evaluation indicators are AUC, ACC, inference time (ms), and FLOPs (GFLOPs) when increasing or decreasing the total number of experts.

As can be seen from Table 6, AUC and accuracy improve with the increase in the number of experts, indicating that more experts can better capture complex forgery features, and the corresponding inference time and computational cost are also increasing. However, it is worth noting that when the number of experts exceeds 5, the performance improvement gradually slows down, so when we set the total number of experts to 5, it may be the best compromise between performance and computational efficiency, especially in resource-constrained environments.

In addition, in order to analyze the feasibility of deploying the model in a resource-constrained environment, we conducted a comparative experiment. Compared with the baseline, our method increased FLOPs by 14 GFLOPs, and the inference time increased by 20(%) accordingly. Despite the increase in computational overhead, the experimental results show that our proposed method can still maintain good performance in a resource-constrained environment, especially under reasonable optimization. In addition, we also discussed possible optimization directions, such as using lightweight expert networks or model pruning techniques to reduce computational costs. We expect that further optimization of these parts can effectively reduce computational overhead while ensuring performance and adapt to more resource-constrained scenarios.

4.3.5 Impact of different expert network fusion strategies on performance.

In order to further analyze the impact of different expert network fusion strategies on the experimental results, we used three fusion methods for comparison, including weighted feature fusion, maximum fusion and average fusion. We chose to train and test on the FF++ dataset.

The results are shown in Table 7. The weighted feature fusion method achieved the best performance, the maximum fusion method had a slight decrease in performance, and the AUC reached 99.02(%), while the average fusion method had the worst effect, only achieving 98.57(%) AUC. To further explore the reasons for the performance differences, we analyzed the feature distribution and model decision mechanism under different fusion strategies. Weighted feature fusion can dynamically adjust the weights according to the output contributions of different expert models, so as to more effectively aggregate information of different scales and different feature spaces, so it performs best in a variety of forgery detection tasks. Although the maximum fusion strategy can capture some significant features, it is easy to ignore some critical information with small contributions, resulting in a slight decrease in performance. The average fusion strategy gives the same weight to all expert model outputs, which cannot fully reflect the heterogeneity advantages of different expert networks, so it performs the worst.

4.3.6 The impact of different modules within the framework on performance.

To further verify the contribution of the enhancement module and the expert network to the model performance, we designed three sets of comparative experiments, including three configurations: "enhancement module + expert network", "enhancement module only" and "expert network only", and trained and tested them on the FF++ dataset.

The results are shown in Table 8. As can be seen from the results, the combination of the enhancement module and the expert network can achieve the best detection performance, with an AUC of 99.15(%) and an ACC of 96.87(%). This configuration fully utilizes the adaptive optimization capability of the enhancement module for the feature space and the accurate capture capability of the expert network for different scales and feature patterns. The configuration using only the enhancement module can improve the feature expression capability of the model, but due to the lack of diversity support of the heterogeneous expert network, the performance is reduced, with the AUC and ACC reduced

Table 6. Analyze the impact of the total number of experts on performance and computational cost. Evaluation metric are AUC(%), ACC(%), inference time(ms) and FLOPs(GFLOPs).

Number of Experts	AUC (%)	ACC (%)	Inference Time(ms)	FLOPs(GFLOPs)
2	79.85	77.51	15.06	35.10
3	81.64	78.28	17.15	60.21
4	82.15	78.72	20.41	91.02
5	83.25	80.02	25.06	129.57
6	83.51	80.49	31.84	172.02

Table 7. Three fusion methods were used for comparison, including weighted feature fusion, maximum fusion and average fusion, and trained and tested on the FF++ dataset. Evaluation metric are AUC(%), ACC(%)

Fusion method	AUC (%)	ACC (%)
Weighted feature fusion	99.15	96.87
Maximum Fusion	99.02	96.24
Average Fusion	98.57	95.75

to 96.21(%) and 92.65(%) respectively. The configuration using only the expert network performs slightly better than the case of only the enhancement module, with an AUC of 97.54(%) and an ACC of 94.56(%). This shows that the heterogeneity and multi-scale feature extraction capabilities of the expert network play an important role in improving the performance of the model.

4.3.7 Visualizations of the captured artifacts.

We further use GradCAM to locate which regions are activated when detecting forgeries. The visualization results shown in Figure 3 show that when using different expert networks to locate forged regions for the same forged image, experts 3 and 4 can accurately locate the forged regions, while other experts capture limited forged regions. Therefore, the gating network can help select experts 3 and 4 well, thereby capturing accurate forged regions to distinguish between true and false. This visualization further shows that our gating network estimates that the expert network captures more general forgery features.

5. Conclusion

In this paper, we introduced DFMoE, a framework for deepfake detection that integrates a Heterogeneous Mixture of Experts model and adaptive feature enhancement strategies to boost generalizability and robustness. DFMoE dynamically selects experts based on input features, enabling effective handling of diverse deepfake attacks. Compared to existing methods, our approach excels in detecting both known and unknown forgery types. The model leverages multi-scale feature extraction and applies in-domain and cross-domain augmentations to improve robustness. Our optimization strategy balances expert selection through parameter penalty and entropy loss, preventing over-reliance on large experts. Experimental results show DFMoE's superior performance across multiple datasets, particularly in detecting unknown forgery samples. Future work will focus on scaling the model, optimizing expert networks, and developing new augmentation techniques to keep pace with evolving forgery methods.

Acknowledgement

This work is supported by the Natural Science Foundation Innovation and Development Joint Fund Project of Shandong Province under Grant NO. ZR2023LZH009.

References

[1] Deepfakes. www.github.com/deepfakes/faceswap, 2020. Accessed 2020-09-02. 6

[2] Faceswap. www.github.com/MarekKowalski/FaceSwap, 2021. Accessed 2020-09-03. 6

[3] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE international workshop on information forensics and security (WIFS)*, pages 1–7. IEEE, 2018. 8

[4] M. Artetxe, S. Bhosale, N. Goyal, T. Mihaylov, M. Ott, S. Shleifer, X. L. Lin, J. Du, S. Iyer, R. Pasunuru, et al. Efficient large scale language modeling with mixtures of experts. *arXiv preprint arXiv:2112.10684*, 2021. 3

[5] J. Cao, C. Ma, T. Yao, S. Chen, S. Ding, and X. Yang. End-to-end reconstruction-classification learning for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4113–4122, 2022. 2, 8

[6] S. Chen, T. Yao, Y. Chen, S. Ding, J. Li, and R. Ji. Local relation learning for face forgery detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 1081–1088, 2021. 3

[7] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on*

Table 8. Three sets of comparative experiments, including "enhancement module + expert network", "enhancement module only" and "expert network only", were trained and tested on the FF++ dataset. Evaluation metric are AUC(%), ACC(%)

Module composition	AUC (%)	ACC (%)
Enhancement Module	96.21	92.65
Expert Network Module	97.54	94.56
Enhancement + Expert Network	99.15	96.87

Fake Expert1(B1) Expert2(B2) Expert3(B3) Expert4(B4) Expert5(B5)

Figure 3. Different expert networks capture different forgery features. Here we use networks B1 to B5 as five different expert networks.

computer vision and pattern recognition, pages 1251–1258, 2017. 2, 8

[8] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, pages 5781–5790, 2020. 8

[9] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*, 2020. 6

[10] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019. 6

[11] S. Dong, J. Wang, J. Liang, H. Fan, and R. Ji. Explaining deepfake detection by analysing image matching. In *European Conference on Computer Vision*, pages 18–35. Springer, 2022. 3

[12] N. Dufour and A. Gully. Contributing data to deepfake detection research. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html, 2019. Accessed: 2019-09-24. 6

[13] W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. 3

[14] Q. Gu, S. Chen, T. Yao, Y. Chen, S. Ding, and R. Yi. Exploiting fine-grained face forgery clues via progressive enhancement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 735–743, 2022. 3

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[16] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991. 3

[17] D. E. King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758, 2009. 7

[18] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. 3

[19] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen. Advancing high fidelity identity swapping for forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5074–5083, 2020. 6

[20] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5001–5010, 2020. 2, 8

[21] Y. Li, M.-C. Chang, and S. Lyu. In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018. 6

[22] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu. Celeb-df: A new dataset for deepfake forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2020. 6

[23] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 772–781, 2021. 3

[24] Y. Lou, F. Xue, Z. Zheng, and Y. You. Cross-token modeling with conditional computation. *arXiv preprint arXiv:2109.02008*, 2021. 3

[25] B. Mustafa, C. Riquelme, J. Puigcerver, R. Jenatton, and N. Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. In *Advances in Neural Information Processing Systems*, volume 35, pages 9564–9576, 2022. 3

[26] H. H. Nguyen, J. Yamagishi, and I. Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2307–2311. IEEE, 2019. 8

[27] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on computer vision*, pages 86–103. Springer, 2020. 3

[28] C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. Susano Pinto, D. Keysers, and N. Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021. 3

[29] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019. 6

[30] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 3

[31] S. Shen, Z. Yao, C. Li, T. Darrell, K. Keutzer, and Y. He. Scaling vision-language models with sparse mixture of experts. *arXiv preprint arXiv:2303.07226*, 2023. 3

[32] K. Shiohara and T. Yamasaki. Detecting deepfakes with self-blended images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18720–18729, 2022. 2

[33] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 2, 8

[34] J. Thies, M. Zollhöfer, and M. Nießner. Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 6

[35] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016. 6

[36] A. Wang, X. Sun, R. Xie, S. Li, J. Zhu, Z. Yang, P. Zhao, J. Han, Z. Kang, D. Wang, et al. Hmoe: Heterogeneous mixture of experts for language modeling. *arXiv preprint arXiv:2408.10681*, 2024. 2, 3

[37] C. Wang and W. Deng. Representative forgery mining for fake face detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14923–14932, 2021. 8

[38] J. Wang, Z. Wu, W. Ouyang, X. Han, J. Chen, Y.-G. Jiang, and S.-N. Li. M2tr: Multi-modal multi-scale transformers for deepfake detection. In *Proceedings of the 2022 international conference on multimedia retrieval*, pages 615–623, 2022. 3

[39] Z. Yan, Y. Luo, S. Lyu, Q. Liu, and B. Wu. Transcending forgery specificity with latent space augmentation for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8984–8994, 2024. 3, 8

[40] Z. Yan, Y. Zhang, Y. Fan, and B. Wu. Ucf: Uncovering common features for generalizable deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22412–22423, 2023. 8

[41] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2185–2194, 2021. 2