DFCP: Few-Shot DeepFake Detection via Contrastive Pretraining

Bo Zou[1], Chao Yang[3], Jiazhi Guan[1], Chengbin Quan[1], Youjian Zhao[1,2]
[1] Department of computer science and Technology, Tsinghua University, Beijing, China
[2] Zhongguancun Laboratory, Beijing, China [3] Shanghai AI Laboratory, Shanghai, China

***Abstract*—Abuses of forgery techniques have created a considerable problem of misinformation on social media. Although scholars devote many efforts to face forgery detection (a.k.a DeepFake detection) and achieve some results, two issues still hinder the practical application. 1) Most detectors do not generalize well to unseen datasets. 2) In a supervised manner, most previous works require a considerable amount of manually labeled data. To address these problems, we propose a simple contrastive pertaining framework for DeepFake detection (DFCP), which works in a finetuning-after-pretraining manner, and requires only a few labels (5%). Specifically, we design a two-stream framework to simultaneously learn high-frequency texture features and high-level semantics information during pretraining. In addition, a video-based frame sampling strategy is proposed to mitigate potential noise data in the instance-discriminative contrastive learning to achieve better performance. Experimental results on several downstream datasets show the state-of-the-art performance of the proposed DFCP, which works at frame-level (w/o temporal reasoning) with high efficiency but outperforms video-level methods.**

***Index Terms*—Face Forgery Detection, DeepFake, Self-supervised Learning, Contrastive Learning**

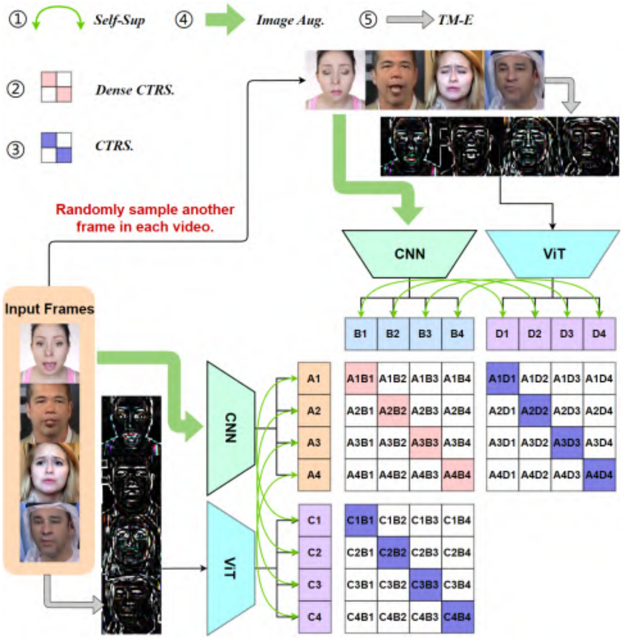

Fig. 1. Overview of DFCP. We adopt three pretraining tasks: ① We maximize the similarity between RGB frames and their texture map. ② We apply dense-contrastive learning that pulls representations from the same video closer by leveraging the contrast between local areas in frames. ③ We perform contrastive learning between frames and texture maps from randomly chosen frames. ④ indicates applying commonly used image augmentations. ⑤ represents high-frequency texture map extraction.

I. Introduction

The face forgery technology, also known as DeepFake, utilizes prime mature methods like Face2Face [1] to generate fake videos by synthesizing the identities, movements, or gestures of a target person into the original video. This technology has been widely spread since developers uploaded user-friendly tools. However, abuse of DeepFake could cause violation of privacy and portrait rights, even sensitive political issues. Therefore, DeepFake detection has become a research hot-spot in recent years. Typically, DeepFake detection can be divided into frame-level [2]–[6] and video-level [7]–[11] methods. The former makes decisions based on single frames, while the latter considers consecutive frame sequences. Generally speaking, video-level methods perform better since they introduce temporal artifacts as the judgment basis. Nevertheless, they bring extra computational overhead.

Most current DeepFake detection models are trained in a supervised manner. They require a large amount of labeled data, while the labeling process is labor-intensive and error-prone. Besides, they fail to give ideal results on unseen types of DeepFake due to inconsistent forgery clues. Retraining models on new datasets is inevitable for practical deployment despite being time-consuming. The prevalent pretraining-finetuning paradigm seems promising to overcome the shortcomings above, but only a few related attempts [12] have been made in the field of DeepFake detection, and there is a noticeable performance gap compared with supervised methods.

In this work, we empirically reveal two major reasons for the failure and propose a novel contrastive pertaining framework for frame-level detection called DFCP. Specifically, we introduce a simple but crucial frame sampling strategy to avoid putting potential positive sample pairs into negative pairs. Inspired by [13], we design dense-contrastive learning for DeepFake detection to overcome the defect of regular contrastive learning that only focuses on global features. In addition, previous work [2] shows high-frequency texture features play a decisive role in DeepFake detection. However, utilizing texture features always needs special encoder designs. We develop a two-stream pretraining framework that enables contrast between RGB frames and texture maps to improve the understanding of high-frequency features without changing commonly used image encoders. To demonstrate the effectiveness of DFCP, we validate our method on FaceForensics++

[1], Celeb-DF [14], and DFDC [15]. The results show our framework outperforms most supervised methods and achieves state-of-the-art performance in cross-dataset evaluations. Our contributions are summarized as follows:

- We propose a contrastive pretraining framework DFCP that simultaneously learns high-frequency texture features and high-level semantics without changing commonly used image encoders.
- We empirically reveal two major reasons for the failure of previous works and specially design a video-based frame sampling strategy and dense-contrastive learning for DeepFake detection.
- Extensive experiments demonstrate that our DFCP outperforms supervised methods, especially in cross-dataset evaluations.

II. Related Work

Supervised DeepFake Detection. Most recent works are supervised. At frame-level, [2] define DeepFake detection as a fine-grained classification problem. They add a Multi-attention module and texture enhancement branch to improve the performance. [3] propose a dynamic augmentation method, which makes the network learn fake clues from limited local characteristics. At video-level, [7] use LSTM to integrate spatial domain knowledge into the temporal domain. [8] come up with Spatio-Temporal modeling by 3DCNN, making the spatial and temporal features complementary in training. Some supervised methods take advantage of the ideal of clustering in contrastive learning to improve model generalization. They are classified as supervised contrastive learning. [5] propose a two-branch framework to realize an intra-instance contrast and an inter-instance contrast. [16] leverage the similarity between videos under different compression qualities. Nevertheless, these methods are still trained in a supervised manner. They can not get rid of large-scale manually annotated datasets.

Self-supervised Contrastive Learning. Contrastive Learning is a typical pretraining-finetuning method. The pretraining aims to maximize the similarity between images with their augmentations (positive pairs) meanwhile against other combinations (negative pairs). [17] presented a simple but powerful framework named SimCLR. It was trained end-to-end, and every model part was differentiable. [18] came up with momentum update mechanics and a dictionary to save samples' representations during training. This framework benefited from the consistency of representations saved in the dictionary. To our knowledge, only a few DeepFake detection works are self-supervised using contrastive learning. [12] adopt settings from simCLR and preliminary confirm that self-supervised methods can detect DeepFake. [19] pretrain a model pulling representations of the audio stream and the image stream in videos closer, then finetune the model for classification. However, these attempts show the distinctiveness of DeepFake detection compared to general image classification tasks. Simply adopting contrastive learning in DeepFake detection will result in a noticeable performance gap compared with supervised methods.

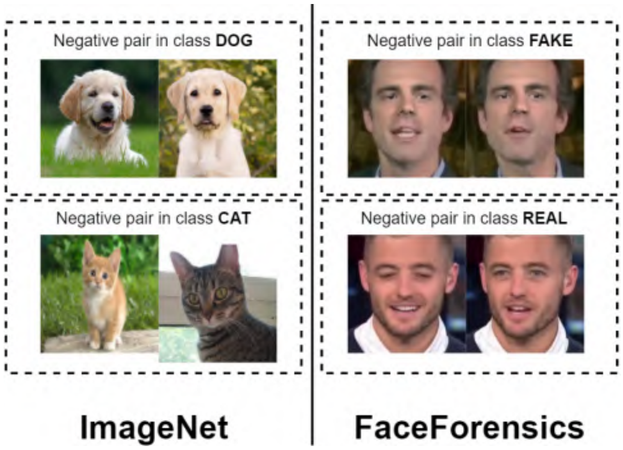
Negative pair in class DOG
Negative pair in class FAKE
Negative pair in class CAT
Negative pair in class REAL
ImageNet
FaceForensics

Fig. 2. For most potentially positive pairs from ImageNet(left), differences are easy to find. By contrast, in DeepFake datasets(right), potentially positive pairs could only have facial expressions or head position differences.

III. Methodology

This section describes technical approaches for DFCP, an efficient pretraining framework for frame-level DeepFake detection. We aim to realize high-performance detection by designing pretraining tasks without changing commonly used image encoders. We first demonstrate the two possible reasons for the performance gap in previous self-supervised works, then specify each part of DFCP.

A. Limitations of Previous Work

Potentially Positive Pairs. Typically, contrastive learning for general image classification treats an image with its augmentation as a positive pair and other combinations as negative pairs. The pretraining task requires the network to maximize the positive pair's similarity while minimizing the negative pair's similarity. This task is "instance-discriminative" because it treats every image as a distinct class. The instance-discriminative task possibly leads pairs consisting of two images from the same class in downstream tasks to be negative, and we define these pairs as "potentially positive" pairs. For general image classification datasets like ImageNet [20], potentially positive pairs will not hinder the downstream performance since the dissimilarity between the images can be found in the background or other noticeable differences like illumination, as shown on the left of Fig. 2. By contrast, in DeepFake detection, many potentially positive pairs from the same video contain frames with only facial expressions or head position differences. They share the same background, light condition, and identity. Since representations of two similar frames in a potentially positive pair are pushed apart during the pretraining, the encoder trend to find subtle differences from image dithering, blur, or local texture. However, these details are essential clues for discriminating between real and fake. Previous works do not deal with potentially positive pairs. Thus their performances are limited.

Absence of Local information. Standard contrastive learning employs global feature vectors generated by pooling feature maps from the image encoder. While many clues of DeepFake, distinguished from general classification, are not global attributes. Former attempts only pretrained on global vectors, thus hindering learning the local features (e.g., local area inconsistency, unnatural blur, and noise pattern) vital to DeepFake detection.

B. Video-based Frame Sampling Strategy

The solution of mitigating potentially positive pairs is straightforward. We propose a video-based frame sampling strategy that samples b videos from the dataset first, then randomly chooses two frames A and B from each video. Now, we define $2b$ positive pairs according to b videos ($\{A, B\}$ and $\{B, A\}$ are considered as different pairs), and $4b^2 - 4b$ negative pairs (other combination of $2b$ frames). In this way, we avoid harmful potentially positive pairs that contain two frames from the same video. Besides, the random time span between A and B increases the diversity of positive pairs. It can be considered as hard data mining that improves the quality of pretraining.

C. Dense-contrastive Learning in RGB Stream

This subsection realizes ② in Fig: 1, which consists of two parts. The first part is a global contrast. We first randomly apply one commonly used augmentation method (horizontal flip, vertical flip, rotation, random grayscale, color jittering, or random noising) for each frame (④ in Fig: 1), then divide each into 4×4 patches and randomly mask 0% to 62.5% patches by black. Denote $\overline{X_{\mathrm{i}}} \in \mathbb{R}^{C \times H \times W}$ as an augmented view of frame X_{i}. A CNN backbone f_{CNN} is used to generate the feature map $m_{\mathrm{i}} = f_{\mathrm{CNN}}\left(\overline{X_{\mathrm{i}}}\right) \in \mathbb{R}^{C' \times H' \times W'}$. m_{i} is pooled by global average pooling, then projected by a MLP projection head to generate the global feature vector $v_{\mathrm{i}} \in \mathbb{R}^{C'}$. We compute a global contrastive loss formulated as:

$$L_{\mathrm{global}} = -\sum_{\mathrm{i} \in \mathrm{B}} \log \frac{\exp\left(\operatorname{sim}\left(v_{\mathrm{i}}, v_{\mathrm{i}+}\right)/\tau\right)}{\sum_{\mathrm{j} \neq \mathrm{i}^+} \exp\left(\operatorname{sim}\left(v_{\mathrm{i}}, v_{\mathrm{j}}\right)/\tau\right)}. \tag{1}$$

where $\mathrm{B} = \{1, \ldots, 2b\}$, v_{i} and $v_{\mathrm{i}+}$ denote vectors of a positive pair defined in subsection III-B, τ is the temperature parameter equals 0.07. sim represents the cosine similarity function:

$$\operatorname{sim}\left(v_{\mathrm{i}}, v_{\mathrm{j}}\right) = \frac{v_{\mathrm{i}} \cdot v_{\mathrm{j}}}{\|v_{\mathrm{i}}\| \times \|v_{\mathrm{j}}\|}, \tag{2}$$

The second part is a dense contrast between local areas to mitigate the absence of local features in standard contrastive learning. Inspired by [13], we discard the global average pooling and rearrange the feature map m_{i} into a sequence of local feature vectors $v_{\mathrm{i}}^{\mathrm{k}} \in \mathbb{R}^{C'}$, where $\mathrm{k} \in \mathrm{N}$ and $\mathrm{N} = \{1, \ldots, H' \times W'\}$. Unlike [13], we normalize $v_{\mathrm{i}}^{\mathrm{k}}$ by v_{i} through element-wise division. This operation eliminates the influence of global attributes like illumination and makes our dense contrast more focused on local features. Then we pass local feature vectors through another projection head, and the outcomes are denoted as $u_{\mathrm{i}}^{\mathrm{k}}$. The positive vector of $u_{\mathrm{i}}^{\mathrm{k}}$ is defined as its most similar vector in $m_{\mathrm{i}+}$, and negative vectors are defined as global feature vectors v_{j} for all $\mathrm{j} \neq \mathrm{i}^+$. This definition allows the encoder to find similarities in local areas while not learning dissimilarities from local details. The dense contrastive loss is formulated as:

$$L_{\mathrm{local}} = -\sum_{\mathrm{i} \in \mathrm{B}} \sum_{\mathrm{k} \in \mathrm{N}} \log \frac{\exp\left(\operatorname{sim}\left(u_{\mathrm{i}}^{\mathrm{k}}, u_{\mathrm{i}+}^{\mathrm{k}^+}\right)/\tau\right)}{\sum_{\mathrm{j} \neq \mathrm{i}^+} \exp\left(\operatorname{sim}\left(u_{\mathrm{i}}^{\mathrm{k}}, v_{\mathrm{j}}\right)/\tau\right)}. \tag{3}$$

where $\mathrm{k}^+ = \operatorname*{argmax}_{\mathrm{l} \in \mathcal{N}} \operatorname{sim}\left(u_{\mathrm{i}}^{\mathrm{k}}, u_{\mathrm{i}+}^{\mathrm{l}}\right)$. Finally, the target of dense-contrastive learning in RGB stream is to minimize $L_{\mathrm{dense}} = L_{\mathrm{global}} + L_{\mathrm{local}}$.

D. Contrastive Learning in Texture Stream

As aforementioned, high-frequency texture features are just as important as high-level semantics in DeepFake detection. To leverage texture features, we first define the texture map (⑤ in Fig: 1) $X_{\mathrm{i}}^{\mathrm{T}} = X_{\mathrm{i}} - X_{\mathrm{i}}^{-}$. X_{i}^{-} denotes a recovered down-sampled view of X_{i} under the control of the down-sampled rate r, which means resizing X_{i} to $\mathbb{R}^{C \times rH \times rW}$ and then up-scaling to the original shape. We use a weak ViT encoder f_{ViT} (with a few blocks and attention heads) to generate feature map $m_{\mathrm{i}}^{\mathrm{T}} = f_{\mathrm{ViT}}\left(X_{\mathrm{i}}^{\mathrm{T}}\right) \in \mathbb{R}^{C' \times H' \times W'}$, since the high-level understanding of low-level features is unnecessary. Same as the global contrast in subsection III-C, $m_{\mathrm{i}}^{\mathrm{T}}$ is pooled by global average pooling, then projected by a MLP projection head to generate the global texture feature vector $v_{\mathrm{i}}^{\mathrm{T}} \in \mathbb{R}^{C'}$. For ③ in Fig: 1, we computes $L_{\mathrm{global}}^{\mathrm{T}}$ by replacing v_{i} in equation (1) with $v_{\mathrm{i}}^{\mathrm{T}}$. It ensures the CNN encoder of RGB frames concerns useful texture information. Besides, to further improve the quality of texture encoding, we introduce an instance-discriminative task ① in Fig: 1, requiring texture map $X_{\mathrm{i}}^{\mathrm{T}}$ to retrieve the original RGB view X_{i} among all frames. Its self-supervised loss $L_{\mathrm{self}}^{\mathrm{T}}$ is also calculated by equation (1) with replacing v_{i} to $v_{\mathrm{i}}^{\mathrm{T}}$, but i^+ is defined as $\mathrm{i}^+ = \mathrm{i}$. The target of contrastive learning in texture stream is to minimize $L_{\mathrm{texture}} = L_{\mathrm{global}}^{\mathrm{T}} + L_{\mathrm{self}}^{\mathrm{T}}$.

The intuition of using ViT rather than CNN in texture stream is: Texture map extraction drastically changes signal intensity in local areas of X_{i}. $X_{\mathrm{i}}^{\mathrm{T}}$ is sparse and almost black in most areas since it is generated by subtraction. Consequently, lines and edges in X_{i} dominate the signal intensity of $X_{\mathrm{i}}^{\mathrm{T}}$. However, we do not want clear and strong edges to disturb the learning of wake blur and dithering brought by image forgery. CNN generates image representations by convolution. It is sensitive to local signal intensity, while the attention mechanism adopted in ViT can reduce the impact of signal intensity in local areas. Another merit of applying ViT is the ability to capture long-distance dependencies in $X_{\mathrm{i}}^{\mathrm{T}}$ with few blocks, then reduce the computational overhead of a two-stream framework setting. Nevertheless, we still use CNN in the RGB stream because of the limited scale of current DeepFake datasets. Typically, ViT needs more data to release the potential. Overall, the total loss of DFCP is formulated as:

$$\mathit{Loss} = \lambda \times L_{\mathrm{dense}} + (1 - \lambda) \times L_{\mathrm{texture}} \tag{4}$$

where λ acts as the weight to balance the importance of two terms. We set λ to 0.5 for all experiments in this paper.

IV. Experiment

A. Datasets

We evaluate our DFCP on three commonly used datasets. **FF++** [1] includes 1000 real videos and 4000 manipulated videos generated by four forgery methods. In this paper, all performances are based on high-quality videos with quantization parameters c23. **Celeb-DF (V2)** [14] is a widely used dataset that contains 590 original videos collected from YouTube with subjects of different ages, ethnic groups, genders, and 5639 DeepFake videos. Celeb-DF is challenging since most fake videos are designed to evade noticeable artifacts. **DFDC** [15] contains 1131 real videos and 4113 fake videos generated by several manipulated methods. The light conditions and the proportion of faces in the picture change significantly among videos.

B. Implement Details

Preprocessing. We uniformly sample 50 frames from the first 288 frames in each video, then crop them to align human faces using bounding boxes detected by MTCNN [21]. The split of training and testing videos follows the official guidelines of each dataset.

Pretrianing setup. We resize all frames to 256×256. The down-sampling rate r of texture map extraction is 0.1. We adopt Xception [22] pretrained on ImageNet as the CNN backbone. The ViT encoder in the texture stream has 6 layers, each with a hidden dimension of 384. The number of attention heads is set to 8, and the inputs are divided into 64 patches. We train the ViT encoder from scratch. All projection heads have two FC layers, and the output dimension is set to 256. Our batch size is 256, and the learning rate is $3e-3$. We adopt Adam optimizer with a weight decay of $5e-4$ and pretrain encoders on FF++ for 60 epochs.

Evaluation protocol. We evaluate DFCP by finetuning a classification model consisting of the pretrained CNN encoder in the RGB stream and a single-layer classification head. The ViT encoder in the texture stream is unnecessary for classifications since it is designed to help the RGB stream learn high-frequency textures. The classification model is finetuned end-to-end on a testing set with labels. To compare with previous methods, We perform both full-data finetuning and few-shot finetuning. The few-shot finetuning relies on a randomly extracted subset (5%) of the testing set.

C. Quantitative Results

Comparison with supervised methods at frame-level. To demonstrate the effectiveness of DFCP, we conducted both in-dataset and cross-dataset evaluations and compared results with predominant supervised methods at frame-level. In-dataset evaluation means training and testing on the same dataset. In table I, all previous works are trained on the entire FF++ in a supervised manner. Besides the last two rows, all cross-dataset results are zero-shot. We use a classification model (defined in subsection IV-B) without pretraining as the baseline. From comparisons with these methods, we list the following observations: (1) DFCP outperforms cutting-edge supervised methods in zero-shot cross-dataset evaluations with 5.15% and 2.58% performance gains. It shows the generalization of DFCP. (2) Full-data finetuning achieves the runner-up of in-dataset evaluation but damages the generalization, possibly due to over-fitting. In addition, we also report our few-shot performance on Celeb-DF and DFDC.

TABLE I
In-dataset evaluations on FF++ and cross-dataset evaluations on Celeb-DF&DFDC for **Frame-Level** methods.

Method	AUC(%)		
	FF++	Celeb-DF	DFDC
EN-b4 [5]	99.22	68.52	70.12
Face X-ray [4]	87.40	74.20	70.00
F3-Net [23]	98.10	71.21	72.88
MAT(EN-b4) [2]	99.27	76.65	67.34
GFF [24]	98.36	75.31	71.58
LTW [25]	99.17	77.14	74.58
Local-relation [6]	**99.46**	78.26	76.53
Capsule [26]	96.60	57.50	-
Two Branch [27]	93.18	73.41	-
DCL [5]	99.30	82.30	76.70
baseline (100% FF++)	98.91	67.14	64.98
baseline (5% FF++)	59.20	55.75	56.51
Ours (100% FF++)	99.38	84.35	75.02
Ours (5% FF++)	94.95	**87.45**	**79.28**
Ours (5% DFDC)	-	-	81.61
Ours (5% Celeb-DF)	-	89.15	-

TABLE II
In-dataset results from **Video-Level** works and finetuning results of DFCP. †denotes the video-level method.

Method	ACC(%)		
	Celeb-DF	DFDC	Avg
D-FWA† [28]	98.58	85.11	91.85
I3D† [29]	99.23	80.82	90.03
S-MIL-T† [9]	98.84	**85.11**	91.98
LPS (c23)† [30]	92.55	80.25	86.40
baseline(100%)	95.95	80.30	88.13
Ours(5%)	85.10	76.92	81.01
Ours(100%)	**100.0**	84.03	**92.02**

Comparison with supervised methods at video-level. To further present the generalization of DFCP, we adopt evaluation metrics from [9] and compare DFCP with video-level methods that are full-data trained on Celeb-DF and DFDC. As shown in table II, DFCP is competitive with video-level methods and achieves impressive 100% accuracy on Celeb-DF. As for videos in DFDC, some forgery clues hide in camera movements, changes in picture proportion of people, and the variation of illumination. The lack of temporal reasoning makes it hard for DFCP to overcome these difficulties.

Comparison with self-supervised methods. We also list comparisons between our DFCP with UCL [12] and Zhao et al. [19], the only two published DeepFake detection works that are fully self-supervised to the best of our knowledge. UCL has a standard contrastive learning framework without much special design for DeepFake detection and can be regarded as a baseline for all self-supervised methods. Zhao et al. [19] is a multi-modal method that simultaneously concerns video and audio. They utilize the inconsistency between video

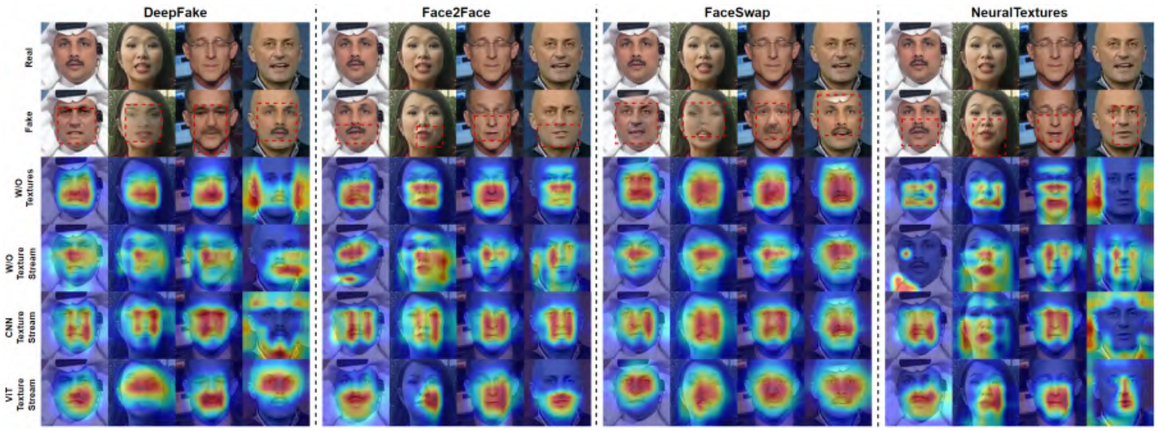

Fig. 3. Class activation maps (CAM) for variants of DFCP.

and audio as the judgment basis. From the better cross-dataset performance of DFCP, we guess high-frequency texture features are more general DeepFake clues compared with inconsistencies between video and audio.

D. Visualization results

We provide class activation maps (CAM) to demonstrate the benefits of the two-stream setting and the encoder type setting in DFCP. Fig. 3 consists of CAMs of: (1) DFCP without utilizing texture maps (baseline). (2) DFCP leveraging texture maps without the texture stream. (3) DFCP with CNN encoder in texture stream. (4) DFCP with ViT encoder in texture stream. We validate these models on 16 fake faces generated by four forgery methods. Red dotted boxes mark the manipulated region in each fake face. Generally speaking, vital clues of forgery often appear along the edge of manipulated regions or in heavily changed parts inside the manipulated regions. Comparison between (1) and (2) demonstrates that putting texture features into the RGB stream will make the encoder emphasize high-frequency features too much. Separating the encoding of texture maps into the texture stream mitigate this defect, but CNN is incompetent to improve the performance further. (3) shows the model with a CNN texture stream fails to combine low-frequency semantics and high-frequency texture features. By contrast, (4) can pinpoint the manipulated regions and high-frequency edges around them.

E. Ablation Study

Effectiveness of components. All results reported in this subsection from models trained and finetuned on full-scale FF++. To further demonstrate the effectiveness of each component, we specially design the following variant models. 1) DFCP that doesn't focus on texture features. 2) DFCP that treats texture maps as normal augmentations in RGB stream. 3) DFCP that only has global contrastive learning. 4) DFCP that does not deal with potentially positive pairs. 5) Full DFCP with all specially designed components.

Comparisons between variation 1, 2, and 5 show the improvement brought by using texture features. In table III, the comparison between variation 2 and variation 1 shows using texture features can improve in-dataset performance by 1.61%, but it damages cross-dataset performance. This demonstrates that directly adding texture maps into the RGB stream harms model generalization. However, we observe over 6% improvement in cross-dataset evaluation after applying texture stream to learn texture features. The 0.26% drop in in-dataset performance is nothing compared to such a huge generalization improvement.

TABLE III
PERFORMANCE OF SELF-SUPERVISED METHODS. ‡DENOTES THE MULTI-MODAL METHOD.

Method	AUC(%)		
	FF++	Celeb-DF	DFDC
UCL (100% FF++) [12]	93.00	58.90	-
Zhao et al. (100% FF++)‡ [19]	**99.60**	84.20	74.50
Ours (100% FF++)	99.38	84.35	75.02
Ours (5% FF++)	94.95	**87.45**	**79.28**

Comparing variation 5 and variation 3 shows dense contrastive learning brings 3.01% and 1.94% gains in in-dataset and cross-dataset evaluation. In variation 4, we don't apply the video-based sampling strategy. That means $i^{+} = i$ in equation 1. Significant performance drops in variation 4 show the importance of handling potentially positive pairs.

Important hyper-parameters. Table V shows the results of different masking ratios in the RGB stream and reports performances under different down-sampling rates in texture map extraction.

V. Conclusion

In this paper, We reveal two major reasons for the failure of previous pretraining-finetuning DeepFake detection works and propose a high-performance pretraining framework DFCP, which has dedicated pretraining tasks designed for DeepFake detection and does not change commonly used image encoder. DFCP spontaneously learns high-frequency texture features and high-level semantics information and can quickly adapt to new domains by few-shot finetuning. Extensive experiments demonstrate that our DFCP outperforms supervised methods, especially in cross-dataset evaluations. One limitation is that DFCP does not take temporal inconsistency into account.

TABLE IV
ABLATION STUDIES ON THE EFFECTIVENESS OF DIFFERENT COMPONENTS.

variant	Common Aug	Texture map	Texture Stream	Dense Contrast	Video-based Sampling	FF++	Celeb-DF
1	✓			✓	✓	98.03	79.04
2	✓	✓		✓	✓	**99.64**	78.25
3	✓	✓	✓		✓	96.37	82.41
4	✓	✓	✓	✓		94.50	77.23
5	✓	✓	✓	✓	✓	99.38	**84.35**

TABLE V
ABLATION STUDIES ON THE RATIO OF RANDOM MASKING AND THE DOWN-SAMPLING RATE

masking ratio	0%	37.5%	62.5%	87.5%
FF++	96.90	98.10	**99.38**	94.51
down-sampling rate r	0.05	0.1	0.3	0.5
FF++	99.00	**99.38**	98.71	98.20

Although learning from temporal features will bring extra computational overhead, they are still vital clues for forgery detection. Fortunately, experiments in table II suggest that the purely frame-level method DFCP is competent for detecting most current DeepFake videos.

REFERENCES

[1] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner, "Faceforensics++: Learning to detect manipulated facial images," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1–11.

[2] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu, "Multi-attentional deepfake detection," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 2185–2194.

[3] Sowmen Das, Selim Seferbekov, Arup Datta, Md Islam, Md Amin, et al., "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, 2021, pp. 3776–3785.

[4] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo, "Face x-ray for more general face forgery detection. in 2020 ieee," in CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[5] Ke Sun, Taiping Yao, and Rongrong Ji Shen Chen, Jilin L, "Dual contrastive learning for general face forgery detection," arXiv, 2021.

[6] Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, Jilin Li, and Rongrong Ji, "Local relation learning for face forgery detection," arXiv preprint arXiv:2105.02577, 2021.

[7] I. Masi, A. Killekar, R. M. Mascarenhas, S. P. Gurudatt, and W. Abdalmageed, Two-Branch Recurrent Network for Isolating Deepfakes in Videos, Computer Vision – ECCV 2020, 2020.

[8] Ipek Ganiyusufoglu, L Minh Ngô, Nedko Savov, Sezer Karaoglu, and Theo Gevers, "Spatio-temporal features for generalized detection of deepfake videos," arXiv preprint arXiv:2010.11844, 2020.

[9] Xiaodan Li, Yining Lang, Yuefeng Chen, Xiaofeng Mao, Yuan He, Shuhui Wang, Hui Xue, and Quan Lu, "Sharp multiple instance learning for deepfake video detection," in Proceedings of the 28th ACM international conference on multimedia, 2020, pp. 1864–1872.

[10] Jiazhi Guan, Hang Zhou, Zhibin Hong, Errui Ding, Jingdong Wang, Chengbin Quan, and Youjian Zhao, "Delving into sequential patches for deepfake detection," in Advances in Neural Information Processing Systems, 2022, vol. 35, pp. 4517–4530.

[11] Jiazhi Guan, Hang Zhou, Mingming Gong, Youjian Zhao, Errui Ding, and Jingdong Wang, "Detecting deepfake by creating spatio-temporal regularity disruption," arXiv preprint arXiv:2207.10402, 2022.

[12] Sheldon Fung, Xuequan Lu, Chao Zhang, and Chang-Tsun Li, "Deepfakeucl: Deepfake detection via unsupervised contrastive learning," arXiv preprint arXiv:2104.11507, 2021.

[13] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li, "Dense contrastive learning for self-supervised visual pre-training," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3024–3033.

[14] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," arXiv, 2019.

[15] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer, "The deepfake detection challenge dataset," arXiv e-prints, pp. arXiv–2006, 2020.

[16] Jian Zhang, Jiangqun Ni, and Hao Xie, "Deepfake videos detection using self-supervised decoupling network," in 2021 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2021, pp. 1–6.

[17] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in International conference on machine learning. PMLR, 2020.

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[19] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Weiming Zhang, and Nenghai Yu, "Self-supervised transformer for deepfake detection," arXiv preprint arXiv:2203.01265, 2022.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet:a large-scale hierarchical image database," in IEEE conference on computer vision and pattern recognition, 2009.

[21] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, 2016.

[22] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[23] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao, "Thinking in frequency: Face forgery detection by mining frequency-aware clues," pp. 86–103, 2020.

[24] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu, "Generalizing face forgery detection with high-frequency features," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 16317–16326.

[25] Ke Sun, Hong Liu, Qixiang Ye, Jianzhuang Liu, Yue Gao, Ling Shao, and Rongrong Ji, "Domain general face forgery detection by learning to weight," in Proceedings of the AAAI Conference on Artificial Intelligence, 2021, vol. 35, pp. 2638–2646.

[26] Huy H Nguyen, Junichi Yamagishi, and Isao 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). IEEE, 2019, pp. 2307–2311.

[27] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed, "Two-branch recurrent network for isolating deepfakes in videos," in European Conference on Computer Vision. Springer, 2020, pp. 667–684.

[28] Yuezun Li and Siwei Lyu, "Exposing deepfake videos by detecting face warping artifacts," arXiv preprint arXiv:1811.00656, 2018.

[29] Joao Carreira and Andrew Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[30] Shiming Ge, Fanzhao Lin, Chenyu Li, Daichi Zhang, Weiping Wang, and Dan Zeng, "Deepfake video detection via predictive representation learning," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2022.