TI^{2}Net: Temporal Identity Inconsistency Network for Deepfake Detection

Baoping Liu[1], Bo Liu[1], Ming Ding[2], Tianqing Zhu[1], Xin Yu[1]
[1]University of Technology Sydney, NSW, Australia
[2] Data61, CSIRO, Sydney, NSW, Australia
baoping.liu@student.uts.edu.au, ming.ding@data61.csiro.au
{Bo.liu, Tianqing.Zhu, Xin.Yu}@uts.edu.au

Abstract

*In this paper, we propose the **T**emporal **I**dentity **I**nconsistency **Net**work (TI^{2}Net), a Deepfake detector that focuses on temporal identity inconsistency. Specifically, TI^{2}Net recognizes fake videos by capturing the dissimilarities of human faces among video frames of the same identity. Therefore, TI^{2}Net is a reference-agnostic detector and can be used on unseen datasets. For a video clip of a given identity, identity information in all frames will first be encoded to identity vectors. TI^{2}Net learns the temporal identity embedding from the temporal difference of the identity vectors. The temporal embedding, representing the identity inconsistency in the video clip, is finally used to determine the authenticity of the video clip. During training, TI^{2}Net incorporates triplet loss to learn more discriminative temporal embeddings. We conduct comprehensive experiments to evaluate the performance of the proposed TI^{2}Net. Experimental results indicate that TI^{2}Net generalizes well to unseen manipulations and datasets with unseen identities. Besides, TI^{2}Net also shows robust performance against compression and additive noise.*

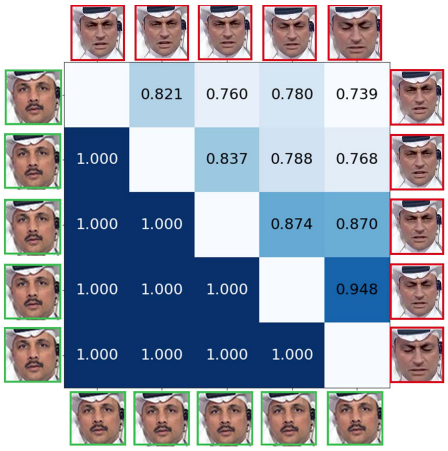

Figure 1: Pair-wise identity similarities between real video frames (lower triangle) and fake video frames (upper triangle). Each value in the lower triangle represents the similarity between the face at the left of the row and the face at the bottom of the column (in green boxes). Each value in the upper triangle represents the similarity between the face at the right of the row and the face at the top of the column (in red boxes). The diagonal represents self-similarity and is left blank.

1. Introduction

Recent developments in Deep Neural Networks (DNNs), especially Generative Neural Networks (GANs), have enabled Deepfake to generate realistic images and videos and confuse the public. Therefore, Deepfake detection has become an urgent topic to protect people from misinformation caused by Deepfake.

Existing Deepfake detectors can be categorized into image-level (frame-level) detectors and video-level detectors. Image-level detectors rely on spatial artifacts such as texture inconsistency [26], [5] and color distortion [12], [11]. Some detectors also transform spatial information to the frequency domain and capture artifacts in frequency domain [8], [9], [14]. Furthermore, with the help of powerful DNNs, many detectors analyze artifacts in the latent space to identify fake images [23], [25], [24]. Especially, as face swapping operation blends faces of different identities, identity inconsistency artifacts have recently been found to be an effective clue for fake detection. For example, Dong et al. [5] [6] detect the **spatial identity inconsistencies** between the inner face region and the outer face region. Although such identity-based detectors can be used to detect video frame-by-frame, they do not perform well on fake videos, since they do not account for temporal inconsistencies, which are more intuitive and important artifacts to detect Deepfake videos.

In order to detect fake videos, some detectors have utilized temporal artifacts such as inter-frame variation [7], temporal frequency artifacts [15] and general inconsisten-

cies among frames [27], [22]. In particular, the works in [2] and [3] studied the temporal inconsistency of identities in fake videos and proposed detectors to discriminate fake videos from real ones. However, the proposed models are only for **close-set** scenes, which means they require reference sets of identities to provide candidate identities. Therefore, their methods do not generalize well and perform poorly in **open-set** scenes, where detectors are used to find out Deepfake of unseen identities.

In this paper, we propose a new Deepfake detection framework called Temporal Identity Inconsistency Network (TI^2Net). The key idea is to detect temporal identity inconsistencies in suspect videos, i.e., the low similarity of identity features captured from the same video with the given identity. Therefore, An intuitive solution would be to measure the similarities between identity features captured from the frames containing the same identity. Fig. 1 is an example of temporal identity inconsistency measured by the similarity of identity features in frames of the same video. In detail, we randomly select one real video and one fake video from FaceForensics++ (FF++) dataset [18] and randomly sample five frames from each video. For selected frames, we extract the identity vectors of faces by arcface [4]. We then calculate the pair-wise similarities of identity vectors between frames in each video. The results are shown in the matrix in Fig. 1. It can be observed that although the fake frames look visually realistic, the inter-frame similarity values are significantly lower than those of the real frames, which indicates that temporal inconsistency can be exploited to expose Deepfake. Unlike spatial inconsistency-based detectors that need identity labels for inner face identities and outer face identities during generation, temporal identity inconsistency is based on the dissimilarity of the same identity among video frames and does not request external information about identity. Besides, detectors based on temporal identity inconsistency do not need a reference set to provide candidate identities.

Based on the above discussion, our proposed TI^2Net captures temporal identity inconsistency to detect fake videos. Real and fake sequences are first generated from video sets during pre-processing. Identity vectors are then extracted by an identity encoder. TI^2Net captures identity inconsistency among frames rather than the identity itself. Thus, we calculate the difference between successive frames to generate the temporal difference of identity vectors. RNNs are then adopted to learn the temporal embedding of the temporal difference, and the information of identity inconsistency is represented by the temporal embedding. The embedding is applied to construct triplet loss to optimize temporal modeling. Besides, the temporal embedding is also fed to the classification head for binary classification.

The contributions of our work are as follows:

- We propose **T**emporal **I**dentity **I**nconsistency **Net**work (TI^2Net), a new Deepfake video detection framework based on temporal identity inconsistencies. TI^2Net calculates temporal difference to capture inconsistency among frames and avoid over-fitting to identities. Thus, TI^2Net does not need a reference set with candidate identities so that TI^2Net can work in open-set scenes.

- TI^2Net adopts RNNs to learn embeddings of temporal identity inconsistency and incorporate triplet loss to optimize the extraction of temporal embeddings.

- We conduct comprehensive experiments to evaluate the performance of the proposed framework. The results demonstrate that our framework makes an improvement in terms of cross-manipulation generalization, cross-dataset generalization, and robustness against image degradation.

2. Related Work

2.1. Deepfake Generation

Deepfake generation task has branched into two main categories: face swapping and face reenactment. Face swapping [28] [17] manipulation aims to replace the face of the source identity in an image with the face of another identity. Face swapping manipulation usually includes the combination of two faces, color and texture transfer and processing of edges of combination. However, imperfect generations easily leave artifacts like visible edges or inconsistency of color and texture patterns between the source face and target face.

Manipulations like face reenactment and facial attribute manipulation [21] [20] apply alteration to facial parts to customize human faces. Different from face swapping that harms the identity feature of the source, face reenactment only applies manipulations to facial attributes, thus the identity features could be preserved to ensure the face can be recognized as the same person before and after the manipulation.

2.2. Identity-based Deepfake Detection

The work in [2] is based on the idea that the biometric measures, such as appearance and behavior patterns, of the same identity should be consistent. Thus, they proposed a two-branch framework, where one branch extracts appearance features, and the other branch captures behavior features. Then the cosine similarity between the two-branch features is compared with the reference set to select the best matching reference videos. If the identities of the two best matching videos are consistent, the video is considered authentic.

LRNet[19] represents identity with facial landmarks and applies such identity features to detect Deepfakes. In detail, LRNet mines temporal identity features of videos to tell Deepfake videos from real ones. To extract more accurate and precise features, LRNet adopts a calibration module to fine-tune the extracted landmarks, and the fine-tuned landmarks are then fed to a two-branch RNN to analyze temporal patterns of identity landmarks. Though LRNet provides solid pre-processing and achieves promising detection performance, it performs poor generalization ability.

ID-reveal [3] captures temporal identity feature by the adoption of 3DMM model. Then a temporal ID network analyzes the temporal pattern of 3DMM models. The model is trained in an adversarial manner leveraging a set of reference videos.

Identity Inconsistency Transformer (ICT) [5] is based on the inconsistency of the inner face region and outer face region in face-swapped images. Specially, ICT adopts a powerful Transformer to capture the inconsistent information from patch sequences of images, the outputs of the inner token and outer token of the last block are regarded as the identity information. ICT is designed for face-swapped images and requests a reference video set during evaluation.

3. Temporal Identity Inconsistency Network

We propose the Temporal Identity Inconsistency Network TI^2Net (as illustrated in Fig. 2) to detect Deepfake videos by capturing the temporal inconsistency of the same identities in a video.

Given an input video clip or a series of frames with the same identity, TI^2Net first extracts identity vectors from all frames with an identity encoder. Then the temporal difference of identity vectors is generated by differencing operation to transform identity features into temporal inconsistency among video frames. An RNN is then adopted to extract the temporal embedding of the identity inconsistency. The embedding was finally fed to the classification head to predict the sequence into the right class. During training, a triplet loss is adopted to benefit the temporal modeling. Firstly, the anchor sequence and the positive sequence are sampled from real videos, and a negative sequence is sampled from fake videos. These three sequences all go through the above-mentioned processing, and then we get the anchor embedding, positive embedding and negative embedding. The distance between anchor embedding and positive embedding (**anchor-positive distance**) indicates the similarity between the same class sequences, while the distance between anchor embedding and negative embedding (**anchor-negative distance**) indicates the similarity between the sequences of different classes. Thus, the triplet loss can be constructed to optimize the temporal modeling by minimizing the anchor-positive distance and maximizing the anchor-negative distance.

3.1. Pre-processing and Differencing

Pre-processing: Pre-processing includes frame extraction, face cropping and identity encoding. Sequences of identical lengths l are generated from videos firstly. All sequences are categorized into the real set and fake set according to the labels of their original videos. For the construction of triplet loss, the anchor sample and the positive sample are sampled from the real set, the negative sample is sampled from the fake set.

The identity features in a sample sequence can be encoded by the identity encoder. Therefore, the anchor sample, positive sample and negative sample are then transformed into anchor identity vectors I_a, positive identity vectors I_p and negative identity vectors I_n respectively. The identity encoder is pretrained and is not updated during joint training, so that the whole framework will focus on temporal information extraction.

Differencing: To make our model focus on identity inconsistency among frames rather than identities themselves, we first apply differencing to identity vectors and get the **temporal difference** of the identity vectors. Differencing aims to better explore the temporal patterns in identity vectors by calculating the difference between consecutive frames.

For identity vectors I_s:

$$I_s = \{i_{s,1}, i_{s,2}, ..., i_{s,l}\}. \tag{1}$$

where $i_{sk} \in \mathcal{R}^{\mathcal{D}}$ is the k^{th} identity vector, k $\in [1, l]$, l is the sequence length of identity vectors.

The corresponding temporal difference D_s is:

$$\begin{aligned} D_s &= \{d_{s,1}, d_{s,2}, ..., d_{s,l-1}\} \\ &= \{i_{s,2} - i_{s,1}, i_{s,3} - i_{s,2}, ..., i_{s,l} - i_{s,(l-1)}\}, \end{aligned} \tag{2}$$

Different from identity vectors that indicate identity information within frames, the temporal difference emphasizes the features among video frames, especially the identity inconsistency among frames. Therefore, The temporal difference sequences of I_a, I_p and I_n are D_a, D_p and D_n.

3.2. Temporal Identity Inconsistency Learning

Temporal difference contains rich information on among-frame inconsistencies, especially temporal identity inconsistency. We then adopt RNNs to learn the temporal embedding of identity inconsistencies from the temporal difference. For the given temporal difference D_s, RNN, i.e., Gated Recurrent Unit (GRU) in our framework, processes D_s in a sequential manner. As shown in Fig. 3, at time step t, RNN processes component $d_{s,t}$, the t^{th} component of D_s. The memory of the GRU is controlled by the reset gate and the update gate. Both gates are based on the hidden status of last time step h_{t-1} and current input $d_{s,t}$:

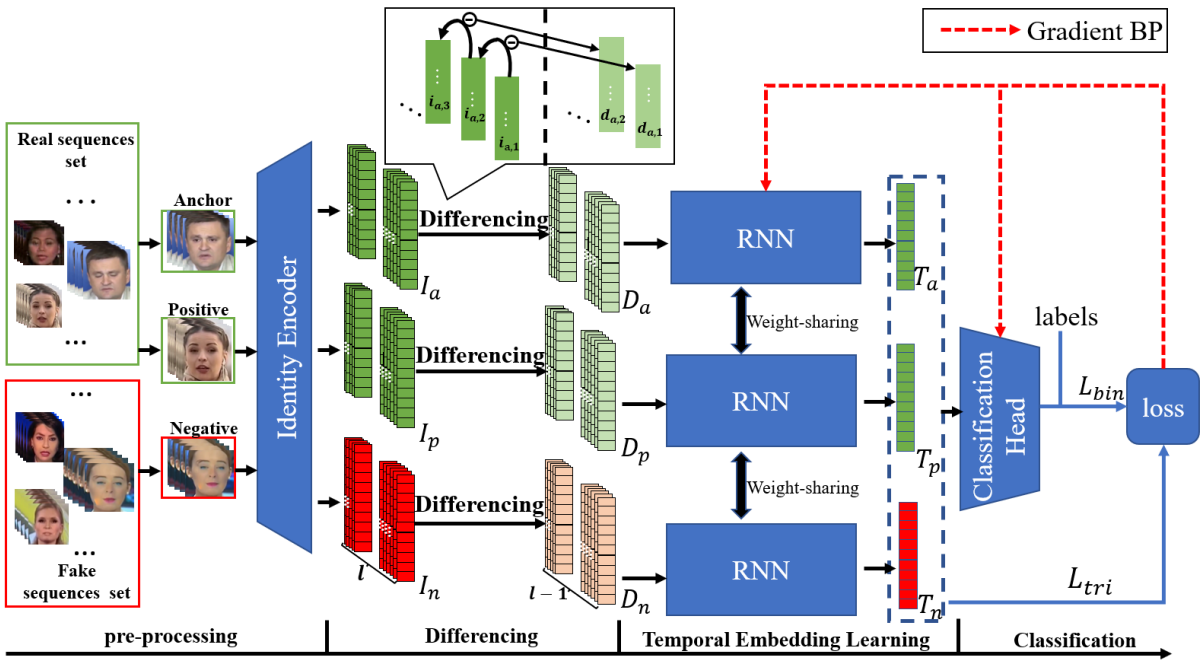

Figure 2: Framework of the proposed Temporal Identity Inconsistent Network (TI^2Net). The raw sequence of video frames is transformed into identity vectors through the identity encoder. Differencing operation is then adopted to generate the temporal difference, which better captures the temporal inconsistency of the identity. From the temporal difference, RNN then learns the temporal embedding, which represents temporal identity inconsistency. Temporal embeddings of anchor, positive and negative samples are used to construct triplet loss to benefit the temporal embedding learning and binary classification.

$$z_t = \sigma(\mathcal{W}_z \cdot [h_{t-1}, d_{s,t}] + b_z), \tag{3}$$
$$r_t = \sigma(\mathcal{W}_r \cdot [h_{t-1}, d_{s,t}] + b_r), \tag{4}$$

where z_t and r_t are update gate and reset gate respectively. W_z and W_r are weight matrix of update gate and reset gate, b_z and b_r are corresponding bias. σ is the Sigmoid activation function.

Then r_t updates the h_{t-1}:

$$\hat{h}_{t-1} = tanh(\mathcal{W}_h \cdot [r_t \odot h_{t-1}, x_t] + b_n), \tag{5}$$

where $tanh$ is the tanh activation function and $\odot$ is the point-wise multiply operation.

Then the hidden status gets updated as:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_{t-1}. \tag{6}$$

The **temporal embedding** T_s is the output of the last time step. Therefore, for temporal difference D_a, D_p and D_n, RNN learns embeddings T_a, T_p and T_n that represents the temporal identity inconsistency of anchor, positive and negative samples, respectively.

Triplet loss: To make the RNNs learn more discriminative temporal embeddings, we adopt a triplet loss that measures the distance between temporal embeddings. As the positive sample is from the same set as the anchor sample (real sequence set) and the negative sample is from the fake sequence set, T_a and T_p should be more similar than T_a and T_n. Therefore, we adopt a triplet loss $L_{tri}(T(a), T(p), T(n))$ to minimize the anchor-positive distance and maximize the anchor-negative distance. In detail, the L2-norm is adopted to measure the distance between embeddings. The triplet loss of temporal represen-

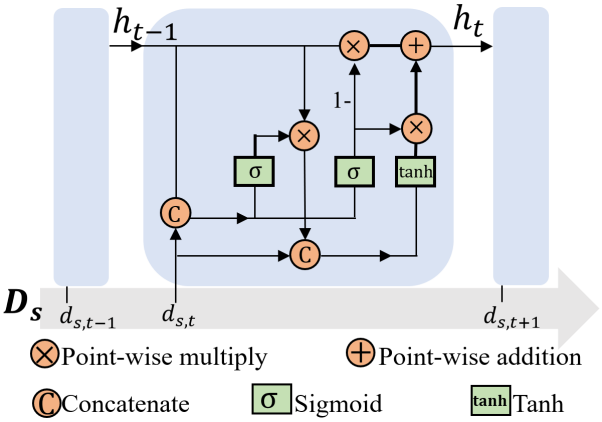

Figure 3: Temporal embedding leaning with RNN.

tation T_a, T_p and T_n is:

$$L_{tri}(T(a), T(p), T(n)) = \max(||T(a) - T(p)||^2 - ||T(a) - T(n)||^2 + \alpha, 0), \quad (7)$$

where $||\cdot||^2$ is the L2-norm and α_1 is the triplet loss margin.

To help the convergence of triplet loss by avoiding the situation in which both anchor-positive distance and anchor-negative distance increase, we also adopt anchor-positive distance as a regularization item:

$$L_{ap} = ||T(a) - T(p)||^2. \quad (8)$$

3.3. Fake Video Classification

The temporal embeddings T_a, T_p and T_n are also fed to classification head for the prediction of binary labels. The binary classification loss is as follows:

$$L_{clf}(y_i, p(y_i)) = CE(y_i, p(y_i)) - \frac{1}{N}\sum_{i=1}^{N}(y_i \cdot log(p(y_i))) + (1 - y_i) \cdot log(1 - p(y_i)), \quad (9)$$

where CE(,) is cross-entropy loss, y_i is the label of a sample i and p(y_i) is the possibility of the sample i being classified as the positive class.

The overall loss of TI^2Net is:

$$L = L_{clf} + \lambda_1 L_{tri} + \lambda_2 L_{ap}, \quad (10)$$

where L_{clf} is the binary classification loss, L_{tri} is the triplet loss and L_{ap} is the regularization item. Besides, λ_1 and λ_2 are hyperparameters to control the importance of loss items.

4. Experiments

4.1. Experiment Settings

Datasets: In our experiments, we train our model with FaceForensics++ (FF++) dataset [18], which contains 1000 real videos and 4000 fake videos. Specially, the fake dataset consists of 4 subsets corresponding to four Deepfake manipulations, two of which (Deepfakes and FaceSwap) are for the face-swapping task and the other two (Face2Face and NeuralTextures) are for of face reenactment task.

We also use benchmark datasets to test the performance of the proposed methods:

(1) DeepFake Detection (DFD) [16]: A dataset released by google, containing hundreds of real videos of some paid actors and thousands of fake videos generated with the identities of the real videos.

(2) DeeperForensics-1.0 (Deeper) [10] : A large-scale forgery detection dataset. The adoption of the more recent generation makes videos of high quality.

(3) Celeb-DeepFake v1 (CDF1) [13]: A Deepfake detection dataset that contains 408 real videos and 795 Deepfake videos.

(4) Celeb-DFv2 (CDF2) [13]: An extended version of CelebDF1, containing 590 real videos and 5639 fake videos.

Implementation details: Real and fake sequences are generated by randomly sampling 64 frames (sequence length l=64) from original videos. We generate 20 different sequences from each video to form the fake sequence set and real sequence set. The identity encoder is arcface [4] pretrained using ResNet-18 without se. During encoding identities with arcface, we adopt flipping and concatenation operations to improve encoding performance. Each sequence sample is transformed to identity vectors of dimension $\mathcal{D}$=1024. More details of training implementation can be seen in supplementary materials.

Baselines: We compare our TI^2Net with the following open-set baselines:

(1) **MesoNet** [1]: A Deepfake detector based on mesoscopic properties of images.

(2) **Xception** [18]: A Deepfake detector with Xception. According to the different datasets used for training, we compared our method with **Xception-c0**, which is trained on FF++ raw set and **Xception-c23**, which is trained on FF++-c23 set.

(3) **LRNet** [19]: A Deepfake detector that processes landmarks sequences with RNN.

Besides, we also compare our method with two close-set baselines:

(4) **A&B** [2]: A Deepfake detector that integrates behaviour and appearance of identities.

(5) **ICT** [5]: A Deepfake detector based on spatial identity inconsistency.

Evaluation Metrics: We use classification accuracy (ACC) and area under the Receiver Operating Characteristic curve (AUC) for evaluation.

4.2. Comparison with State-of-the-art Works

To compare the predictive performance and the generalization ability of TI^2Net with other baselines, We conduct **in-set evaluation** (train and test on FF++) and **cross-set evaluation** (train on FF++ and test on other datasets). We train our model on the FF++ dataset and test the trained models on other datasets. For fair comparisons, baselines MesoNet, Xception-c0, Xception-c23 and LRNet are also trained and tested in the same manner, i.e., train on FF++ and test on both FF++ for in-set evaluation and other datasets for cross-set evaluation. The results are shown in Table 1. We mark the best results on each dataset. As for close-set baselines A&B and ICT, they detect fake videos with reference sets, which makes detection much easier. Therefore, it is hard to make a fair comparison to TI^2Net

Table 1: Comparison of TI^2Net with state-of-art works in terms of video-level AUC (%). In-set evaluation refers to training and testing models on FF++. Cross-set refers to training models on FF++ and testing on unseen datasets.

Methods	in-set	cross-set				
	FF++	DFD	Deeper	CDF1	CDF2	Avg
MesoNet [1]	99.91	56.55	52.36	54.69	53.97	54.39
Xception-c0 [18]	99.94	61.23	60.03	58.82	58.23	59.58
Xception-c23 [18]	99.92	63.33	62.58	60.03	62.16	62.0
LRNet [19]	99.89	52.29	56.77	52.84	53.2	53.78
TI^2Net(Ours)	**99.95**	**72.03**	**76.08**	**66.65**	**68.22**	**70.75**

Table 2: Cross-manipulation evaluation results in terms of sequence-level AUC(%). The model is trained with one subset of FaceForensics++ and tested on the other three subsets of the datasets. The grey values are AUC trained and tested on the same manipulation. The bold values are the best cross-manipulation metrics.

Training set		Testing set			
		FS		**FR**	
		df	fs	ff	nt
FS	df	100.0	**89.45**	0.7003	85.82
	fs	**95.02**	100.0	74.23	81.27
FR	ff	90.73	90.64	100.0	**95.47**
	nt	**98.56**	93.07	98.45	99.99

*Note: the abbreviation **FS** represents the face swapping category including Deepfakes (df) and FaceSwap (fs). **FR** represents the face reenactment category including Face2Face (ff) and NeuralTextures (nt).

with close-set baselines. Thus the results are roughly compared, which can be found in our supplementary material.

Compared with open-set baselines, our TI^2Net achieves the best classification performance on all test datasets. On the Deeper dataset, our method achieves 76.08% AUC, higher than other datasets because the Deeper dataset is constructed based on the training set FF++ and some identities in the training set are also in the Deeper dataset. On most datasets, our method achieves over 70% AUC except CDF1 and CDF2, which are usually considered challenging. Our framework achieve close performance on CDF1 and CDF2 because the CDF2 is the extension of the CDF1 dataset.

4.3. Cross-manipulation Evaluation

To test the performance of our model on unseen manipulation, we train and test our model on the FF++ dataset. In detail, we train the model on one of four subsets (df for Deepfakes, fs for FaceSwap, ff for Face2Face and nt for NeuralTextures) and test the model on the other three subsets. The results are shown in Table 2.

Our model achieves almost perfect AUC when detecting videos from the same subset. Even on cross-manipulation settings, the minimum detection AUC is over 0.7 when trained on df subset and tested on ff subset. When trained on nt subset, our model achieves AUC as high as 0.9856 and 0.9845 on df and ff subsets, respectively.

In terms of FS and FR categories, it can be observed that when trained on data in the FR category, the model achieves AUC over 0.9 on FS category subsets, which is significantly higher than training on FS but testing on FR. Because the FR category subset contains modules to preserve the identity of source videos, FR category subsets are more challenging for identity-based detectors. When trained on more challenging datasets, the model is more likely to be generalized to subsets that contain obvious identity-related artifacts.

4.4. Ablation Study

Differencing: We first evaluate the effect of differencing operation by comparing the performance when the input to RNN are identity vectors and temporal difference, respectively. The results are shown in Table 3.

Compared with feeding RNN with I_s without differencing, feeding T_s slightly harms the performance of prediction on the seen dataset but significantly improves the generalization. Feeding I_s to RNN makes RNN learn the temporal information of identity vectors, thus, the whole framework is more likely to be over-fitted to identities, harming the generalization to unseen datasets. Temporal difference T_s contains more information about identity inconsistency among video frames, which is more general in different datasets.

Table 3: Ablation study of differencing operation in terms of AUC (%). Identity vectors are fed to RNNs **without** differencing when sequence type is I_s.

Sequence	FF++	DFD	Deeper	CDF1	CDF2
I_s	99.99	52.98	69.32	59.45	53.4
T_s	99.95	72.03	76.08	66.65	68.22

Table 4: Ablation study of loss items in terms of AUC (%). Full loss refers to loss function with items L_{clf}, L_{tri} and L_{ap}.

Loss	FF++	DFD	Deeper	CDF1	CDF2
L_{clf}	99.20	60.09	63.62	52.01	42.35
L_{clf}, L_{tri}	99.82	68.54	74.96	65.65	64.67
Full	99.95	72.03	76.08	66.65	68.22

With/without Triplet Loss: To evaluate the impact of items in the loss function of our framework, we train our model with different loss settings, the results can be seen in Table 4.

Comparing the results of L_{clf} setting and L_{clf}+L_{tri}, it is notable that L_{tri} significantly improves the generalization ability of the model to unseen datasets, since L_{tri} pull the temporal representations of real sequences closer and push the temporal representations further apart if two sequences are from different classes, the model with L_{tri} loss item learns more discriminative temporal embeddings of temporal identity inconsistency. Besides, L_{ap} improves the performance of the model on both seen and unseen datasets, which owes to the contribution of L_{ap} to the convergence of anchor-positive distance. We normalized the anchor-positive distance, negative distance and triple loss during training epochs and visualized them to indicate the contribution of L_{ap} in Fig. 4.

In Fig. 4 (a) it can be observed that without L_{ap}, although the triplet loss decreases until converges to 0, both anchor-positive and anchor-negative distances keep rising until they are no longer updated after triplet loss becomes converged, which means the anchor-positive and the anchor-negative distance will not be optimized after the convergence of triplet loss. While in Fig. 4(b), although triplet loss converged and anchor-negative is no longer optimized after that, anchor-positive distance keeps decreasing until the anchor-positive margin, which makes the temporal embeddings of real and fake more discriminative to improve the predictive performance.

Sequence length: We also test the performance of the model with different sequence lengths. The length candidates are selected from 16 to 128 with a step of 16. The results are shown in Table 5.

It can be observed that when sequence length is no more than 64, both AUC and Acc increase as sequence length increases, which indicates that short sequences may contain insufficient temporal patterns information for RNN to learn. Then when sequences are longer than 64, the performance drops gradually with the increase of length since long sequences lead to complex patterns to learn and also bring difficulties to model training convergence. Hence, a careful choice of the sequence length is vital to ensuring the effectiveness of the proposed method.

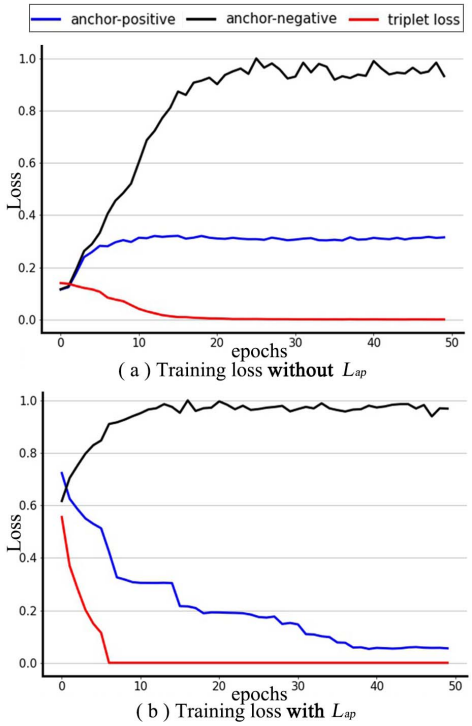

Figure 4: Anchor-positive distance, anchor-negative distance and triplet loss during training epochs.

Sampling Strategy: During the generation of sequences from videos, we evaluate two strategies and compare their performances. The first strategy is random sampling, which is to randomly select frames in temporal order to generate sequences. The second strategy is sliding window sampling, which is to apply a sliding window of sequence length and sample a short clip from videos. The performance under both sampling strategies is in Table 6.

While random sampling captures global identity inconsistency in a video, sliding window sampling is more focused on local identity inconsistency. Table 6 indicates that, compared with sliding window sampling, random sampling significantly outperforms sliding window sampling. This is because our model applies differencing before feeding sequences to RNNs. The coherence of videos makes consecutive video frames highly similar, making a large portion of temporal difference tensors zero and eliminating much information on temporal patterns.

4.5. Robustness Analysis

To evaluate the robustness of our methods, We challenge models with compressed images and images added with noise. Our model is compared with Xception and LRNet in robustness evaluation. We generate 20 groups and com-

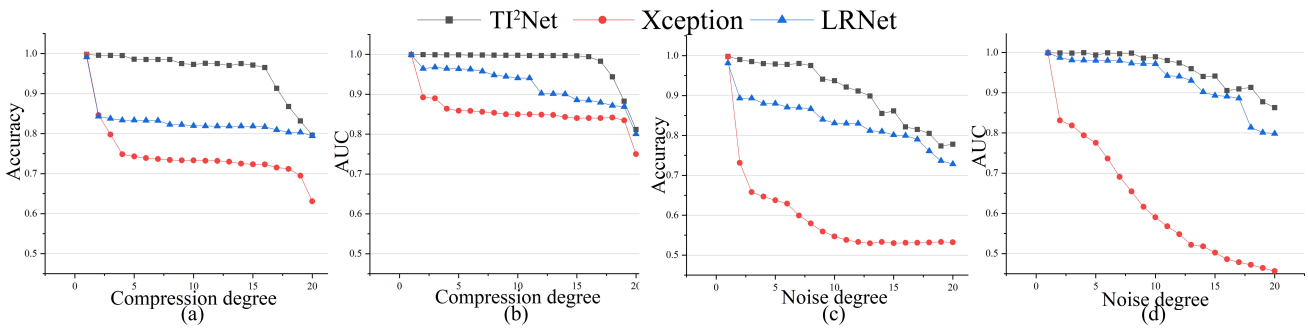

Figure 5: The evaluation of robustness against compression in terms of (a) Accuracy and (b) AUC; and the evaluation of robustness against additive noise in terms of (c) Accuracy and (d) AUC.

Table 5: Evaluation of models with different sequence lengths in terms of prediction Acc (%) and AUC (%).

Length	16	32	48	64	80	96	112	128
Acc	83.02	85.51	89.88	98.33	94.85	93.5.	90.01	88.00
AUC	89.97	98.35	98.53	99.77	99.65	98.99	97.59	97.15

Table 6: Evaluation of models with different sampling strategies to generate sequences in terms of AUC (%).

	FF++	DFD	Deeper	CDF1	CDF2
Random	99.95	72.03	76.08	66.65	68.22
Sliding	96.91	67.05	65.22	51.05	50.91

pressed samples and 20 groups of samples with noise according to the intensity (degree) of compression and noise respectively. For both compression and noise, a lower degree indicates lower degradation intensity. More details and samples can be found in supplementary materials. The results of degradation evaluation can be seen in Fig. 5.

As can be seen from Fig. 5(a) and Fig. 5(b), our model significantly outperforms the other two methods. In terms of both metrics, Xception suffers a large performance drop even when images are slightly compressed, and then it exhibits a stable performance until a high compression degree kicks in, causing the second quick decline of performance. LRNet, as a landmarks-based method, shows a similar performance trend as Xception, but its prediction performance is significantly higher than that of Xception at the relatively stable stage. Different from baselines, our TI^{2}Net shows high robustness against slight compression, i.e., the AUC hovers over 0.99 until the compression degree reaches 16. After the compression degree hits 17, the performance of TI^{2}Net in terms of both AUC and ACC starts to drop significantly. Nevertheless, it still remains better than the baselines.

In Fig. 5(c) and Fig. 5(d), we can see that the performance of Xception keeps decreasing as the noise degree grows. The classifier reaches a random-guessing stage when the noise degree climbs over 10. Both LRNet and TI^{2}Net show promising robustness against noise, especially in terms of AUC, they are both higher than 0.8 even when the noise degree is over 15. But in terms of ACC, TI^{2}Net performs better than LRNet, especially at the low and middle noise degrees.

5. Conclusion

In this work, we propose TI^{2}Net, a reference-agnostic Deepfake detector based on temporal identity inconsistency. We transform the identity vectors to temporal difference by differencing and learn temporal embeddings of identity inconsistency with RNN. Extensive experiments are conducted to evaluate the effectiveness of our framework. Our framework shows promising generalization ability to unseen datasets, especially to unseen manipulations. Our framework is also very robust to image compression and additive noise. We also notice that although we adopt differencing to avoid over-fitting to identities, our model performs better on seen identities, such as the high predictive performance in cross-manipulation evaluation and better cross-dataset performance on the Deeper dataset, which is constructed based on our training dataset. Therefore, we hope our work can inspire more future works on temporal identity inconsistency to make improvements.

Acknowledge. This research is funded in part by ARC-Linkage grant (LP180101150 to TZ and BL), ARC-Discovery grant (DP220100800 to XY) and ARC-DECRA grant (DE230100477 to XY). We thank all anonymous reviewers and ACs for their constructive suggestions.

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE international workshop on information forensics and security (WIFS)*, pages 1–7. IEEE, 2018.

[2] Shruti Agarwal, Hany Farid, Tarek El-Gaaly, and Ser-Nam Lim. Detecting deep-fake videos from appearance and behavior. In *2020 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2020.

[3] Davide Cozzolino, Andreas Rössler, Justus Thies, Matthias Nießner, and Luisa Verdoliva. Id-reveal: Identity-aware deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15108–15117, 2021.

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[5] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Ting Zhang, Weiming Zhang, Nenghai Yu, Dong Chen, Fang Wen, and Baining Guo. Protecting celebrities from deepfake with identity consistency transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9468–9478, 2022.

[6] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Dong Chen, Fang Wen, and Baining Guo. Identity-driven deepfake detection. *arXiv preprint arXiv:2012.03930*, 2020.

[7] Ziheng Hu, Hongtao Xie, Yuxin Wang, Jiahong Li, Zhongyuan Wang, and Yongdong Zhang. Dynamic inconsistency-aware deepfake video detection. In *IJCAI*, 2021.

[8] Yonghyun Jeong, Doyeon Kim, Seungjai Min, Seongho Joe, Youngjune Gwon, and Jongwon Choi. Bihpf: Bilateral high-pass filters for robust deepfake detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 48–57, 2022.

[9] Gengyun Jia, Meisong Zheng, Chuanrui Hu, Xin Ma, Yuting Xu, Luoqi Liu, Yafeng Deng, and Ran He. Inconsistency-aware wavelet dual-branch network for face forgery detection. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(3):308–319, 2021.

[10] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2889–2898, 2020.

[11] Dong-Keon Kim and Kwang-Su Kim. Generalized facial manipulation detection with edge region feature extraction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2828–2838, 2022.

[12] Prabhat Kumar, Mayank Vatsa, and Richa Singh. Detecting face2face facial reenactment in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2589–2597, 2020.

[13] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3207–3216, 2020.

[14] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 772–781, 2021.

[15] 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*, pages 667–684. Springer, 2020.

[16] Google Research Nick Dufour and Jigsaw Andrew Gully. Contributing data to deepfake detection research. `https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html`.

[17] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7184–7193, 2019.

[18] 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*, pages 1–11, 2019.

[19] Zekun Sun, Yujie Han, Zeyu Hua, Na Ruan, and Weijia Jia. 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*, pages 3609–3618, 2021.

[20] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.

[21] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016.

[22] Loc Trinh, Michael Tsang, Sirisha Rambhatla, and Yan Liu. Interpretable and trustworthy deepfake detection via dynamic prototypes. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1973–1983, 2021.

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

[24] Deressa Wodajo and Solomon Atnafu. Deepfake video detection using convolutional vision transformer. *arXiv preprint arXiv:2102.11126*, 2021.

[25] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deep-

fake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2185–2194, 2021.

[26] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15023–15033, 2021.

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

[28] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.