Deepfake Video Detection via Predictive Representation Learning

SHIMING GE, FANZHAO LIN, CHENYU LI, DAICHI ZHANG, and WEIPING WANG,
Institute of Information Engineering, Chinese Academy of Sciences, and School of Cyber Security, University of Chinese Academy of Sciences, China
DAN ZENG, School of Communication and Information Engineering, Shanghai University, China

Increasingly advanced deepfake approaches have made the detection of deepfake videos very challenging. We observe that the general deepfake videos often exhibit appearance-level temporal inconsistencies in some facial components between frames, resulting in discriminative spatiotemporal latent patterns among semantic-level feature maps. Inspired by this finding, we propose a predictive representative learning approach termed Latent Pattern Sensing to capture these semantic change characteristics for deepfake video detection. The approach cascades a Convolution Neural Network-based encoder, a ConvGRU-based aggregator, and a single-layer binary classifier. The encoder and aggregator are pretrained in a self-supervised manner to form the representative spatiotemporal context features. Then, the classifier is trained to classify the context features, distinguishing fake videos from real ones. Finally, we propose a selective self-distillation fine-tuning method to further improve the robustness and performance of the detector. In this manner, the extracted features can simultaneously describe the latent patterns of videos across frames spatially and temporally in a unified way, leading to an effective and robust deepfake video detector. Extensive experiments and comprehensive analysis prove the effectiveness of our approach, e.g., achieving a very highest Area Under Curve (AUC) score of 99.94% on FaceForensics++ benchmark and surpassing 12 states of the art at least 7.90%@AUC and 8.69%@AUC on challenging DFDC and Celeb-DF(v2) benchmarks, respectively.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

Additional Key Words and Phrases: Deepfake video detection, representation learning, deep learning, video understanding

This work was partially supported by grants from the National Key Research and Development Plan (2020AAA0140001), Beijing Natural Science Foundation (19L2040), and National Natural Science Foundation of China (61772513). Shiming Ge is also supported by the Youth Innovation Promotion Association, Chinese Academy of Sciences.
Authors' addresses: S. Ge (corresponding author), F. Lin, C. Li, D. Zhang, and W. Wang, Institute of Information Engineering, Chinese Academy of Sciences, and School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China, 100095; emails: geshiming@iie.ac.cn, linfanzhao@iie.ac.cn, lichenyu.iie.ac.cn, zhangdaichi@iie.ac.cn, wangweiping@iie.ac.cn; D. Zeng (corresponding author), School of Communication and Information Engineering, Shanghai University, Shanghai, China, 200444; email: dzeng@shu.edu.cn.

1551-6857/2022/10-ART115
https://doi.org/10.1145/3536426

ACM Reference format:
Shiming Ge, Fanzhao Lin, Chenyu Li, Daichi Zhang, Weiping Wang, and Dan Zeng. 2022. Deepfake Video Detection via Predictive Representation Learning. *ACM Trans. Multimedia Comput. Commun. Appl.* 18, 2s, Article 115 (October 2022), 21 pages.
https://doi.org/10.1145/3536426

1 INTRODUCTION

Deepfake [34] refers to the methods that generate deceptive contents using deep learning approaches, especially images and videos containing manipulated faces. While highly intriguing from computer vision perspective, synthetic videos have a strong impact on the trustworthiness of media content and entail a large number of threats via distorting what is perceived as reality [51], arising concerns and urges about developing deepfake detection technologies. Benefiting from the success of deep generative models, existing approaches for deepfake detection usually focus on controlled scenarios, with poor generalization capacity over different synthesis approaches [47]. Therefore, it is important to develop more general and practical approaches for deepfake video detection.

Many video-based works have proven that the spatiotemporal cues play an important role in improving video representation [13, 20]. Following this, we focus on the spatiotemporal consistency that is often neglected for common framewise video synthesis approaches [6, 27, 45, 46, 48]. To better understand the temporal characteristics of realistic and synthesized videos, we investigate the patterns of temporal variance in both appearance and context spaces. We find that these patterns are different between real and fake videos, as shown in Figure 1 which shows a batch of three continuous frames from a real and a fake video as well as the corresponding context patterns by computing the second-order derivative of feature maps with a pretrained FaceNet [41]. It is easy to notice that, although each frame in the fake videos is generally realistic, the colors and shadows sometimes exhibit unnatural discontinuity among multiple frames. Therefore, the variances in context patterns exhibit differentiable characteristics, such as the patterns extracted from real frames look random while they extracted from the fake frames exhibit an outline similar to the forgery trace (e.g., salient parts in white).

Inspired by the observation, in this work, we propose to focus on the spatiotemporal inconsistency at the context level. Different from previous works that directly enforce the semantic features to represent temporal characteristics, we propose to learn representations from both spatial and temporal dimensions. After the video is sampled and input as batches, a spatial feature extractor first generates local representations, and then the spatial semantic features are fed into the temporal feature extractor, which gives a global representation of video contexts over a longer time. In this way, the model is enforced to learn representations containing spatiotemporal information and help the model capture temporal artifacts for the decision process. To achieve more discriminative representations, we employ predictive learning mechanism [20] to train the spatial and temporal feature extractors in a self-supervised fashion instead of the traditional supervised way [17, 40]. During training, we predict the future states recursively and punish the discrepancies between the predicted states and the true observations. In this way, the feature extractors are enforced to learn the latent patterns of temporal variances. Therefore, the learned model can acutely sense the temporal latent inconsistencies in synthesized videos during inference and we call the model as **Latent Pattern Sensing (LPS)**. In addition, we propose selective self-distillation finetuning to further improve the performance and robustness of the model.

Our main contributions are as follows. First, we discover the fact that there exist discriminative temporal inconsistencies in deepfake videos on both appearance-level and semantic-level. Second,

Fig. 1. General deepfake videos often exhibit appearance-level temporal inconsistencies in some facial components across frames, resulting in discriminative spatiotemporal latent patterns among semantic-level feature maps. The real and fake videos are marked in green and red rectangles, respectively, showing very different spatiotemporal latent patterns (right column).

we propose the Latent Pattern Sensing model for deepfake video detection. A predictive learning mechanism is adopted to train the feature extractors in a self-supervised fashion. Third, we study the intra-video and inter-video variability of classification probabilities predicted by the hard-label learned LPS models and propose a selective self-distillation finetuning method to further improve model performance and robustness. Finally, we conduct extensive experiments on three benchmarks to verify the effectiveness of our approach.

A preliminary version of this work was published in Reference [14]. In this article, we extend the earlier work in three folds. First, we find that the hard-label learned LPS models often give unstable classification probabilities to video instances. Inspired by that, we present a selective self-distillation finetuning method to facilitate robust classification, leading to improved performance. Second, we conduct extensive and comprehensive analysis to study the effect of each component, which reveals the effectiveness and potential of our approach. Third, we demonstrate some failure cases and try to give feasible solutions to address them.

2 RELATED WORKS

Deepfake Video Synthesis. Quickly evolving deep learning approaches provide a general and efficient way to synthesize photo-realistic videos [22]. In the early works, Bregler et al. [6] designed an effective image-based Video Rewrite approach to synthesize fake videos of target person frame by frame. Similarly, Dale et al. [9] proposed an automatic Video Face Replacement model that can swap the faces between different input videos. Thies et al. [45, 46] presented a Face2Face model that can perform real-time expression transfer from origin video to target one.

Recent **generative adversarial networks (GANs)** [16] have promoted the manner and quality of synthesis approaches significantly. With GANs, the synthesis approaches can swap the person identity between the origin and target video [27], manipulate a person's facial attributes [46], and

even synthesize full-body actions [48]. In general, these approaches do not synthesize the whole video frame by frame out of nowhere; instead, they just perform local manipulations on the original video, such as face swap or changing facial attributes. This may imply that there exists some hidden cues between the video frames, which could be the key to uncover photo-realistic deepfake videos.

Deepfake Video Detection. The recent approaches for deepfake video detection mainly utilize the spatiotemporal information to capture informative features. Afchar et al. [1] proposed two networks (Meso-4 and MesoInception-4) that can focus on the mesoscopic properties of the video frames. Liu et al. [30] explored spectrum and other underlying patterns in a video. Luo et al. [32] proposed to utilize the high-frequency noises for face forgery detection to promote the model's generalization ability. Zi et al. [58] proposed to leverage the attention masks on real/fake faces to improve detection. Zhao et al. [55] proposed three strategies for deepfake detection, including attending to different local parts, zooming in the subtle artifacts and aggregating the low-level textural features. Guera et al. [17] combined **Convolution Neural Networks (CNNs)** and RNNs in an end-to-end manner to explore the temporal information to detect fake videos' artifacts. To improve the accuracy of detector, some approaches [18, 36, 43] proposed to utilize the motion patterns of landmarks, which implies the importance of pose tracking [5, 31]. To improve model's adaptation, Kim et al. [25] employed knowledge distillation paradigm to enpower the student model with quick adaption to deepfake images of new kinds. To fully explore the informative cues, Zhu et al. [57] found that the devil lies in the light and identity textures and took them as clues for the deepfake detection. Agarwal et al. [2] proposed using inconsistencies between visemes and spoken phonemes to detect whether a video is real or fake. By exploring more discriminative cues, the video-based approaches often exhibit better performance compared with the image-based approaches.

Predictive Learning for Video Understanding. Predictive learning is a technique of machine learning in which an agent tries to build a model of its environment by trying out different actions in various circumstances. It requires no manual annotation and uses knowledge of the effects its actions appear to have, turning them into planning operators. As a subarea of self-supervised learning, predictive learning has attracted much interest recently. Han et al. [19] trained a model using predictive learning to extract video representations for video prediction task. Han et al. [20] further used frames' semantic prediction to capture the movements in videos for action recognition by introducing an extra memory module to augment the effects. Denton et al. [11] introduced a new adversarial loss in corporation with the recurrent predictive learning framework. The approach can learn representations to factorize each frame into a stationary part. Babaeizadeh et al. [13] combined RNNs and variational autoencoders to cope with the uncertainty of the spatiotemporal sequences, i.e., the multi-modal mappings from the historical observations to future frames. Recent work of He et al. [21] proposed masked autoencoders to achieve scalable vision task learning. As verified by these approaches, predictive learning is an effective solution that can extract more intrinsic and robust information in videos.

3 PROPOSED APPROACH

3.1 Problem Formulation

Given a video $\mathbb{V} = \{\mathbf{x}_i\}_{i=1}^{n}$ consisting of n frames, the aim of deepfake video detection is to learn a binary classifier $l(\cdot)$ to determine its authenticity. To this end, our video-level detection approach takes a clip sequence as input. Each clip includes c continuous frames, and the model combines information gathered from multiple clips to identify whether the video is real,

$$p(r=0|\mathbb{V}) = \bigcup_{t=1}^{n-c+1} p(r=0|\mathbb{C}_t) = \bigcup_{t=1}^{n-c+1} l(f(\mathbb{C}_t;\boldsymbol{W}_f);\boldsymbol{W}_l), \tag{1}$$

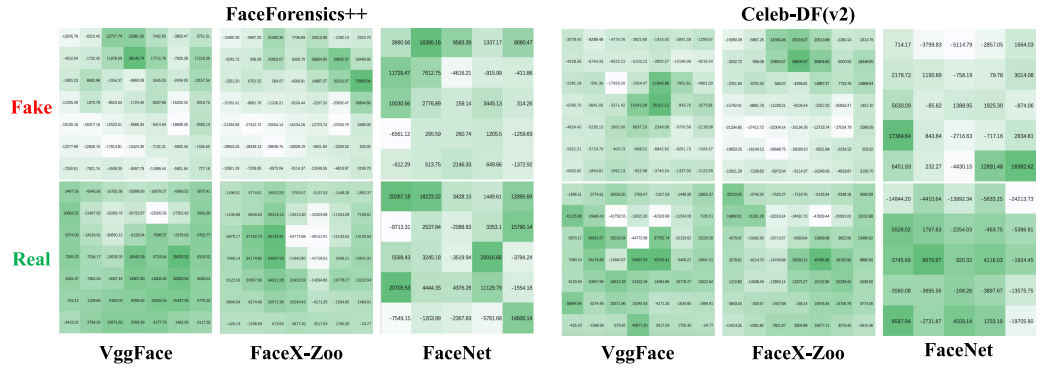

Fig. 2. Distribution of semantic features for real and fake videos on two datasets (FaceForensics++ [39] and Celeb-DF(v2) [27]) generated by three face recognizers (VggFace [38], FaceX-Zoo [50], and FaceNet [41]).

where $p(r = 0|\mathbb{V})$ and $p(r = 0|\mathbb{C}_t)$ denote the possibility of the input video $\mathbb{V}$ and a single clip (input example) $\mathbb{C}_t$ being fake, respectively; $\mathbb{C}_t = \{\mathbf{x}_t, \ldots, \mathbf{x}_{t+c-1}\}$ consists of c continuous frames starting from the t th frame; f is the feature extractor with parameters $\boldsymbol{W}_f$ and l denotes the classifier with parameters $\boldsymbol{W}_l$; and $\bigcup$ is a combination operator. From Equation (1), we can find that the main components include feature extraction, feature classification, and prediction combination, and we need to address the key challenge: How do we learn discriminative representations for effectively describing the spatiotemporal deepfake traces?

Toward this end, we first experimentally study the effectiveness of spatiotemporal information in video representation ability by predictive learning [20]. To find out how the learned features reflect spatiotemporal relations, we design a simple yet illuminating experiment. We collect 1,000 real and 1,000 fake videos from the Celeb-DF(v2) [28] and FaceForensics++ [39] datasets. We randomly extract three continuous frames from each video and then feed them into three pretrained face recognizers (VggFace [38], FaceX-Zoo [50], and FaceNet [41]) to obtain the semantic features. In the experiment, we extract the output of the last convolutional layer and take an average over all the channels. The resulted two-dimensional feature maps are taken as the semantic features, with the size of 7×7 for VggFace and FaceX-Zoo and 5×5 for FaceNet. Finally, we get the features' second derivative by adding the first and third frame's semantic features and subtracting two times of the second frame's semantic features. As shown in Figure 2, several interesting patterns appear. The feature maps of the fake video generally look alike, having a prominent center of high feature values, denoted in darker color. On the contrary, the feature maps of real videos are different for different recognizers and datasets. They all exhibit an irregular distribution with diffused high feature values. The clear differences between fake and real ones are found on different datasets and recognizers, proving the discovery to be common. Moreover, we also find discrepancies in specific values. The real videos' feature maps' values are closer and the fake ones are the opposite. The above conclusions provide enough insights for us to easily distinguish the real and fake videos.

Inspired by the above observation, we propose to fully utilize the spatiotemporal information within the video clip by introducing another aggregator $g(\cdot)$. It performs combination in the temporal dimension by aggregating the spatial features extracted by $f(\cdot)$ instead of combining classification results, enforcing the model to exploit temporal characteristics and discriminate the fake videos. Moreover, to reduce feature extraction computation and temporal redundancy, the input examples are sampled without overlapping. In this way, the deepfake video detection problem can

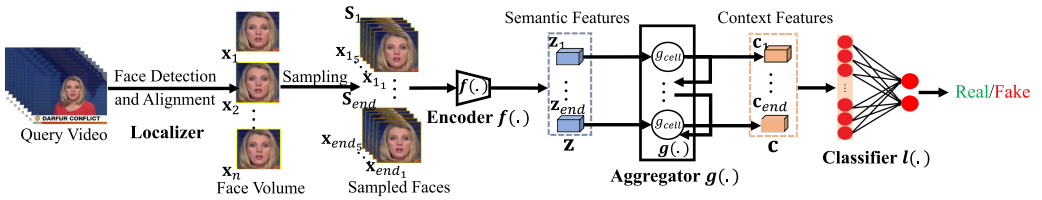

Fig. 3. The framework of our LPS deepfake video detector. It first uses localizer to detect and align faces from a query video. Then the face images are fed into encoder $f(\cdot)$ to extract spatial semantic features, following by aggregator $g(\cdot)$ to achieve spatiotemporal context features. Finally, classifier $l(\cdot)$ predicts the context features to identify the video into real or fake.

be reformulated as

$$p(r=0|\mathbb{V}) = \bigcup_{t=1}^{m} p(r=0|\mathbb{S}_t) = l(g(\{f(\mathbb{S}_t;\boldsymbol{W}_f)\}_{t=1}^{m};\boldsymbol{W}_g);\boldsymbol{W}_l), \tag{2}$$

where $m = \lfloor n/c \rfloor$ is the number of input examples, $\mathbb{S}_t = \mathbb{C}_{t\times c-b}$ is a sampled example, $b < c$ is a random index denoting the start frame, $\boldsymbol{W}_g$ denotes the parameters for the aggregator $g(\cdot)$, and $\lfloor\cdot\rfloor$ indicates the floor function.

3.2 Latent Pattern Sensing

As shown in Figure 3, our LPS framework consists of four main modules: (1) localizer, which detects and aligns the face region to prepare the input examples; (2) encoder $f(\cdot)$, which extracts spatial semantic features from each example; (3) aggregator $g(\cdot)$, which aggregates spatial semantic features into spatiotemporal context features; and (4) classifier $l(\cdot)$, which identifies spatiotemporal context features and gives the final discrimination result.

Localizer. It first detects the faces and landmarks from frames by a pretrained MobileNet[1] and a facial landmark detector,[2] respectively. Then, the faces are aligned and cropped into the size of 224×224 with similarity transformation. Besides, we take continuous 60 frames from a random starting position and form a group of every 5 frames to serve as an input example.

Encoder $f(\cdot)$. It learns the spatial semantic feature of each short video clip by a CNN. Specifically, we use a convolutional layer $conv_1$ to get each frames' semantic features and two pooling layers $pool_1, pool_2$ for eliminating redundancy. Moreover, to capture the subtle artifacts intricately distributed in various places, we use four residual modules $res_1, res_2, res_3, res_4$ to increase model complexity and capture global features including more details. All kernels are three-dimensional (3D) tensors, and the dimension of the output $f(\cdot)$ is 4, organized as $time\times height\times width\times channel$. All the spatial semantic features are then concatenated in order for later processing in Aggregator $g(\cdot)$, denoted as $\mathbf{Z} = \{\mathbf{z}_t\}_{t=1}^{m}$, where $\mathbf{z}_t = f(\mathbb{S}_t;\boldsymbol{W}_f)$.

Aggregator $g(\cdot)$. This module learns the descriptor of spatiotemporal information across whole video frames. It aggregates former embedding sets from encoder $f(\cdot)$. Considering the temporal attribute of videos, we regard the changes across continuous frame sets as the changes in temporal. So the spatiotemporal context features can be modeled as $\mathbf{c}_t = g(\mathbf{z}_1, \ldots, \mathbf{z}_t; \boldsymbol{W}_g)$. As for $g(\cdot)$, we adopt a common ConvGRU cells [4], in which two different activation functions, sigmoid σ and tanh $tanh$ are used (shown in Figure 4). As for a recursive neural networks, the information

[1] https://github.com/yeephycho/tensorflow-face-detection.
[2] https://github.com/1adrianb/face-alignment.

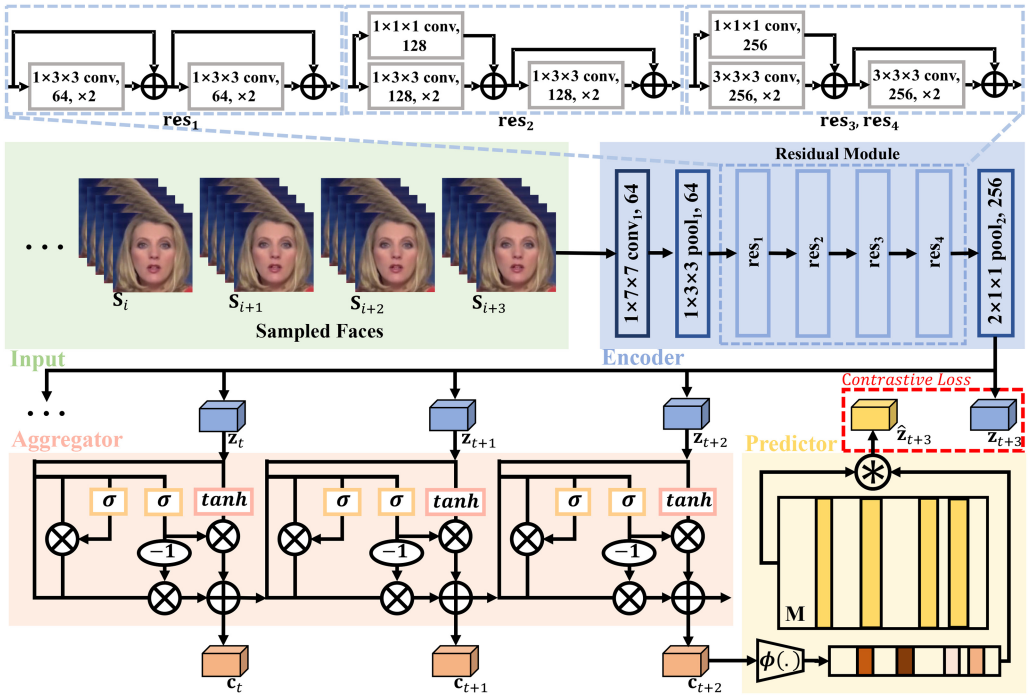

Fig. 4. The predictive representation learning framework. It uses the sampled faces to train encoder, aggregator, and predictor. The predictor produces predicted spatial semantic features $\hat{\mathbf{z}}_{t+3}$ by spatiotemporal context feature $\mathbf{c}_{t+2}$ and calculates the contrastive loss. $\mathbf{z}_*$ and $\mathbf{c}_*$ denote spatial semantic features and spatiotemporal context features, respectively. $\mathbf{M}$ is the compressive memory bank.

received by each ConvGRU cell is passed on to all the following cells. Then we achieve the transfer of information and can capture the changes of $\mathbf{z}_t$ in temporal. Meanwhile, there are three gates contained in each activation function and also convolutional layers. We also concatenate all of the spatiotemporal context features that are then fed into classifier.

Classifier $l(\cdot)$. Since the achieved spatiotemporal context features from the encoder and aggregator are discriminative, we use a fully connected layer to construct classifier $l(\cdot)$, which takes spatiotemporal context features as input and outputs two probabilities of being fake and real for authenticity, as shown in Equation (2).

3.3 Predictive Representation Learning

Predictive representation learning is a kind of training method to improve the representation ability via iterative prediction. In our work, we employ it as the pretraining step to make the backbone focus on semantic changes. The intuition behind the predictive task is that if one can infer future semantics from the present ones, then it means the representations must have encoded rich context about the temporal characteristics. Thus, the pretraining datasets are not limited to the downstream task datasets, but we just use the deepfake video training datasets for better matching. We use both real and fake videos in the training datasets to promote the model's representation ability.

To learn representative features via prediction, we first adopt the same combination of encoder $f(\cdot)$ and aggregator $g(\cdot)$ as the backbone to represent former spatiotemporal context features $\mathbf{c}_t = g(\mathbf{z}_1, \ldots, \mathbf{z}_t; \boldsymbol{W}_g)$ for predicting future state (spatial semantic features). According to our discovery,

the artifacts are distributed in a face image. Thus we adopt memory mechanism [20] to encourage diverse prediction results [3], enforcing a general and robust predictor. As shown in Figure 4, the predictor is composed by a compressive memory bank and a **multi-layer perception (MLP)** $\phi(\cdot)$ with parameters $\boldsymbol{W}_\phi$. The memory bank $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_K]^{\mathrm{T}} \in \mathbb{R}^{K \times C}$ consists K slots of compressive memory and the dimension of compressive memory C is equal to the channel number of feature maps. We use random values that satisfy the standard normal distribution to initialize the memory bank $\mathbf{M}$ and update its values with back propagation algorithm. The predictive hypothesis is $\mathbf{p}_{t+1} = Softmax(\phi(\mathbf{c}_t; \boldsymbol{W}_\phi))$ where the $Softmax$ function is used to normalize the outputs from ϕ. The normalized value will not enlarge or reduce the scale of the feature maps in the iterative process, which makes the training process difficult. And it can also map them to non-negative space smoothly and prevent the selected memory slots too sparse, and the predicted future state is then computed recursively with $\hat{\mathbf{z}}_{t+1} = \sum_{i=1}^{K} \mathbf{p}_{(i,t+1)} \cdot \mathbf{m}_i = \mathbf{p}_{t+1} \cdot \mathbf{M}$, where $\mathbf{p}_{(i,t+1)} \in \mathbb{R}^{B \times H \times W}$ referring to the contribution of ith memory slot with the same batch size B, the height H, and the width W as the feature map $\mathbf{z}_t$. $\phi(\cdot)$ projects the context features to each memory slot's hypothesis $\mathbf{p}_{(i,t+1)}$, utilizing the former information to predict the future. Considering the parameters of memory bank and predictor are updated synchronously, to simplify the equation, we just use $\boldsymbol{W}_\phi$ to refer to both parameters in the following. The general predictor $\mathcal{P}(\cdot)$ can be formulated as

$$\begin{aligned} \hat{\mathbf{z}}_{t+1} &= \mathcal{P}(\mathbb{S}_1, \ldots, \mathbb{S}_t; \boldsymbol{W}_f, \boldsymbol{W}_g, \boldsymbol{W}_\phi, \mathbf{M}) \\ &= Softmax\left(\phi(g(\{f(\mathbb{S}_1; \boldsymbol{W}_f), \ldots, f(\mathbb{S}_t; \boldsymbol{W}_f)\}; \boldsymbol{W}_g); \boldsymbol{W}_\phi)\right) \cdot \mathbf{M}. \end{aligned} \tag{3}$$

Considering the predictor produces multiple possible hypotheses, it is difficult to ensure the correctness of the hypotheses while guaranteeing possible future states' diversity. Thus we use the contrastive loss to ask the model to predict future states by assigning higher similarity to the true observation than other observations (from different videos or from elsewhere in the same video) rather than abandoning them. The contrastive loss is defined as

$$\mathcal{L}_{con}(\hat{\mathbf{z}}_{i,k}, \mathbf{z}_{i,k}) = \mathbb{E}\left[-\sum_{i,k} \log \frac{\exp\left(\hat{\mathbf{z}}_{i,k}^{\mathrm{T}} \mathbf{z}_{i,k}\right)}{\exp\left(\hat{\mathbf{z}}_{i,k}^{\mathrm{T}} \mathbf{z}_{i,k}\right) + \sum_{(j,m) \neq (i,k)} \exp\left(\hat{\mathbf{z}}_{i,k}^{\mathrm{T}} \mathbf{z}_{j,m}\right)}\right], \tag{4}$$

where $\mathbb{E}$ measures the expectation. Since each semantic feature $\mathbf{z}$ and $\hat{\mathbf{z}}$ are dense feature maps extracted from different training videos and multiple temporal clips, we represent the index into two parts for clarity. Thus, we denote temporal index with i and batchwise as well as spatial index as k, where spatial index means the index of each value in the dense feature map and batchwise index means the index in the current mini-batch, $k \in \{(1,1,1), (1,1,2), \ldots, (B,H,W)\}$, and use $\mathbf{z}_{i,k}$ as true spatial semantic features and $\hat{\mathbf{z}}_{i,k}$ as predicted features for calculating contrastive loss. Thus, only the predicted states and the true observations from the same video and spatiotemporal aligned position are positive pairs. To summarize, the better the prediction, the more similar the positive pairs. We conduct the predictive representation learning as an energy minimization problem:

$$\min_{\boldsymbol{W}_f, \boldsymbol{W}_g, \boldsymbol{W}_\phi, \mathbf{M}} \sum_{\mathbb{S}, \mathbb{S}'} \sum_{t=3}^{end} \mathcal{L}_{con}\left(\mathcal{P}(\mathbb{S}_1, \ldots, \mathbb{S}_t; \boldsymbol{W}_f, \boldsymbol{W}_g, \boldsymbol{W}_\phi, \mathbf{M}), f(\mathbb{S}'_{t+1}; \boldsymbol{W}_f)\right), \tag{5}$$

where the total energy is accumulated in all training examples and *end* indicates the frame number in an example. $\boldsymbol{W}_f$ and $\boldsymbol{W}_g$ are trained with Equation (5) and then fine-tuned [42] in detector learning mode that incorporates classifier together. Both of $\mathbb{S}$ and $\mathbb{S}'$ are input sampled examples, and they can be from the same videos for positive pairs or different videos for negative pairs in Equation (5). The algorithm flowchart is given in Algorithm 1.

The implementation details are as follows. In encoder $f(\cdot)$, the first convolutional layer $conv_1$ has a kernel size of $1 \times 7 \times 7$. The residual modules can be divided into two pairs: res_1 and res_2 have

ALGORITHM 1: Predictive representation learning for videos

Require:
f: encoder with parameters $\boldsymbol{W}_f$, composed by 2d3d-ResNet;
g: aggregator with parameters $\boldsymbol{W}_g$, composed by ConvGRU;
ϕ: predictor with parameters $\boldsymbol{W}_\phi$, composed by MLP;
$\mathcal{S}$: all input sampled examples;
$\mathbb{S}$: one mini-batch of input sampled examples;
$\mathbf{M}$: the compressive memory bank;
$\mathbf{z}$: spatial semantic features;
$\hat{\mathbf{z}}$: predicted semantic features;
$\mathbf{c}$: spatiotemporal context features;
$\mathbf{p}$: hypothesizes for each memory slots;
end: the number of frames in each example;

for $\mathbb{S} \in \mathcal{S}$ **do**
  $\mathbf{z}_i = f(\mathbb{S}_i; \boldsymbol{W}_f),\ i = 1, \ldots, end$
  for $i = 3$ *to* $end-1$ **do**
    $\mathbf{c}_i = g(\mathbf{z}_1, \ldots, \mathbf{z}_i; \boldsymbol{W}_g)$
    $\mathbf{p}_{i+1} = Softmax(\phi(\mathbf{c}_i; \boldsymbol{W}_\phi))$
    $\hat{\mathbf{z}}_{i+1} = DotProduct(\mathbf{p}_{i+1}, \mathbf{M})$
    $\mathbf{z} = Concat(\mathbf{z}, \mathbf{z}_{i+1})$
    $\hat{\mathbf{z}} = Concat(\hat{\mathbf{z}}, \hat{\mathbf{z}}_{i+1})$
  end for
  $loss = CrossEntropyLoss(\hat{\mathbf{z}}, \mathbf{z})$
  $loss.backward()$
end for
return

the same kernel size 1×3×3, while res_3 and res_4 have the same kernel size 3×3×3. Meanwhile, the number of filters changes to 64, 128 and 256 with the size of feature map halving. For aggregator $g(\cdot)$, the convolutional kernel size is $1 \times 1 \times 1$. The fully connected layer in detector has 512 input channels and 2 ways for output. The size of the compressive memory bank $\mathbf{M}$ is $1{,}024 \times 256$.

3.4 Selective Self-Distillation Fine-tuning

Intuitively, the deepfake videos usually are synthesized in different photo-realistic degrees between video examples as well as among different clips within a video. As a result, a latent pattern sensing model should distinctively consider video training examples in learning. However, a video example usually is annotated into "real" or "fake" with a hard binary label. Therefore, the resulting latent pattern sensing model learned with hard labels cannot consistently identify the clips within a video. Figure 5 shows some examples of the fake probability on different clips within videos predicted by the learned models. We can see that some video clips are identified as fake with high probability while some fake video clips are misclassified. This may reduce the robustness of a model [23].

To take advantage of the inconsistent authenticity existing in deepfake videos, we present a selective self-distillation fine-tuning method to further improve the model. Inspired by selective knowledge distillation [15], we use the trained latent pattern sensing model with the above predictive representation learning as the teacher model, which consists of an encoder, aggregator, and classifier and is indicted as $\phi_t = \{f, g, l\}$ with parameters $\boldsymbol{W} = \{\boldsymbol{W}_f, \boldsymbol{W}_g, \boldsymbol{W}_l\}$. We are given N

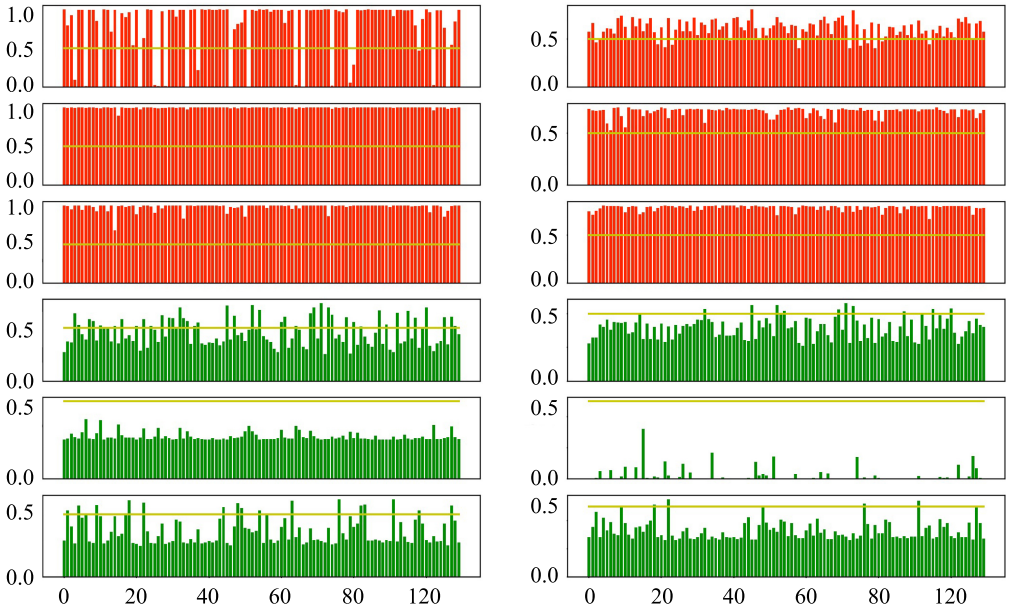

Fig. 5. Fake probabilities on continuous clip examples within some real and fake videos identified by the predictive representation learned LPS models (left) and selective self-distillation fine-tuned LPS models (right). The abscissa axis and ordinate axis represent the clip example and fake probability, respectively. The fake and real videos are in red and green, respectively.

training examples $\mathcal{T} = \{(\mathbb{X}_i, y_i)\}_{i=1}^{N}$ where $\mathbb{X}_i$ and $y_i \in \{0, 1\}$ are the data and the groundtrue hard label of the ith example. Then, each example can be identified by the teacher model, forming the set of soft labels $\hat{\mathcal{L}} = \{\hat{\mathbf{y}}_i\}_{i=1}^{N}$ where $\hat{\mathbf{y}}_i = \phi_t(\mathbb{X}_i; \boldsymbol{W})$ is a 2D vector. Finally, the selective self-distillation finetuning can be formulated as an optimization problem that can be solved by minimizing the following loss function $\mathcal{L}_{ssd}$:

$$\mathcal{L}_{ssd}(\boldsymbol{W}^+, \mathcal{T}) = \lambda \sum_{\mathcal{T}} \ell\left(\phi_s(\mathbb{X}_i; W^+), y_i\right) + (1-\lambda) \sum_{\mathcal{T},\ \arg\max \hat{\mathbf{y}}_i = y_i} KL\left(\phi_s(\mathbb{X}_i; \boldsymbol{W}^+), s(\hat{\mathbf{y}}_i, T)\right), \quad (6)$$

where $\boldsymbol{W}^+ = \{\boldsymbol{W}_f, \boldsymbol{W}_g^+, \boldsymbol{W}_l^+\}$ are the parameters of the student ϕ_s, ϕ_s has the same network architecture with the teacher ϕ_t, ℓ is a cost function to measure the classification loss and we use a cross-entropy function, $\lambda \in [0, 1]$ is a tuning factor to balance the effect of two loss terms and we set $\lambda = 0.7$ in our experiments, KL denotes the KL divergence to measure the matching of two probability distributions, and $s(\cdot)$ is a soften function with a temperature T. Using a higher value for T produces a softer probability distribution over classes [23], and we set $T = 1.0$ in our experiments. In practice, the performance will get worse when λ is lower than 0.7, while the model traps into overfitting when λ is higher than 0.7. the situation is similar for the temperature T. In Equation (6), the first term is classification energy that is measured on the whole examples, and the second term is selective distillation energy that is measured on the examples classified correctly by the teacher. This selective manner can mitigate the transfer of incorrect knowledge from the teacher [15]. In the approach, we fix the encoder parameters and only fine-tune the classifier parameters and part of the aggregator parameters, leading to efficient optimization.

The optimization is easy and direct. We first take the pretrained latent pattern sensing model as the teacher and then calculate the outputs from the classifier as the soft labels. The student inherits the same weights $\boldsymbol{W}$ for initialization and then is fine-tuned with both hard and soft labels by

minimizing Equation (6). Aiming at the comprehensiveness and accuracy of forgery trace capture, we design a selective distillation loss. As for the correctly classified examples, the student network just makes little adjustments according to the soft labels, while the misclassified examples will be further corrected using the hard labels for capturing more latent patterns. As shown in the right of Figure 5, the fine-tuned LPS models deliver more stable predictions while reducing errors.

4 EXPERIMENTS

4.1 Experimental Settings

We conduct experiments on three popular benchmarks: FaceForensics++ [39], DFDC [12], and Celeb-DF(v2) [27]. FaceForensics++ consists of 1,000 original video sequences from YouTube and 4,000 corresponding manipulated videos with four different manipulation methods including Deepfakes,[3] FaceSwap,[4] NeuralTextures [44], and Face2Face [46]. We split the dataset into training, validation, and testing sets with a ratio of 6:2:2. DFDC consists of 19,197 real videos from 430 actors and 100,000 fake videos. We also randomly split the dataset with a ratio of 6:2:2, obtaining 7,528 training videos, 2,482 validation videos, and 2,541 testing videos. Celeb-DF(v2) contains 590 original videos with subjects of different ages, ethnic groups, and genders, and 5,639 corresponding synthesized videos, which makes it more challenging for deepfake detection.

To train the models, we set the batch size as 16 and the total epoch as 500 and use the Adam [26] optimizer. The initial learning rate is 10^{-3} decaying to 10^{-4} when the validation loss plateaus. The experiments are implemented with Pytorch on NVIDIA TITAN Xp GPU and 2.6-GHz Intel CPU.

4.2 State-of-the-Art Comparison

We make comparisons with 12 state-of-the-art approaches. We use the **Area Under Curve (AUC)** score as a metric, since it can well eliminate the interference of human factors. The results are shown in Table 1. We can find that these approaches achieve different scores across different datasets, which reveals detection difficulties and forgery traces are different across datasets. Despite this, our LPS model still achieves at least 0.03%, 5.04%, and 7.92% higher than other approaches on FaceForensics++, DFDC, and Celeb-DF(v2), respectively. The selective self-distillation fine-tuned LPS_{ssd} model further improves the performance. The results imply a robust representation learning ability of our approach in deepfake video detection. We bold our proposed method in the first column and also bold the state of art results achieved by our models in Table 1.

We also analyze our model's ROC curve, which can better eliminate the interference of human factors in setting different thresholds. The results from LPS model are shown in Figure 6. It is obvious that the results on two datasets are very close to the full coverage of the entire coordinate plane, which means our model can accurately identify all positive samples (True Positive Rate = 1) while ensuring that negative samples will not be misjudged (False Positive Rate = 0). In other words, all of the samples can be almost classified correctly. The results demonstrate that our detector can achieve near perfect discriminate ability with proper threshold settings. Meanwhile, as for the two curves, when the false-positive rates are the same, the true-positive rates from Celeb-DF(v2) are higher than those from FaceForensics++, which is consistent with the accuracy of the results.

Besides, to demonstrate effectiveness of the predictive representation learning, we further train an extra model $\textbf{LPS}_{ri}$ by first randomly initializing with random parameters and then learning in an end-to-end manner like [40]. As shown in Table 1, it is observed that the predictive representation learning boosts detection performance by around 10% on average, proving its efficient pattern learning ability with a better initial starting point.

[3] https://github.com/deepfakes/faceswap.
[4] https://github.com/MarekKowalski/FaceSwap/.

Table 1. AUC Score (%) Comparisons with 12 State-of-the-Art Approaches on FaceForensics++, DFDC, and Celeb-DF(v2) Datasets

Approach	FaceForensics++	DFDC	Celeb-DF(v2)	Publication
MesoNet [1]	—	75.30	54.80	WIFS 2018
Visual Artifacts [33]	78.00	66.20	55.10	WACVW 2019
Multi-task [36]	76.30	53.60	54.30	BTAS 2019
Audio-Visual [35]	—	84.40	—	MM 2020
Face+Context [37]	75.00	—	66.00	TPAMI 2021
SPSL [30]	96.91	66.16	—	CVPR 2021
Multi-attention [55]	97.60	—	—	CVPR 2021
LipForensics [18]	97.10	73.50	82.40	CVPR 2021
TD-3DCNN [54]	72.22	78.97	88.83	IJCAI 2021
LRNet [43]	99.90	—	—	CVPR 2021
FD^2Net [57]	99.45	66.09	—	CVPR 2021
FTCN [56]	99.70	74.00	86.90	ICCV 2021
Our LPS$_{ri}$	81.25	80.25	92.55	—
Our LPS	**99.93**	**89.44**	**96.75**	—
Our LPS$_{ssd}$	**99.94**	**92.30**	**97.52**	—

The subscript ri means the model adopts random initialization, and ssd denotes the model is finetuned by selective self-distillation.

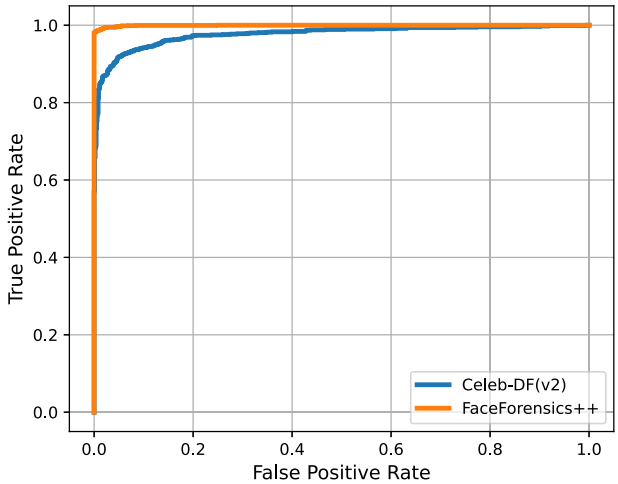

Fig. 6. ROC curves on Celeb-DF(v2) and FaceForensics++ with **LPS** model.

4.3 Generalizability Analysis

To demonstrate the model's generalization ability, we conduct cross-dataset experiments. The results are presented in Table 2. We discover that using Celeb-DF(v2) as a training set gets better testing results generally, proving it contains more representative features in all kinds of fake videos. Although the results vary across different datasets, we can also find that their AUC scores are totally over 70% with LPS and 80% with LPS$_{ssd}$. Meanwhile, as for deepfake datasets, the fake samples are usually 2 or 3 times more than real samples and fake clues are too weak to confuse with real samples [53]. When learning on unbalanced training samples, the CNN-based detector easily

Table 2. The Results of Cross-dataset Generalization Ability

Train set	Test set	AUC (%)	Recall (%)	Precision (%)
FaceForensics++	FaceForensics++	99.93/99.94	92.14/92.33	96.17/96.18
	DFDC	73.24/88.74	89.77/91.05	81.40/85.37
	Celeb-DF(v2)	87.72/90.17	91.61/94.39	87.35/90.41
DFDC	FaceForensics++	77.91/80.57	89.53/91.62	88.14/89.17
	DFDC	89.44/92.30	90.11/92.06	88.17/93.21
	Celeb-DF(v2)	84.22/88.52	91.55/94.23	84.23/85.74
Celeb-DF(v2)	FaceForensics++	88.69/90.66	98.15/98.74	89.25/92.59
	DFDC	77.35/80.29	94.62/95.03	83.19/85.11
	Celeb-DF(v2)	96.75/97.52	89.57/95.01	93.26/97.38

The learned and finetuned models are trained and tested on three datasets: FaceForensics++, DFDC, and Celeb-DF(v2). The AUC, Recall, and Precision are used for evaluation. We show the results from LPS and LPS$_{ssd}$ models as "LPS/LPS$_{ssd}$."

falls into the trap of overfitting [24], especially in the DFDC dataset [10]. To evaluate the model robustness against overfitting, we also report the Recall and Precision scores that are mainly defined with true positive, false negative, and false positive, which calculate the effect of positive samples' proportion. Thus, Recall and Precision are suitable to measure the overfitting situation [7]. From Table 2, the Recall and Precision scores suggest that our model well avoids overfitting and obtains an impressive performance on different datasets. We also note that the AUC on Celeb-DF(v2) reaches 84.22% with LPS and 88.52% with LPS$_{ssd}$ even when training on DFDC, which still exceeds many prior approaches such as Multi-task [36] and MesoNet [1]. We also note that the selective self-distillation fine-tuned LPS$_{ssd}$ consistently improves model performance. These results show the generalization ability of our approach and the strong adaptability of the learned representations.

4.4 Component Analysis

After the promising detection performance is achieved, we further analyze the impact of each component in our approach, including localizer, encoder, aggregator, predictor, memory bank, classifier, and selective self-distillation fine-tuning.

Localizer. The localizer mainly performs face detection and then face alignment. Figure 7 shows some examples, where face alignment can make the facial regions more complete and improve the detection performance. To study this effect, we conduct experiments on three benchmarks by removing the face alignment function and just using the detected faces as input. The results are shown in Table 3. We can see that the detection performance drop happens consistently over three benchmarks on all three metrics. For example, without face alignment, it has an AUC score drop of 3.27%, 5.89%, and 4.64% on FaceForensics++, DFDC, and Celeb-DF(v2), respectively. Their results reveal the importance of landmark localization for capturing artifacts. The main reason is that face alignment can make the input faces more standardized with the unified center position, rotation angle and scaling ratio, which is convenient for the encoder to capture the characteristics of human faces and further facilitate representation enhancement by the aggregator.

Encoder. The encoder serves as the basic feature extraction and is very critical for video representation. To study its effect, we take the original detector as a baseline and conduct two ablation experiments by removing it and enhancing it.

First, we remain the aggregator and remove the encoder from the deepfake video detector learning framework during the whole training process. Different from the original baseline framework, the temporal information is mainly captured by the aggregator, thus only one aggregator

Fig. 7. Some examples of facial images are generated by face detection and alignment (top) and only face detection (bottom). It is obvious that face alignment can make the facial regions more complete and standardized, which is very helpful to improve deepfake detection performance.

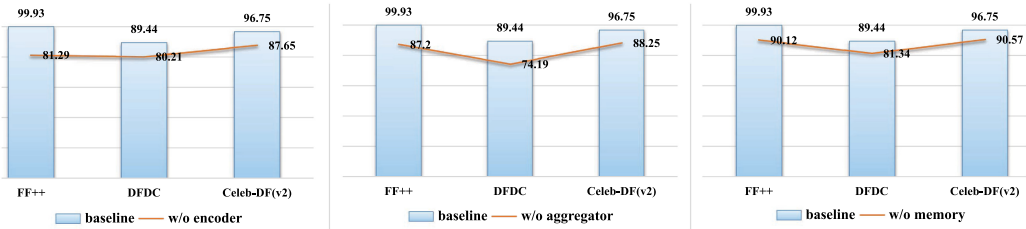

Fig. 8. The component effect on AUC score (%) over three benchmarks. Left: encoder. Center: aggregator. Right: memory bank.

Table 3. The Effect of Face Alignment in Localizer: AUC (A), Recall (R), and Precision (P) Are Evaluated on Three Benchmarks

Face Alignment	FaceForensics++			DFDC			Celeb-DF(v2)		
	A(%)	R(%)	P(%)	A(%)	R(%)	P(%)	A(%)	R(%)	P(%)
with	99.93	92.14	96.17	89.44	90.11	88.17	96.75	89.57	93.26
without	96.66	90.18	90.22	83.55	90.05	83.27	92.11	84.23	90.61
performance ↓	−3.27	−1.96	−5.95	−5.89	−0.06	−4.90	−4.64	−5.34	−2.65

can complete a certain degree of feature extraction. In this case, the spatial details extracted by the original encoder will be weakened or ignored. As shown in Figure 8, when comparing with the baseline model, the AUC score decreases by over 18% on FaceForensics++ and almost 10% on DFDC and Celeb-DF(v2). The main reason comes from the difficulty of deepfake video detection in which forgery trace is too subtle and scattered in spatial due to high realistic frame synthesis.

To further study the importance of the encoder, we conduct the second experiment by modifying the encoder network. The baseline encoder is composed of 2d3d-ResNet18, which is a kind of 3DCNN for extracting short-term temporal information. Inspired by Reference [54], we replace it with 3D Inception [8] and pretrain the encoder on the deepfake detection task. Then, we train a new LPS model termed $\mathbf{LPS}_e$ on Celeb-DF(v2) with the same setting and conduct a comparison with the baseline. As shown in Table 4, all metrics have increased by varying degrees, for example, the AUC score, recall, and precision are increased by 1.31%, 1.84% and 2.22%, respectively. These results reflect the excellent effect of the enhanced encoder. As mentioned in Reference [54], the 3D Inception network is more complicated than 3D ResNet18. It contains four different branches for focusing on more details on different scales, resulting in the better capturing ability of spatial features.

Table 4. The Effect of Encoder and Classifier on Celeb-DF(v2)

Model	AUC (%)	Recall (%)	Precision (%)
LPS	96.75	89.57	93.26
LPS$_e$	98.06 (+1.31)	91.41 (+1.84)	95.48 (+2.22)
LPS$_c$	97.35 (+0.60)	90.31 (+0.74)	93.59 (+0.33)

LPS$_e$ replaces encoder with 3D Inception and **LPS**$_c$ replaces classifier with SVM. It implies that more discriminative encoder and classifier can improve model performance.

Aggregator. Similarly to the analysis on the encoder, we remain the encoder and remove the aggregator with the same settings of learning baseline detection models. Then, we train the new models on three datasets and compare them with the baseline. The results are reported in Figure 8. As we thought, the performance also gets a big drop on all three benchmarks. It is because the encoder composed by 2d3d-ResNet18 can only capture short-term temporal features and is unable to effectively describe spatiotemporal context clues. Although we contacted all the short-term outputs for classification, the resulting representations still contain so much redundant information that the long-term context information is not extracted effectively. The results show that an effective deepfake video detector needs to identify not only short-term temporal inconsistency but also long-term one, which can be demonstrated again in the encoder's experiment. Thus, our approach cascades encoder and aggregator into a unified backbone for video representations.

Memory Bank. Besides, the predictive representation learning as our main idea, we also conduct an ablation experiment on its core component, the memory bank in the predictor, for proving the effectiveness of our predictor design. As we introduced in Subsection 3.3, the memory bank can provide more possible results for prediction diversity. In practice, the memory bank is updated by new semantic features from the encoder in each epoch. We just remove the memory bank and force the multi-layer perceptron $\phi(\cdot)$ to predict the specific future states. Compared with computing each memory slot's hypothesis, this target task is more difficult. The most obvious change is reflected in the training process. After removing the memory bank, the training loss becomes larger and the convergence is slower. In Figure 8, the downstream task is also reflected with decreased results.

Classifier. Furthermore, we analyze the classifier's effect on our detection model. Both in the training and testing phase, the classifier plays an important role. In the training phase, our core objective is to learn an effective feature extractor as well as the backbone. However, the back propagation algorithm decides the training process relying on the outputs from the classifier. If the discrimination of output results is better, then the training efficiency is higher. In the testing phase, the discrimination of output results is just our objective. Such two situations greatly rely on the classifier's mapping ability. When the boundaries between different types of feature points in the feature space are clear, traditional machine learning methods, such as **support vector machine (SVM)**, which pay attention to the maximum interval of classification surfaces, often have better results. To validate this, we train a new model termed **LPS**$_c$ on Celeb-DF(v2) by replacing the baseline classifier with SVM and conducting a comparison with the baseline **LPS** model. The results are reported in Table 4, where the new model **LPS**$_c$ achieves improved performance over the baseline, such as an AUC improvement of 0.6%, implying the effect of a more discriminative classifier.

Selective Self-Distillation Finetuning. A robust deepfake video detector should stably identify a video example as real or fake. The proposed selective self-distillation fine-tuning method aims to make the models robust and generalizable. As shown in Table 1, compared with the baseline **LPS**, the fine-tuned **LPS**$_{ssd}$ can consistently improve the detection performance on all three benchmarks. Furthermore, as illuminated in Figure 5, the classification probabilities on video examples

Table 5. The Effect of Hyperparameters λ and T on Celeb-DF(v2) in Selective Self-distillation Finetuning

$T = 1$	λ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
	Loss	0.57	0.45	0.44	0.39	0.32	0.29	0.15	0.27	0.31	0.41
$\lambda = 0.7$	T	1	2	3	4	5	6	7	8	9	10
	Loss	0.15	0.38	0.51	0.58	0.66	0.79	0.82	0.85	0.94	1.01

identified by $\mathbf{LPS}_{ssd}$ are more stable, e.g., most values are around 0.5 for fake videos. The main reason is that the method inherits the advantages of knowledge distillation [23], which boosts the robustness and generalizability. We experimentally study the effect of the hyper-parameters λ and T in Equation (6) to find the best choice. As shown in Table 5, we first fix $T = 1$ and then change λ from 0.1 to 1.0 with step 0.1. We find that the performance on the training set is improved continuously and the performance on the validation set starts to get worse with a large gap away from that of the training set when λ is beyond 0.7. Then, we fix $\lambda = 0.7$ and change temperature T from 1 to 10 with step 1. We find that when T gets larger, the AUC metric gets worse, since the larger T can soften the distribution and leads to the narrowing of the difference between the two classification results and the unclear boundary. Thus, we choose $\lambda = 0.7$ and $T = 1$ in our experiments.

4.5 Further Analysis

Representation Visualization. To check the effectiveness of the learned spatiotemporal context features in identifying fake artifacts, we visualize the feature maps from our detector's intermediate-layers [29, 52]. In our experiment, the encode $f(\cdot)$ is constructed by CNNs, we put the final convolutional layer's outputs across two pooling layers to get our demand feature map. Meanwhile, our encoder is basically constructed by 3D-convolutional layers. Then, the output feature map from $f(\cdot)$ is denoted by a 4D tensor. So we randomly pick a 2D feature map from the output of the last convolutional layer of the encoder $f(\cdot)$, as shown in the last column of Figure 9. Then, we draw the artifact regions (the light parts) in gray scale image and a sequence of testing faces and its output feature map are next to it. The higher feature response values can be observed in the feature maps' brighter regions, corresponding with artifacts in the fake face videos during the frame change. From Figure 9, we can find that the abnormal temporal changes generally fall in the bright regions in the feature maps, and the feature maps effectively highlight the fake regions, demonstrating the effectiveness of our predictive representation learning approach. Also compared with the results from pretrained FaceNet in Figure 1, it is obvious that our model extracts more prominent feature maps with less noise. Such a phenomenon proves that the attention of our model is more focused, and the extracted features are more discriminative and can provide strong generalization ability.

We further use t-SNE [49] to visualize the representations. Figure 10 presents the visualization of spatial semantic features and spatiotemporal context features, showing these two features have obvious discriminability between fake and real ones. It also shows that the spatiotemporal context features have smaller intra-class distances with fewer outliers, which may benefit from the learned spatiotemporal context features that can capture more subtle temporal artifacts.

Efficiency Analysis. We evaluate the time cost of all modules on 100 videos composed of 180 sampled face frames on a 2.4-GHz CPU. We use the Titan Xp GPU for data loading and inference with a memory of 12 GB. The average time costs of each video are 31.7, 114.1, 24.4, and 14.7 ms for the localizer, encoder, aggregator, and classifier, respectively. Thus, it is feasible to deploy the detector in practical scenarios. Besides, the time cost of pretraining is higher, because of the

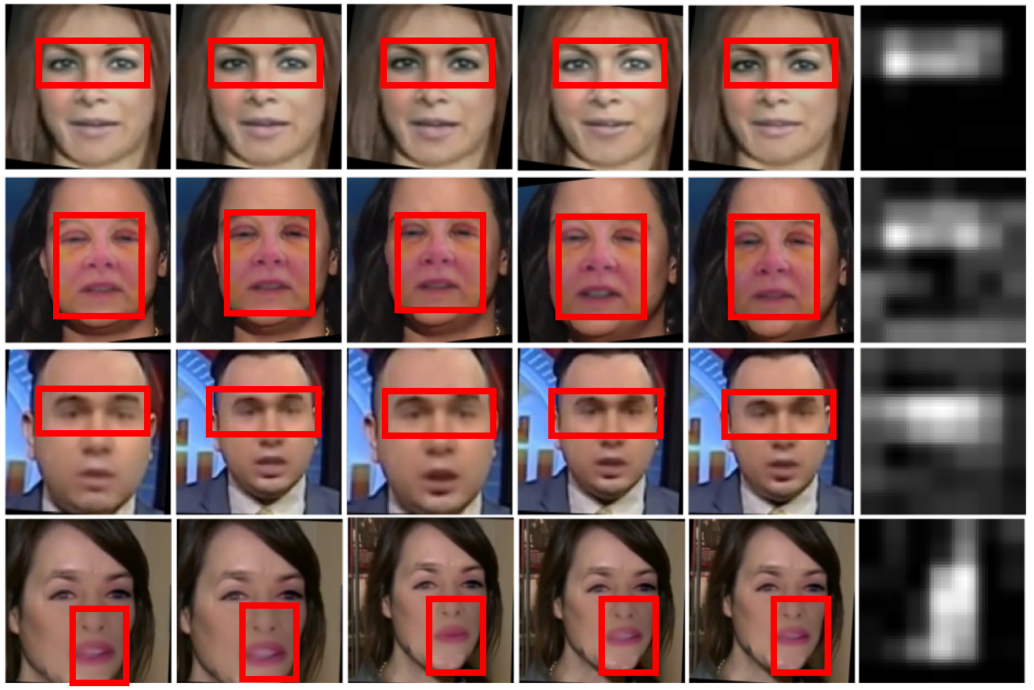

Fig. 9. Some visualization examples of fake video frames and their corresponding feature maps (last column). It shows that the fake regions (marked in red rectangles) are effectively identified from the feature maps. The first three rows use the models trained on Face-Forensics++, DFDC, and Celeb-DF(v2), respectively. The last row uses the model trained on mixed datasets.

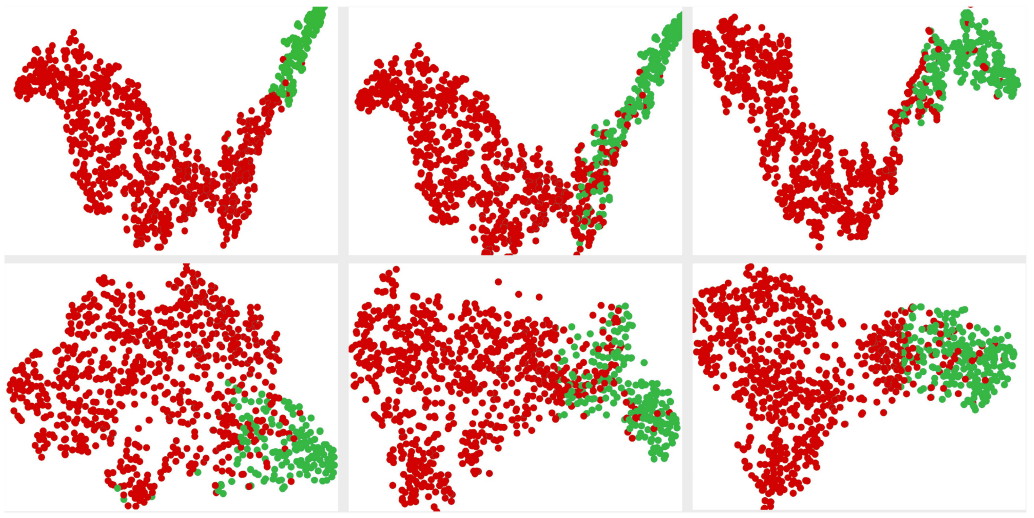

Fig. 10. Representation visualization of spatial semantic features (top) and spatiotemporal context features (bottom) on FaceForensics++, DFDC and Celeb-DF(v2) from left to right, respectively. Red: Fake, Green: Real.

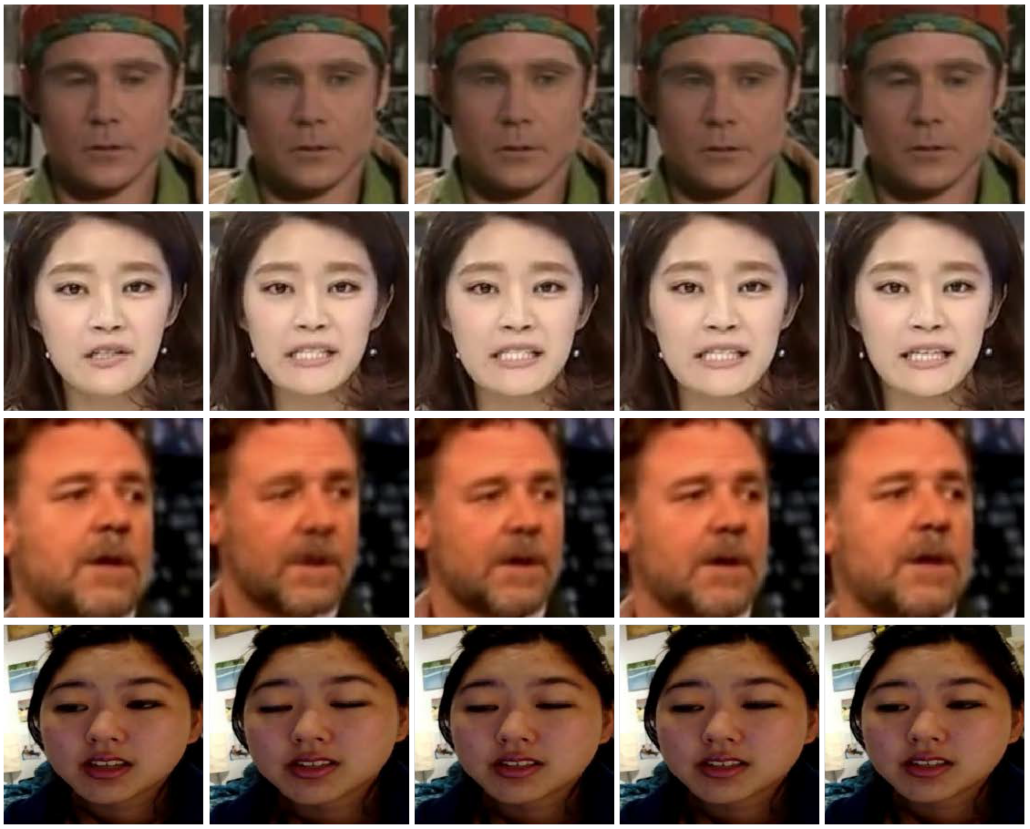

Fig. 11. Some detection failure cases. The first two rows are fake videos but classified as real ones, and the last two rows are real videos but misclassified.

predictor module. We also evaluate the inference efficiency of the predictor by averaging the time cost on 100 videos. The average time cost is 21.7 ms in inferring a video and 13.9 ms in updating the memory bank for a video.

Failure Analysis. Figure 11 demonstrates some detection failure cases, from which we can get several meaningful findings. First, we can see that high photo-realistic synthesis across multiple frames typically leads to great difficulty in identifying fake traces from short-term temporal clues, which often results in detection failure. It is expected to develop more discriminative video representations to capture context clues on a longer-term temporal scale. Second, the real and fake videos often are very confused especially when their image quality is poor. In this case, the binary annotation of video examples with real or fake usually cannot provide sufficient prior information for model learning, leading to the poor ability of a simple classifier. Maybe using a strong classifier can be a probable solution. Thus, our future work is to develop a more discriminative encoder also a more effective solution to describe and identify the fake traces on a fine-grained and long-term level.

5 CONCLUSION

In this work, we found that the deepfake videos' spatiotemporal inconsistencies can be revealed with more distinguished features at the semantic level. Following this finding, we propose an LPS

model to learn such artifact patterns and verify their authenticity. We use the predictive learning mechanism and selective self-distillation fine-tuning method to boost the model's detection performance and robustness. The extensive experiments and comprehensive analysis show the effectiveness of our approach. In the future, we will explore the potential of our predictive representation learning approach to improve the detector by identifying deepfake artifacts at multi-grained levels.

REFERENCES

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. 2018. Mesonet: A compact facial video forgery detection network. In *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS'18)*. 1–7.

[2] Shruti Agarwal, Hany Farid, Ohad Fried, and Maneesh Agrawala. 2020. Detecting deep-fake videos from phoneme-viseme mismatches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'20)*. 2814–2822.

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*. 5297–5307.

[4] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. 2016. Delving deeper into convolutional networks for learning video representations. In *Proceedings of the International Conference on Learning Representations (ICLR'16)*.

[5] Qian Bao, Wu Liu, Yuhao Cheng, Boyan Zhou, and Tao Mei. 2021. Pose-guided tracking-by-detection: Robust multi-person pose tracking. *IEEE Trans. Multimedia* 23 (2021), 161–175.

[6] Christoph Bregler, Michele Covell, and Malcolm Slaney. 1997. Video rewrite: Driving visual speech with audio. In *Proceedings of the ACM Proceedings of Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'97)*. 353–360.

[7] Michael Buckland and Fredric Gey. 1994. The relationship between recall and precision. *J. Am. Soc. Inf. Sci.* 45, 1 (1994), 12–19.

[8] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? A new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 6299–6308.

[9] Kevin Dale, Kalyan Sunkavalli, Micah K. Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. 2011. Video face replacement. *ACM Trans. Graph.* 30, 6 (2011), 130.

[10] Sowmen Das, Selim Seferbekov, Arup Datta, Md. Saiful Islam, and Md. Ruhul Amin. 2021. Towards solving the deepfake problem: An analysis on improving deepFake detection using dynamic face augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW'21)*. 3769–3778.

[11] Emily Denton and Vighnesh Birodkar. 2017. Unsupervised learning of disentangled representations from video. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS'17)*. 4417–4426.

[12] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton-Ferrer. 2019. The deepfake detection challenge (dfdc) preview dataset. arXiv:1910.08854. Retrieved from https://arxiv.org/abs/1910.08854.

[13] Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. 2020. Stochastic latent residual video prediction. In *Proceedings of the International Conference on Machine Learning (ICML'20)*. 3233–3246.

[14] Shiming Ge, Fanzhao Lin, Chenyu Li, Daichi Zhang, Jiyong Tan, Weiping Wang, and Dan Zeng. 2021. Latent pattern sensing: Deepfake video detection via predictive representation learning. In *Proceedings of the ACM Multimedia Asia (MMAsia'21)*. 6:1–6:7.

[15] Shiming Ge, Shengwei Zhao, Chenyu Li, and Jia Li. 2019. Low-resolution face recognition in the wild via selective knowledge distillation. *IEEE Trans. Image Process.* 28, 4 (2019), 2051–2062.

[16] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS'14)*. 2672–2680.

[17] David Güera and Edward J. Delp. 2018. Deepfake video detection using recurrent neural networks. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'18)*. 1–6.

[18] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. 2021. Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 5039–5049.

[19] Tengda Han, Weidi Xie, and Andrew Zisserman. 2019. Video representation learning by dense predictive coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW'19)*. 1–10.

[20] Tengda Han, Weidi Xie, and Andrew Zisserman. 2020. Memory-augmented dense predictive coding for video representation learning. In *Proceedings of the European Conference Computer Vision (ECCV'20)*. 312–329.

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*.
[22] Javier Hernandez-Ortega, Rubén Tolosana, Julian Fiérrez, and Aythami Morales. 2021. Deepfakeson-phys: Deepfakes detection based on heart rate estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence Workshop on Artificial Intelligence Safety (AAAIW'21)*. 1–8.
[23] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS'15) Workshop*. 1–9.
[24] Sohail Ahmed Khan, Alessandro Artusi, and Hang Dai. 2021. Adversarially robust deepfake media detection using fused convolutional neural network predictions. arXiv:2102.05950. Retrieved from https://arxiv.org/abs/2102.05950.
[25] Minha Kim, Shahroz Tariq, and Simon S. Woo. 2021. FReTAL: Generalizing deepfake detection using knowledge distillation and representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 1001–1012.
[26] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR'15)*.
[27] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. 2020. Advancing high fidelity identity swapping for forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*. 5073–5082.
[28] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. 2019. Celeb-DF (v2): A new dataset for deepfake forensics. arXiv:1909.12962. Retrieved from https://arxiv.org/abs/1909.12962.
[29] Ruofan Liang, Tianlin Li, Longfei Li, Jing Wang, and Quanshi Zhang. 2020. Knowledge consistency between neural networks and beyond. In *Proceedings of the International Conference on Learning Representations (ICLR'20)*.
[30] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. 2021. Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 772–781.
[31] Wu Liu, Qian Bao, Yu Sun, and Tao Mei. 2022. Recent advances in monocular 2d and 3d human pose estimation: A deep learning perspective. *Comput. Surv.* (2022). https://doi.org/10.1145/3524497
[32] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. 2021. Generalizing face forgery detection with high-frequency features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 16317–16326.
[33] Falko Matern, Christian Riess, and Marc Stamminger. 2019. Exploiting visual artifacts to expose deepfakes and face manipulations. In *Proceedings of the IEEE Winter Applications of Computer Vision Workshops (WACVW'19)*. 83–92.
[34] Yisroel Mirsky and Wenke Lee. 2022. The creation and detection of deepfakes: A survey. *Comput. Surv.* 54, 1 (2022), 7:1–7:41.
[35] Trisha Mittal, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha. 2020. Emotions don't lie: An audio-visual deepfake detection method using affective cues. In *Proceedings of the ACM International Conference on Multimedia (MM'20)*. 2823–2832.
[36] Huy H. Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. 2019. Multi-task learning for detecting and segmenting manipulated facial images and videos. In *Proceedings of the IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS'19)*. 1–8.
[37] Yuval Nirkin, Lior Wolf, Yosi Keller, and Tal Hassner. 2021. Deepfake detection based on discrepancies between faces and their context. *IEEE Trans. Pattern Anal. Mach. Intell.* (2021), 1–1. https://doi.org/10.1109/TPAMI.2021.3093446
[38] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. 2015. Deep face recognition. In *Proceedings of the British Machine Vision Conference*. 41.1–41.12.
[39] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. 2019. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV'19)*. 1–11.
[40] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. 2019. Recurrent convolutional strategies for face manipulation detection in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'19)*. 80–87.
[41] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*. 815–823.
[42] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. 2019. Learning video representations using contrastive bidirectional transformer. arXiv:1906.05743. Retrieved from https://arxiv.org/abs/1906.05743.
[43] Zekun Sun, Yujie Han, Zeyu Hua, Na Ruan, and Weijia Jia. 2021. Improving the efficiency and robustness of deepfakes detection through precise geometric features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 3609–3618.

[44] Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph.* 38, 4 (2019), 66:1–66:12.

[45] Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. 2015. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.* 34, 6 (2015), 183:1–183:14.

[46] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR'16)*. 2387–2395.

[47] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. 2020. Deepfakes and beyond: A survey of face manipulation and fake detection. *Inf. Fus.* 64 (2020), 131–148.

[48] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. 2018. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR'18)*. 1526–1535.

[49] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, 86 (2008), 2579–2605.

[50] Jun Wang, Yinglu Liu, Yibo Hu, Hailin Shi, and Tao Mei. 2021. FaceX-Zoo: A pyTorch toolbox for face recognition. In *Proceedings of the ACM International Conference on Multimedia (MM'21)*. 3779–3782. arXiv:2101.04407. Retrieved from https://arxiv.org/abs/2101.04407.

[51] Yaohui Wang and Antitza Dantcheva. 2020. A video is worth more than 1000 lies. Comparing 3DCNN approaches for detecting deepfakes. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG'20)*. 515–519.

[52] Natalie Wolchover and Lucy Reading. 2017. New theory cracks open the black box of deep learning. *Quanta Mag.* 3 (2017).

[53] Chin-Yuan Yeh, Hsi-Wen Chen, Shang-Lun Tsai, and Sheng-De Wang. 2020. Disrupting image-translation-based DeepFake algorithms with adversarial attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACV'20)*. 53–62.

[54] Daichi Zhang, Chenyu Li, Fanzhao Lin, Dan Zeng, and Shiming Ge. 2021. Detecting deepfake videos with temporal dropout 3DCNN. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'21)*. 565–573.

[55] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. 2021. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 2185–2194.

[56] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. 2021. Exploring temporal coherence for more general video face forgery detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV'21)*. 15044–15054.

[57] Xiangyu Zhu, Hao Wang, Hongyan Fei, Zhen Lei, and Stan Z. Li. 2021. Face forgery detection by 3d decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 2929–2939.

[58] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. 2020. Wilddeepfake: A challenging real-world dataset for deepfake detection. In *Proceedings of the ACM International Conference on Multimedia (MM'20)*. 2382–2390.

Received February 2022; revised April 2022; accepted May 2022