ExposeAnyone: Personalized Audio-to-Expression Diffusion Models Are Robust Zero-Shot Face Forgery Detectors

Supplementary Material

A. Evaluation Dataset

Most existing deepfake datasets do not provide reference sets explicitly. Therefore, we carefully construct evaluation sets for the experiments. To compare our model with previous methods, datasets should satisfy the following requirements: datasets should include (1) audio channel, (2) identity labels, and (3) multiple clips for each subject. (4) We also exclude datasets [8, 9, 47] built on VoxCeleb2 [16] because some methods [19, 20] and ours are trained on the dataset. We report the attributes of existing deepfake video datasets in Table 4. Through the evaluation datasets, we exclude videos that are not well face-tracked from our experiments, following the convention (*e.g.*, seen in the LipForensics paper [37]). We describe the construction of our evaluation datasets below:

DF-TIMIT [52]. This dataset is built on VidTIMIT [81] dataset. It includes short term videos of 32 subjects. Each subject has 10 videos with different sentences. They are split into three sessions according to when they were recorded. We use six videos of Session 1 for reference and four remaining videos of Session 2 and 3 for testing. Also, though this dataset provides two different qualities of deepfake videos, we adopt higher (*i.e.*, more challenging) one as lower one is easy to detect for current state-of-the-art methods.

DFDCP [26]. This dataset was released as a preview for a Kaggle competition on deepfake detection[2]. It contains two types of face-swapped deepfake videos. We uniformly sample eight videos for reference and use the remaining videos for evaluation, ensuring that the same scene indexes do not overlap between reference and test sets. The evaluation set contains 39 subjects, and its test set consists of 128 real and fake videos. We uniformly sample real and fake videos for test set, keeping the ratio of real/fake is closer to 1 as much as possible.

KoDF [53]. This dataset focuses on Korean language. We exclude face-swapped videos (*i.e.*, DeepFaceFaceSwap [24], DeepFaceLab [62], and FSGAN [66]) and only adopt face-reenacted videos (*i.e.*, Audio-driven methods [72, 111] and First Order Motion Model [86]) for evaluation. This is because this dataset includes only a single scene (*e.g.*, background and cloth) for each subject; reference-assisted methods can easily spot face-swapped videos by just comparing such irrelevant attributes of input videos with those of reference ones, not based on the talking identities. The evaluation set includes 67 subjects. We use 64 videos at most for each reference set with a duration of eight seconds. We uniformly sample at most four respective real and fake videos for each subject; we exclude videos whose faces are not tracked well and exclude subjects who do not have one or more well-tracked videos for real and fake classes.

IDForge [106]. This dataset provides a large-scale video collection for reference-assisted face forgery detection. It pre-defines the reference and test sets for each subject; therefore, we follow the official split. The evaluation set includes 67 subjects. We use 64 videos at most for each reference set with a duration of eight seconds at most. We uniformly sample at most four respective real and fake videos for each subject; we exclude videos whose faces are not tracked well and exclude subjects who do not have one or more well-tracked videos for real and fake classes.

S2CFP. We create a new dataset to evaluate detection models on the most recent video generation model Sora2 [88].

[2] https://www.kaggle.com/competitions/deepfake-detection-challenge

Dataset	Audio channel	Identity label	Multiple clips for each subject	Independent from VoxCeleb2
DF-TIMIT [52]	✓	✓	✓	✓
UADFV [112]				✓
FF++ [79]				✓
DFD [17]		✓	✓	✓
Celeb-DFv2 [59]		✓	✓	✓
DFDCP [26]	✓	✓	✓	✓
DFDC [27]	✓			✓
DeeperForensics [43]				✓
FFIW [116]				✓
KoDF [53]	✓	✓	✓	✓
FakeAVCeleb [47]	✓	✓	✓	
LAV-DF [8]	✓	✓	✓	
AV-Deepfake1M [9]	✓	✓	✓	
IDForge [106]	✓	✓	✓	✓
Celeb-DF++ [60]		✓	✓	✓

Table 4. **Attributes of deepfake video datasets.** In the main paper, we adopt DF-TIMIT, DFDCP, KoDF, and IDForge datasets that satisfy the requirements for fair evaluation.

Dataset	#ID	Reference Set		Test Set		
		#Real	Duration[s]	#Real	#Fake	Duration[s]
DF-TIMIT	32	192	4.42	128	128	4.00
DFDCP	39	312	15.08	366	378	8.00
KoDF	67	4280	8.00	268	268	8.00
IDForge	53	2880	6.89	187	184	7.08
S2CFP	3	120	8.00	36	36	8.00

Table 5. **Statistics of our evaluation datasets.** #ID, #Real, #Fake, and Duration represent the number of identities, the numbers of real and fake videos, and the mean duration (in seconds) per video, respectively.

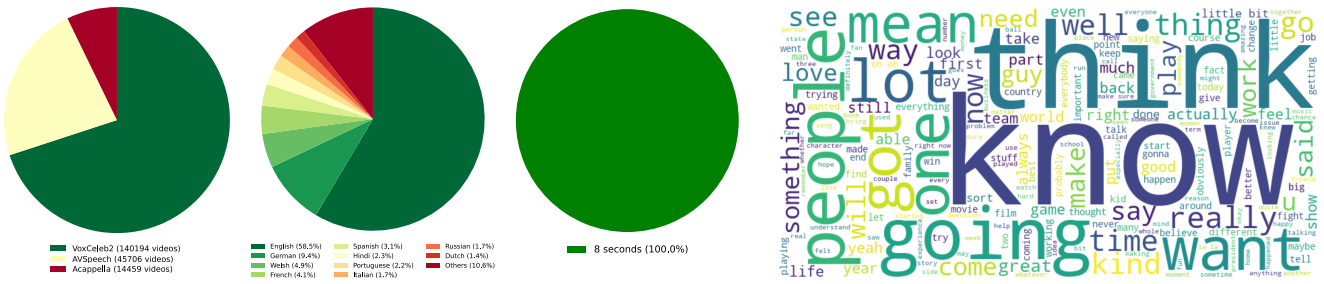


Figure 7. **Pre-training dataset statistics.** (a) We collect monocular videos from three sources; VoxCeleb2, AVSpeech, and Acappella datasets. (b) We visualize the ratio of languages spoken in our dataset. (c) All the sequences of our dataset are clipped into eight seconds. (d) We also visualize the Word Cloud for English.

It provides the Cameo feature that enables us to generate specific subjects although Sora2 does not allow us to generate specific subjects directly from text prompts. Because we can generate only those who make their Cameo avatars public, we collect only three subjects at the moment who are famous and easy to collect real videos on the Web. We will enlarge our dataset so that it includes more subjects in the future.

For each subject, we collect eight real videos of different scenes, then we split them into four videos each for reference and testing. For each video of the reference set, we manually split it into 10 eight-second clips, ensuring that each clip includes only the portions in which the subject is actually speaking and that the subject's face can be detected to extract facial landmarks [7] on all the frames. For that of the test set, we split it into three eight-second clips in the same manner as the reference set.

Then, we generate fake videos with Sora2. To produce a high-quality dataset reducing biases towards video contents, we tried to generate videos whose contents are similar to the original videos. To this end, we used GPT-5 to generate the caption of a frame of each video and used Whisper [75] to generate the transcription of each speech. After that, we input the captions and transcriptions into Sora2 to generate videos corresponding to the original videos, specifying the subjects by the Cameo feature. Note that we replace some sensitive words (*i.e.*, someone's name) in transcriptions with an abstract word such as "a man" because Sora2 refuses such sensitive prompts. The dataset samples of S2CFP are included in the supplementary material. We will make the dataset publicly available.

B. Curated Dataset for Pre-training

We describe the details of curation of our pre-training dataset. The overview of the dataset is shown in Fig. 7. Note that we use Whisper [75] to detect language spoken in videos. The specific data sources and pre-processing are as follows:

Hyperparameter	Value
Optimizer	Adan [104]
Learning Rate	4e-4
Diffusion Steps	1000
β schedule	Linear
Video Duration	8 seconds
Video FPS	25
Expression Dimension	53
Transformer Dimension	512
MLP Dimension	1024
Num Heads	8
Num Layers	8
Dropout	0.1
Classifier-Free Dropout	0.25
Adapter Dimension	512
Adapter Length	8

Table 6. **Hyperparameters for EXAM.**

VoxCeleb2 [16]. This dataset is one of the large-scale datasets of talking head videos from YouTube. The videos are already cropped into a small region around the face. To improve the quality of the samples from this dataset, we only adopt identity-consistent videos ensuring that a single identity appears in each video. We compute identity similarity by ArcFace [25] between all the pairs of frames and then exclude videos that include one or more frames with lower identity similarity than 0.4. As a result, we obtain $140,194$ videos.

AVSpeech [30]. This dataset is also a video dataset of talking people from YouTube. We clip the talking parts using the provided annotations. Because the videos are provided without cropping, we track the faces by checking the overlap of bounding boxes of the adjacent frames. As a result, we obtain $45,706$ videos.

Acappella [11]. This dataset provides videos of singing people from YouTube. Each video contains not only singing parts but also contains talking parts; we adopt both parts to enlarge our dataset. We track and crop the videos in the same manner as the pre-processing for the AVSpeech dataset. As a result, we obtain $14,459$ videos.

C. Implementation Details

C.1. ExposeAnyone Model

Network Architecture. We adopt a similar network with DiT [70] and EDGE [96] but newly introduce TiLM layers for sequential conditioning described in Sec. 3.1. The hyper-parameters of our model are shown in Table 6.

Condition Guidance. Similar to classifier-free guidance (CFG) [40], we train our model with learnable unconditional vectors for both audio and identity conditions to control the strength of conditions during inference. Importantly, we empirically find that, in contrast to generative tasks (*e.g.*, text-to-image synthesis [78]) where models aim to emphasize conditions, the large strength of guidance harms the detection performance. We set the strengths s_a and s_c for audio and identity conditions to 0.5 and 0.25, respectively:

$$\epsilon^{s_a}_{\hat{\theta}_1}(\boldsymbol{z}_t^{1:L}, t, \boldsymbol{w}^{1:H}) = \epsilon_{\hat{\theta}_1}(\boldsymbol{z}_t^{1:L}, t) + s_a\delta_a, \tag{8}$$

$$\delta_a = \epsilon_{\hat{\theta}_1}(\boldsymbol{z}_t^{1:L}, t, \boldsymbol{w}^{1:H}) - \epsilon_{\hat{\theta}_1}(\boldsymbol{z}_t^{1:L}, t) \tag{9}$$

$$\epsilon^{\{s_a,s_c\}}_{\hat{\theta}}(\boldsymbol{z}_t^{1:L}, t, \boldsymbol{w}^{1:H}, \boldsymbol{c}) = \epsilon^{s_a}_{\hat{\theta}_1}(\boldsymbol{z}_t^{1:L}, t, \boldsymbol{w}^{1:H}) + s_c\delta_c, \tag{10}$$

$$\delta_c = \epsilon_{\hat{\theta}}(\boldsymbol{z}_t^{1:L}, t, \boldsymbol{w}^{1:H}, \boldsymbol{c}) - \epsilon_{\hat{\theta}_1}(\boldsymbol{z}_t^{1:L}, t, \boldsymbol{w}^{1:H}) \tag{11}$$

We denote $\epsilon^{s_a}_{\hat{\theta}_1}(\boldsymbol{z}_t^{1:L}, t, \boldsymbol{w}^{1:H})$ and $\epsilon^{\{s_a,s_c\}}_{\hat{\theta}}(\boldsymbol{z}_t^{1:L}, t, \boldsymbol{w}^{1:H}, \boldsymbol{c})$ as $\epsilon_{\hat{\theta}_1}(\boldsymbol{z}_t^{1:L}, t, \boldsymbol{w}^{1:H})$ and $\epsilon_{\hat{\theta}}(\boldsymbol{z}_t^{1:L}, t, \boldsymbol{w}^{1:H}, \boldsymbol{c})$ in Eq. (7) in the main paper for simplicity, respectively.

3DMM Extraction. For pre-training, we assign a single shape shared within each clip, assuming that different videos have different face shapes although some identities overlap each other. For personalization and authentication, we extract a single shape shared between videos considered to belong to an identical subject. This is achieved by using the shape extracted from the first reference video as a fixture in optimizations for all other videos.

C.2. Baselines

EfficientNet-b4. We train the model on FF++ [79] dataset including, real videos and their manipulated ones by Deepfakes [24], Face2Face2 [93], FaceSwap [31], and NeuralTextures [94] using the same pre-process, augmentations, and inference strategy as SBI [83].

Face X-ray. Because there is no official implementation, we re-implement it. We strictly follow the training setting of the original paper; we freeze the backbone [99] for the first 50K iterations and then update all the layers for the remaining 150K iterations on real and blended images. We generate the blended images using the author's unofficial implementation[3].

UCF. We use the DeepfakeBench's implementation [109].

Others. We directly adopt their official implementations.

[3] https://github.com/AlgoHunt/Face-Xray

Setting	DF-TIMIT	DFDCP	KoDF	IDForge	Avg
$t \sim \mathcal{U}[1, 1000]$	**99.94**	91.68	96.02	90.63	94.57
$t \sim \mathcal{U}[101, 900]$	99.84	92.84	**96.14**	91.96	95.20
$t \sim \mathcal{U}[201, 800]$	99.72	93.45	95.31	92.40	**95.22**
$t \sim \mathcal{U}[301, 700]$	99.33	**93.48**	94.07	**92.41**	94.82

Table 7. **Study on timestep sampling.** Excluding early and late timesteps where identity information is not effective for denoising improves the results. Our default setting is highlighted in gray.

Method	Threshold	KoDF
AltFreezing	0.5	86.75
DFD-FCG	0.5	86.57
Ours	$\mu + \sigma$	89.73
Ours	$\mu + 2\sigma$	**90.85**
Ours	$\mu + 3\sigma$	86.38

Table 8. **ACC on KoDF.**

Method	#Params	Time[s]	Avg
LipForensics	36M	0.67	88.92
AltFreezing	27M	3.56	90.34
SBI	18M	0.82	85.50
ForensicsAdapter	435M	0.37	90.33
Ours	31M + 36M	22.2 + 23.6	**95.22**

Table 9. **Model complexity analysis.**

D. Additional Experiments

Timestep Sampling. We explore the sampling strategy of diffusion timesteps during authentication described in Sec. 3.5. We evaluate variants of our method with different sets of timesteps, *i.e.*, $[1, 1000]$, $[101, 900]$, $[201, 800]$, and $[301, 700]$ in Table 7. It can be observed that excluding both sides of timesteps from authentication helps our model detect deepfakes more accurately (94.57% by $[1, 1000]$ vs. 95.22% by $[201, 800]$ in the average AUC). However, the excessive exclusion harms the detection performance (95.22% by $[201, 800]$ vs. 95.20% by $[301, 700]$ in the average AUC). We recommend using the range $[201, 800]$ by default for stable performance on different datasets.

Thresholding. In the main paper, we focus on the potential discriminative ability by evaluating models with a thresholding-free metric, *i.e.*, AUC. Here, we describe how to make decision whether videos are real or fake. We assume that the prediction scores from each subject follow a subject-specific Gaussian distribution. Therefore, we compute the mean μ and the unbiased standard deviation σ from a validation set including eight real videos of the subject. Then, we set a threshold to divide the real and fake classes. To evaluate the effectiveness, we compute the accuracy (ACC) on KoDF in Table 8 with thresholds $\mu + \sigma$, $\mu + 2\sigma$, and $\mu + 3\sigma$. We also show the results of AltFreezing and DFD-FCG, which perform best on KoDF as shown in Table 2, with the threshold simply set to 0.5. Our method achieves consistent accuracy with diverse thresholds, which indicates that our model accurately distinguishes real videos from deepfakes. Note that it is difficult to perform this experiment on DF-TIMIT and DFDCP datasets because they have too limited numbers of videos for each subject to prepare validation sets.

Complexity Analysis. We compare our model with the state-of-the-art methods in terms of the model complexity in Table 9. We compute the number of parameters and the inference time, excluding data loading, per video with eight seconds on a single NVIDIA A100 GPU. Note that we fol-

Setting	DF-TIMIT	DFDCP	KoDF	IDForge	Avg
w/o Audio	99.46	90.76	93.41	91.24	93.72
Ours	**99.72**	**93.45**	**95.31**	**92.40**	**95.22**

Table 10. **Effect of audio conditioning.**

low the official inference strategy for AltFreezing, which can take more time than straightforward inference. Our model has 47M parameters consisting of 31M of SPECTRE and 36M of our diffusion model, which is much smaller than ForensicsAdapter. Our inference takes 22.2 and 23.6 seconds for 3DMM extraction and diffusion authentication, respectively, which is slower than the previous methods. We do not focus on the optimization of inference time in this paper, and it could be improved as follows in future: First, because our 3DMM extraction strategy performs the iterative refinement taking a long time, developing a feed-forward extraction model that directly predicts disentangled FLAME parameters can drastically reduce the overhead. Second, we can reduce the diffusion costs by using a smaller number of noise sequences; we observe in Fig. 5(b) that using a quarter of our default number of noise sequences achieves the same AUC on DFDCP.

Effect of Audio Conditioning. We demonstrate in our framework that conditioning reconstruction on audio helps detection accuracy. To this end, we perform inference by replacing audio conditions with the learned unconditional vector (see App. C.1 and CFG [40] for the unconditional vector). As shown in Table 10, our model without audio conditions drops the AUCs on all the test sets. This result indicates that audio conditioning helps the prediction of the expression coefficients and thus improves detection performance. Notably, our method without audio **still achieves the state-of-the-art generalization ability** in Table 2.

Comprehensive Comparison on S2CFP. We show the result on S2CFP with all the baselines from Table 2 in Table 11. In addition, we refer to the state-of-the-art diffusion-generated image detectors [36, 77, 101]. DIRE [101] argues that diffusion models reconstruct diffusion-generated images more precisely than real images, which enables general diffusion-generated image detection. AEROBLADE [77] applies VAE reconstruction to expose latent-diffusion-generated images. B-Free [36] carefully curates its training set so that there is no content difference between real and fake classes to mitigate biases towards image contents. Even compared with these methods specialized for diffusion-generated image detection, our method achieves the best result with a large margin.

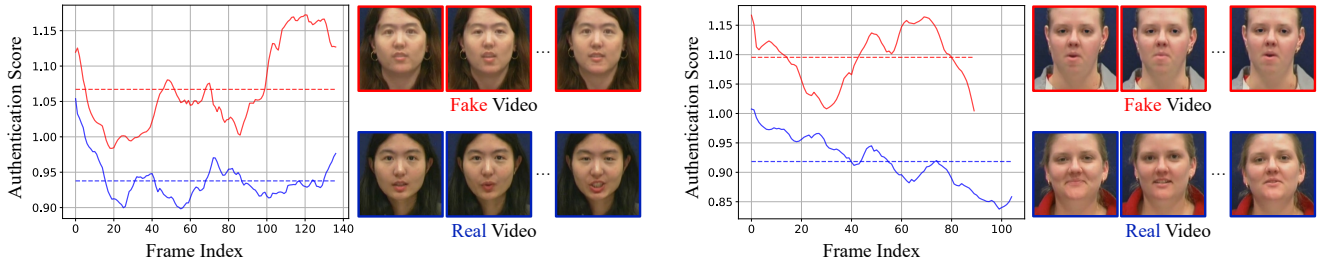
Additional Visualizations. We show additional examples of temporal authentication scores in Fig. 8. Overall, our model performs well on a wide range of subjects and scenes and is robust to a variety of manipulations. We obtain some important observations from the figures: 1) The authentication scores are slightly unstable at silent frames *e.g.*, the beginning and ending of videos. 2) Our model distinguishes real samples from fake ones even in cases where deepfakes' appearances are much closer to real ones, as seen on KoDF and S2CFP, which indicates that our method focuses on highly semantic talking identities for face forgery detection.

Method	Test Set AUC (%) on Each Subject			
	@ijustine	@mcuban	@sama	**Avg**
EfficientNet-b4	65.97	41.67	35.42	47.69
LipForensics	48.61	56.94	33.33	46.29
FTCN	34.03	56.94	29.86	40.28
RECCE	46.53	66.67	49.31	54.17
RealForensics	45.83	58.33	51.39	51.85
AltFreezing	27.78	38.19	15.97	27.31
UCF	36.11	12.50	28.47	25.69
LipFD	17.36	51.39	50.69	39.81
FSFM	40.97	76.39	70.14	62.50
DFD-FCG	36.11	53.47	56.25	48.61
EFFORT	<u>85.42</u>	56.94	61.03	67.80
Face X-ray	4.86	45.83	18.06	22.92
SBI	45.14	45.83	50.69	47.22
ICT	71.53	78.47	34.72	61.57
LAA-Net w/ SBI	22.22	37.50	70.83	43.52
ForensicsAdapter	38.89	82.64	62.50	61.34
ID-Reveal	83.33	**87.50**	75.69	<u>81.02</u>
AVAD	9.03	00.00	15.97	8.33
POI-Forensics	41.67	43.75	72.22	52.55
SpeechForensics	54.86	64.58	63.89	61.11
DIRE	11.11	56.25	43.75	37.04
AEROBLADE	43.75	52.08	<u>91.67</u>	62.50
B-Free	65.97	71.53	76.39	71.30
Ours	**98.61**	<u>84.72</u>	**100.00**	**94.44**

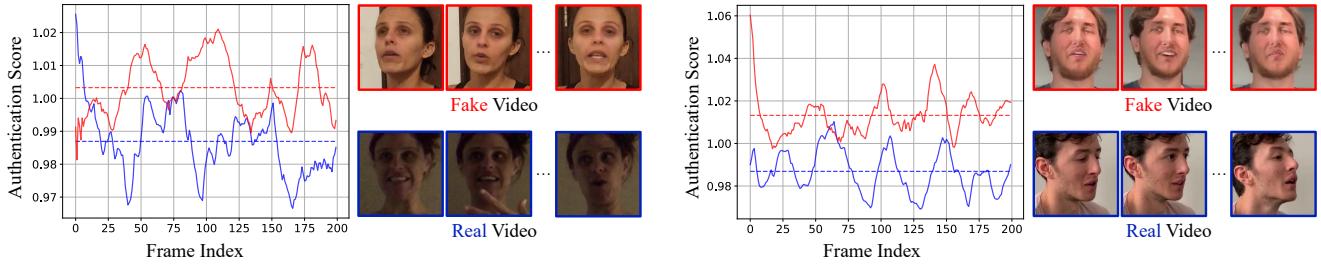
Table 11. **Comprehensive comparison on S2CFP.**

E. Supplementary Video

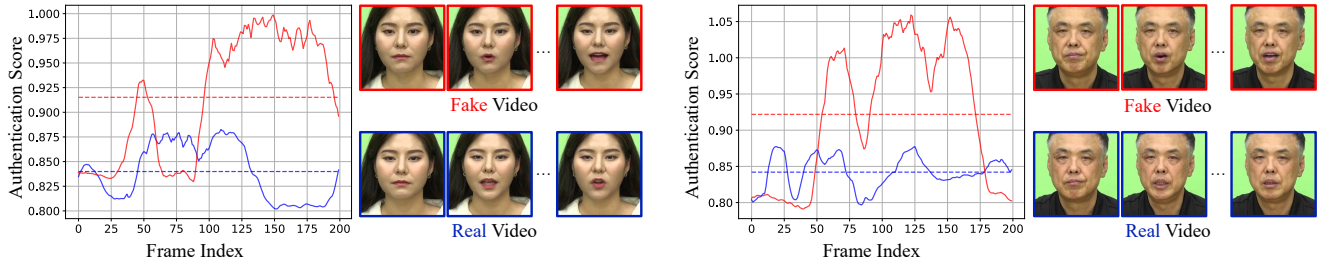
We strongly recommend that readers watch our supplementary video which explains the overview of our work.



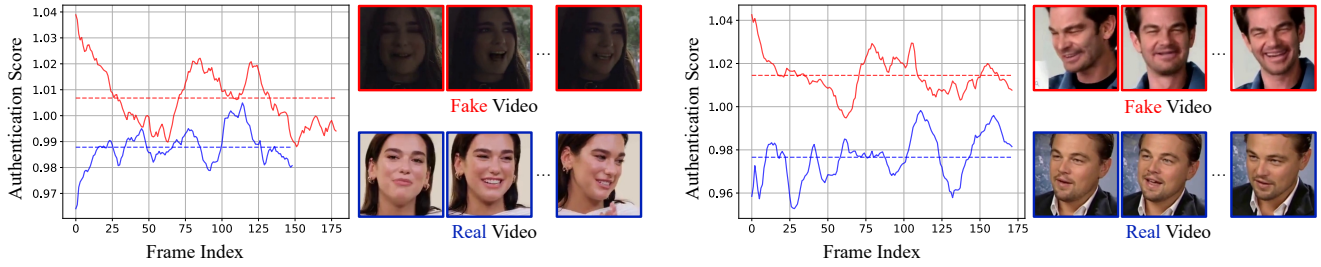
(a) On DF-TIMIT



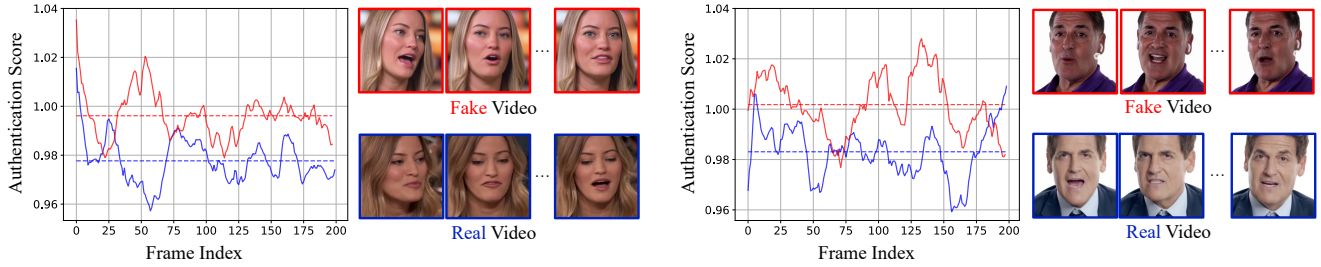
(b) On DFDCP



(c) On KoDF



(d) On IDForge



(e) On S2CFP

Figure 8. **Additional visualizations.** The solid blue and red lines represent the authentication scores over the frame index of real videos and those of deepfakes mimicking the subjects, respectively; the dotted lines denote the corresponding averages.