A-V Representation Learning via Audio Shift Prediction for Multimodal Deepfake Detection and Temporal Localization

Supplementary Material

Algorithm 1 Generate Shift Values for a Single Video

1: **Input:** T_v : video length in frames, τ : segment size
2: **Output:** t : segment-level shift values
3: $shift\_range \leftarrow \{-\tau+1, -\tau+2, \ldots, \tau-1\}$
4: $ns \leftarrow \lfloor T_v/\tau \rfloor$
5: $mid\_segments \leftarrow \max(ns-2, 0)$
6: $rf \leftarrow \left\lceil \frac{mid\_segments + |shift\_range| - 1}{|shift\_range|} \right\rceil$
7: $sample\_pool \leftarrow shift\_range$ repeated rf times
8: $sampled \leftarrow$ randomly sample $mid\_segments$ values from $sample\_pool$ without replacement
9: **if** $ns > 1$ **then**
10: $\quad t \leftarrow [0] + sampled + [0]$
11: **else**
12: $\quad t \leftarrow [0]$
13: **end if**
14: **return** t

1. Deepfake Temporal Localization

Fig. 1 illustrates the pipeline for the Deepfake Temporal Localization task. As described in the main paper, we begin by extracting unimodal and cross-modal features from the pretrained model, concatenating them along the feature dimension, and feed the result into the UMMAFormer [19] regression head. The model first applies the Temporal Feature Abnormal Attention (TFAA) module, which leverages reconstruction-based learning to identify frame-level anomalies. This module uses an encoder-decoder structure to reconstruct features from real samples, then compares the original and reconstructed features through a transformer to highlight abnormal frames via attention. The output is passed to the Parallel Cross-Attention Feature Pyramid Network (PCA-FPN), which performs hierarchical temporal processing by downsampling the features across five levels. Finally, this multi-scale feature pyramid is used to predict frame-level deepfake labels and estimate the start and end timestamps of the closest deepfake segments at each resolution.

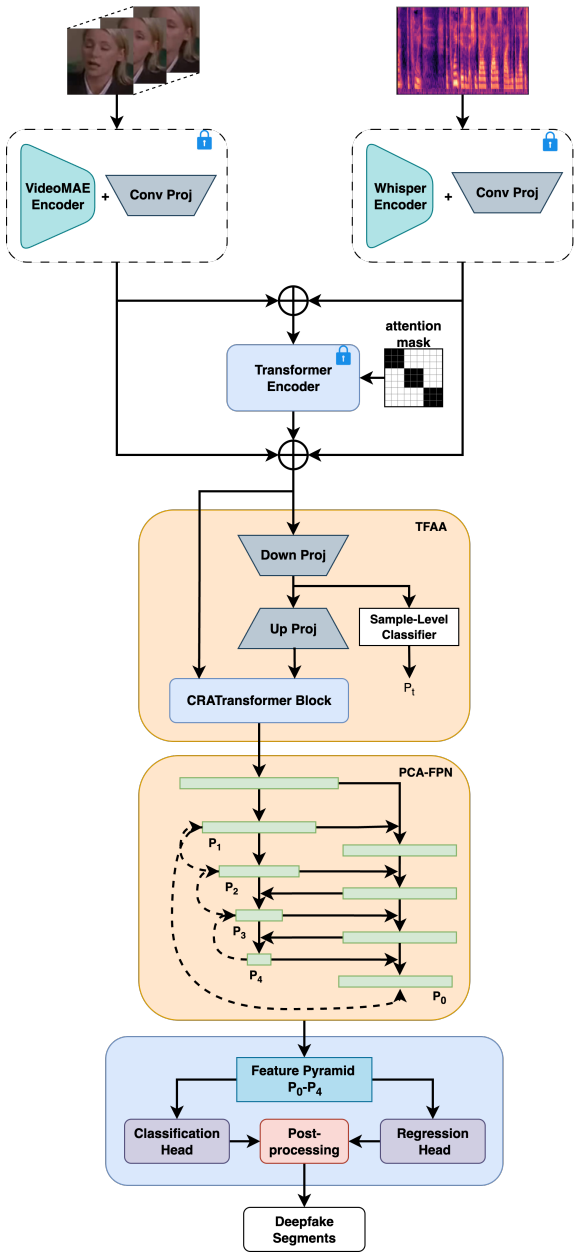

Figure 1. **Deepfake Temporal Localization:** We extract visual, audio, and cross-modal features from the pretrained model, concatenate them, and send them to the regression head, which is adapted from UMMAFormer [19].

2. Shift Value Generation Algorithm

The algorithm described in the main paper (also shown here in Algorithm 1) generates a list of shift values for audio segments in a video, enabling segment-level temporal alignment learning through self-supervised training. Given the video length T_v and segment size τ, it first computes the number of segments ns the video can be divided into (Line 4). Since ns can exceed the size of $shift\_range$, the

range is repeated by a factor rf to create a sample pool large enough for sampling (Line 6 and Line 7). For the middle segments (excluding the first and last), it randomly samples shift values from a predefined range $shift\_range$ (Line 8). The boundary segments are always assigned a zero shift to ensure stability at the edges (Line 10). The final output is a list of shift values, one for each segment, used as targets for the model to learn temporal synchronization.

3. Implementation Details

The videos are processed at 25 fps, and audio is sampled at 16 kHz. For initialization, we use the MARLIN [3] video encoder, based on ViT-Base [6] and pretrained on the YouTube Faces [18] dataset. The audio encoder is initialized with the "large-v3" checkpoint of Whisper [17]. Both encoders' outputs are projected to a 512-dimensional space to ensure feature compatibility (f = 512). We set the segment size τ to 8 and the diversity loss weight λ to 0.001, with all hyperparameters chosen through ablation studies.

To balance memory constraints and provide sufficient content during pretraining, each video is padded to 256 frames (T_v = 256). For audio, we pad each input to 30 seconds before encoding with Whisper, and then truncate the output embeddings to the first 256 frames. Cross-modal fusion is performed using a 6-layer transformer encoder with 4 attention heads and 1024 channels. The shift prediction head consists of two linear layers with output sizes of 512 and $2\tau - 1$, separated by a ReLU activation. Additionally, we follow a speaker-disjoint protocol by removing FakeAVCeleb identities from VoxCeleb2 during pretraining.

For the classification and localization stage, we infer the cross-modal and unimodal features for video input padded to 512 frames. During classification, both cross-modal and unimodal features are projected to 512 dimensions through linear layers, and passed through three feature mixing layers, each comprising two transformer decoder layers with 4 attention heads and 512 channels. For each stage, we train our model on RTX A6000 Ada 48GB GPU for 10 epochs, with AdamW optimizer [13], learning rate 0.0001, and ReduceLROnPlateau scheduler.

4. Dataset Details

VoxCeleb2 [4]: VoxCeleb2 contains real-world YouTube interview videos of over 6,000 public figures, offering more than one million interviews in the development split and approximately 36,000 in the test set. The dataset maintains a relatively balanced gender distribution and spans a wide range of ethnic backgrounds, accents, occupations, and age groups. It also includes recordings under a variety of challenging visual and acoustic conditions. For pretraining our synchronization model, we utilize 17,824 samples for training and 8,928 for validation from this dataset.

FakeAVCeleb [10]: The FakeAVCeleb dataset, built for the task of deepfake detection, consists of 20,000 video samples. Among these, 500 are authentic clips sourced from VoxCeleb2 [4], while the remaining 19,500 are synthetic deepfakes created using a range of generation techniques, including Wav2Lip [16], FaceswapGAN [14], Faceswap [11], and SV2TTS [9]. The dataset is categorized into four distinct groups: Real Visual + Real Audio, Fake Visual + Real Audio, Real Visual + Fake Audio, and Fake Visual + Fake Audio.

KoDF [12]: KoDF is an extensive deepfake video dataset featuring 403 individuals of Korean origin. It includes over 62,000 authentic video samples and more than 175,000 manipulated clips created using six distinct generation techniques, all including visual manipulations. The dataset is designed to support improved generalization to real-world deepfake detection scenarios.

DFDC [5]: The DeepFake Detection Challenge (DFDC) dataset is a multimodal deepfake dataset containing both visual and audio manipulations. It consists of over 100,000 videos from 3,426 individuals, with manipulations generated using eight different methods. While some videos feature a single person, many include extreme camera angles, poses, or lighting conditions. Following [7], we select only videos with a single person and successfully detected faces. As in [7, 8, 15], we sample 3,215 videos for evaluation.

LavDF [1, 2]: The LAV-DF dataset focuses on videos where manipulations are applied only to certain segments rather than the entire clip. It has 78703 samples in the training set, 31501 samples in the validation, and 26100 samples in the test set, with in total of approximately 36,000 genuine samples sourced from VoxCeleb2 and over 99,000 manipulated videos, with alterations affecting the audio, visual, or both modalities. The dataset maintains a balanced distribution across four categories: only visual manipulated, only audio manipulated, both modalities manipulated, and unaltered real samples.

5. Cross-Data Generalization on DFDC

In addition to the KoDF dataset, we further evaluate our model trained on FakeAVCeleb [10] under the cross-dataset setup over samples from the DFDC [5] following the setup of [7, 8, 15]. We achieve an Average Precision of 99.22% and an AUC of 91.86%, showing strong cross-dataset generalization, even including samples recorded in difficult camera angles and lighting conditions.

6. Additional Studies

6.1. Qualitative Result

We visualize the shift predictions for real and deepfake samples from the FakeAVCeleb dataset. Four real and four

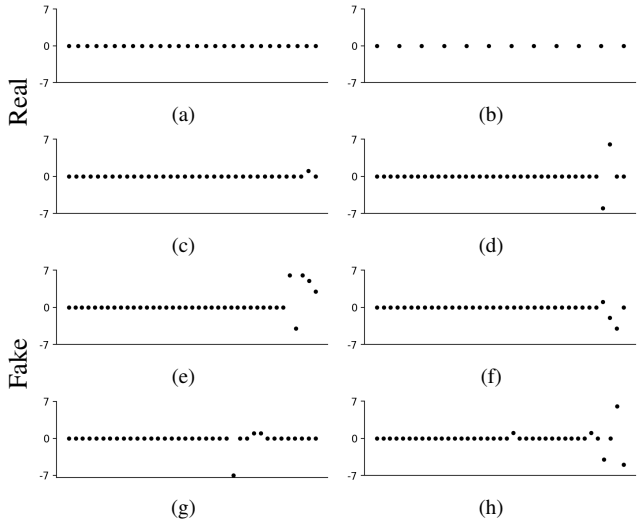

Figure 2. **Shift-prediction for real samples vs deepfakes:** We extract the shift predictions for each segment denoted by a dot and visualize them. Four samples from the real set are shown in (a)–(d), and four from the deepfake set are shown in (e)–(h).

Feature set	AP	AUC
Only C	99.3	80.3
Only V+A	99.6	95.5
Ours (C+V+A)	**99.9**	**97.6**

Table 1. **Feature Set Analysis:** We report the classification performance on FakeAVCeleb when training is conducted using either only cross-modal (C) or only unimodal (V+A) features. Bold values indicate the best results.

fake samples are selected. For each sample, we obtain $ns \times (2\tau - 1)$ logits, apply argmax to derive the predicted shift value for each segment. These predictions are shown as scatter plots in Fig. 2 ((a)–(d) for real and (e)–(h) for fake). From the figure, the pretrained model predicts "no-shift" for most segments in real videos, indicating near-perfect audio-visual alignment. In contrast, deepfakes show multiple segments with non-zero shifts, revealing audio-visual misalignment. Interestingly, some real samples also contain genuine non-zero shifts. The model still classifies these correctly, as it leverages both unimodal cues and audio-visual alignment. The feature mixing layers normalize the reliance on temporal alignment, allowing the model to handle genuine shifts in real videos while remaining sensitive to manipulated inconsistencies.

6.2. Feature Set Analysis

To capture potential artifacts that may be missed during cross-modal alignment, we include both cross-modal and unimodal embeddings in our model through the feature mixing layers. To understand the individual contributions of these embeddings, we evaluate our classification architecture using two configurations: (i) only cross-modal alignment features (C), and (ii) only unimodal visual and audio embeddings (V+A). In each case, since only one type of feature is used, we replace the two transformer decoder layers in each feature mixing block with a single transformer encoder layer for consistency. The results, shown in Tab. 1, demonstrate that combining both cross-modal and unimodal features yields better performance than using either alone. Notably, we observe a significant drop in AUC when using only the cross-modal features (C). This suggests that while cross-modal alignment is effective for multimodal deepfake detection, it may miss certain artifacts that unimodal embeddings can capture. The inclusion of both types of features allows the model to exploit complementary cues, leading to improved overall performance.

Depth	ACC	AUC	Trainable Parameters
3	92.28	97.29	17.4 M
6	**96.23**	**97.59**	**33.2 M**
9	95.65	97.72	48.9 M

Table 2. **Transformer Encoder Depth Analysis:** We evaluated the impact of the number of transformer encoder layers in our pretraining model's cross-modal fusion module. The row denoting the selected depth value is highlighted in bold. The best values under the ACC and AUC metrics are underlined.

6.3. Impact of the Feature Mixing Layers

To evaluate the contribution of our Feature Mixing Layers, we replace them with a simple mixing module of comparable parameter size. This alternative concatenates the cross-modal, audio, and visual features along the feature dimension, followed by two linear layers, one transformer encoder layer, and a final linear layer. The setup achieves an accuracy of 89.1, AP of 99.7, and AUC of 90.1, reflecting drops of 7.1, 0.2, and 7.5, respectively. The substantial decline demonstrates the effectiveness of our feature mixing module, which provides three levels of feature enhancement by enriching cross-modal information with intra-modal cues.

6.4. Transformer Encoder Depth

We experiment with the number of transformer encoder layers in our pretraining model. Table 2 compares depths of 3, 6, and 9 within the cross-modal fusion module. At depth 3, the model shows limited temporal reasoning, with lower accuracy (92.28%) despite a strong AUC (97.29). Increasing to 6 layers raises accuracy to 96.23%, the best among all settings, with a competitive AUC of 97.59, indicating sufficient capacity to capture temporal alignment without overfitting. Depth 9 slightly improves AUC to 97.72, but re-

duces accuracy to 95.65 and increases parameters to 48.9M. Thus, deeper models beyond 6 layers yield diminishing returns, with accuracy saturating or declining while parameter cost rises sharply.

6.5. Impact of the Segmentation Token

We use a learnable segment token, inserted at the beginning of each segment, to capture its summary for the final classification task, following ViT [6]. To assess its impact, we removed the token and instead applied a convolution with kernel size and stride τ (segment size). This convolution-based shift prediction achieved 95.8 accuracy, 99.9 AP, and 97.1 AUC. The segment token yields a modest gain of 0.4% in accuracy and 0.5% in AUC, suggesting that using a dedicated segment-level representation allows the model to capture temporal dependencies more effectively than convolution alone.

6.6. Impact of the Block Attention Mask

Since we shift different audio segments in the input video by varying amounts, we constrain attention in the cross-modal fusion module to within segments using an intra-segment block attention mask. To assess its effect, we removed the mask and allowed the module to attend to all frames, leaving the transformer encoder layers to infer segment-level attention. This setup achieved 94.7 accuracy, 99.9 AP, and 96.2 AUC, reflecting drops of 1.5% in accuracy and 1.4% in AUC. The decline shows that the block-level attention mask eases the burden of figuring out the segment-level attention on the transformer encoder, enabling better modeling of audio shifts and thus yielding higher performance.

6.7. Contribution of Pretraining on Generalization

While our proposed fusion pretraining with pretrained encoders achieves strong intra-dataset performance as well as cross-manipulation and cross-dataset generalization, we assess the specific contribution of shift-prediction pretraining. For this, we use only the pretrained visual and audio encoders to train a single-stage classification network. The visual and audio encodings are max-pooled along the temporal dimension to align with frame-level features, then passed individually through pointwise convolutions to obtain $Z_V \in \mathbb{R}^{T_v \times f}$ and $Z_A \in \mathbb{R}^{T_v \times f}$. These features are concatenated and passed through a linear layer to form a $T_v \times 3f$ representation, matching the input dimension of the classification head. Finally, the feature is fed into the classification head for detection. As a single-stage setup, this approach skips pretraining and relies solely on the pretrained encoders for prediction.

We evaluate this model under intra-dataset, cross-manipulation, and cross-dataset generalization setups, comparing performance with and without pretraining, as reported in Tab. 3. While strong intra-dataset performance is observed even without pretraining, its main contribution lies in improving cross-dataset and cross-manipulation generalization. Pretraining increases intra-dataset AUC by 5.7%, but yields major gains in cross-dataset performance against KoDF, with improvements of 35.1% in AP and 37.1% in AUC. Similarly, in the FVRA-WL category, we observe increases of 3.9% in AP and 32.4% in AUC. These results support the argument that while the single-stage setup using only pretrained encoders performs strongly, it fails to generalize consistently across all cross-manipulation categories and cross-dataset settings, which is effectively addressed by our pretraining.

6.8. Performance Against Benign AV Edits

We evaluated the robustness of our model under several benign audio–visual (AV) transformations. Specifically, we applied three common edits, Gaussian video blurring, Gaussian audio-visual noise, and audio pitch shifting, each tested at multiple intensity levels. We report the performance in Fig. 3. In all three analyses, the value 0 represents the condition with no applied AV edit, serving as the baseline.

Across the experiments, we observe that Average Precision (AP) remains largely stable, showing minimal change even under stronger perturbations. AUC, while more sensitive to noise and distortions, shows only moderate degradation, with the overall performance staying strong, especially under Gaussian blur and pitch shift, where the model maintains high reliability across the full range of edit strengths. These results indicate that the model is consistently robust to benign AV modifications and retains its detection ability under realistic, non-adversarial perturbations.

6.9. Statistical Analysis of Intra- vs Cross-Dataset Performance

To better understand the reliability of our results and whether the differences between intra-dataset and cross-dataset evaluation are statistically significant, we carried out two types of statistical analysis. First, we used bootstrap resampling, which repeatedly samples with replacement from the test set and recalculates the metric to estimate its variability and provide confidence intervals. This allows us to measure how stable the evaluation scores are. Second, we used permutation testing, which randomly shuffles dataset labels between the intra-dataset samples and cross-dataset samples and recomputes the difference in performance. This creates a reference distribution of differences in AUC or AP between intra- and cross-dataset evaluations under the assumption that the dataset assignment does not matter, which we then use to check whether the observed difference in performance is statistically significant.

Using bootstrap resampling with 1,000 iterations, the intra-dataset evaluation achieved an AUC of 97.6 with a 95% confidence interval of [96.88, 98.15] and an AP of 99.9

Method	Intra-dataset		RVFA		FVRA-WL		FVFA-WL		FVFA-GAN		FVFA-FS		KoDF	
	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC
w/o Pretraining	99.8	91.9	97.9	97.2	96.0	66.7	99.7	98.3	**99.9**	**99.9**	99.8	99.2	64.7	62.7
w/ Pretraining	**99.9**	**97.6**	**99.7**	**99.5**	**99.9**	**99.1**	**99.8**	**98.5**	99.6	99.3	**99.9**	**99.3**	**99.8**	**99.8**

Table 3. **Contribution of Pretraining over Generalization:** We compare our model with and without pretraining under the intra-dataset, cross-manipulation, and cross-dataset generalization setups. For cross-manipulation, we consider the same five categories: (i) **RVFA**, (ii) **FVRA-WL**, (iii) **FVFA-WL**, (iv) **FVFA-GAN**, and (v) **FVFA-FS**. We present Average Precision (AP) and AUC scores for each setup. Bold values highlight the best performance in each category.

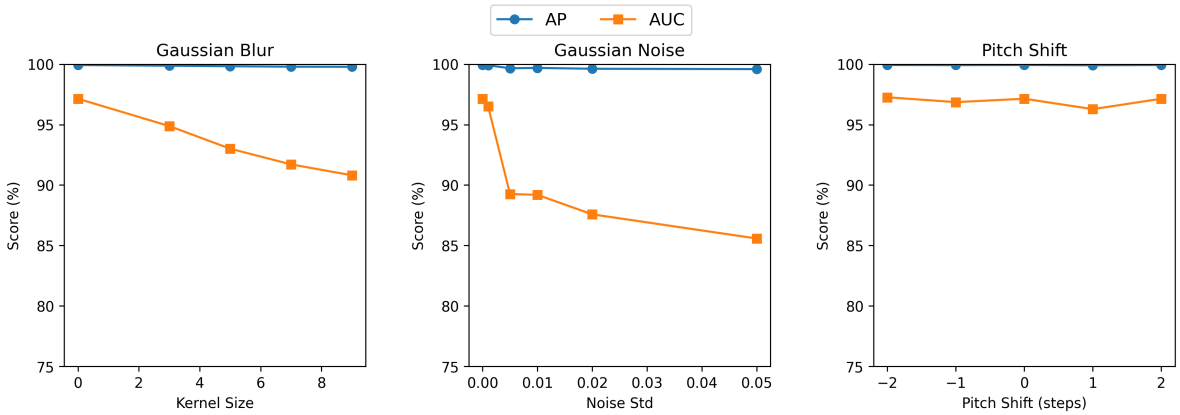

Figure 3. **Benign AV edits**: We test the model's robustness under benign audio–visual edits, including Gaussian blur, Gaussian noise, and pitch shift (0 denotes no edit). AP remains largely stable across all conditions, while AUC shows moderate sensitivity but retains strong overall performance.

with [99.92, 99.96]. The cross-dataset evaluation achieved an AUC of 99.9 with an interval [99.59, 100] and an AP of 99.9 with an interval [99.60, 100]. These narrow confidence intervals show that the scores are highly stable and consistent. Permutation testing further showed that the observed difference in AUC is -0.023 and approaches significance with $p = 0.069$, while the difference in AP is 0.001 and is not significant with $p = 0.199$. Together, these results mean that cross-dataset performance, while numerically close to intra-dataset results, is not statistically different, which supports the claim that the model generalizes well across datasets.

6.10. Effectiveness on Alignment-Preserving Manipulations

One limitation of our approach is its reliance on cross-modal alignment, which may reduce effectiveness against manipulations that preserve temporal consistency across modalities. To evaluate this, we analyze 642 samples (137 real and 505 deepfakes) from FakeAVCeleb where the alignment module predicted no shift, indicating perfect audio-visual synchronization. On these samples, our classification model achieves an AP of 95.9 and an AUC of 87.8, reflecting a drop of 4.0 AP and 9.8 AUC. These results highlight the model's strong dependence on cross-modal alignment, even when augmented with unimodal embeddings through feature mixing layers. A possible mitigation strategy is to redesign the pretraining objective to capture unimodal deepfake artifacts alongside cross-modal ones.

7. Computational Analysis

To evaluate the computational complexity of our model, we measured the inference time and FLOPs separately for the pretraining, classification, and localization stages, as shown in Tab. 4. All measurements were performed with a batch size of 1 on an RTX A6000 Ada 48GB GPU. The pretrained model stage has relatively high GFLOPs and inference time because of the large backbone encoders, whereas the classification module is highly efficient and requires minimal computation. For localization, we report results for our default setting of 512 frames as well as for longer videos with 2048 frames (about 82 seconds). We find that GFLOPs scale linearly with video length, while inference time increases sub-linearly. Finally, the reported localization time also includes the overhead from non-maximum suppression.

Stage	Num Frames	Trainable Params (M)	GFLOPs	Time (ms)
Pretrained Model Inference	512	33.2	4213.13	871.41
Deepfake Classification	512	28.3	6.02	3.27
Deepfake Temporal Localization	512	40.3	9.94	41.18
Deepfake Temporal Localization	2048	40.3	39.75	90.45

Table 4. **Computational analysis:** We report the inference time and computational complexity in terms of GFLOPs for individual stages of the proposed model.

References

[1] Zhixi Cai, Kalin Stefanov, Abhinav Dhall, and Munawar Hayat. Do you really mean that? content driven audio-visual deepfake dataset and multimodal method for temporal forgery localization. In *2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–10. IEEE, 2022. 2

[2] Zhixi Cai, Shreya Ghosh, Abhinav Dhall, Tom Gedeon, Kalin Stefanov, and Munawar Hayat. Glitch in the matrix: A large scale benchmark for content driven audio-visual forgery detection and localization. *Computer Vision and Image Understanding*, 236:103818, 2023. 2

[3] Zhixi Cai, Shreya Ghosh, Kalin Stefanov, Abhinav Dhall, Jianfei Cai, Hamid Rezatofighi, Reza Haffari, and Munawar Hayat. Marlin: Masked autoencoder for facial video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1493–1504, 2023. 2

[4] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018. 2

[5] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset, 2020. 2

[6] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 4

[7] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. 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*, pages 5039–5049, 2021. 2

[8] Alexandros Haliassos, Rodrigo Mira, Stavros Petridis, and Maja Pantic. Leveraging real talking faces via self-supervision for robust forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14950–14962, 2022. 2

[9] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *Advances in neural information processing systems*, 31, 2018. 2

[10] Hasam Khalid, Shahroz Tariq, Minha Kim, and Simon S Woo. Fakeavceleb: A novel audio-video multimodal deepfake dataset. *arXiv preprint arXiv:2108.05080*, 2021. 2

[11] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 3677–3685, 2017. 2

[12] Patrick Kwon, Jaeseong You, Gyuhyeon Nam, Sungwoo Park, and Gyeongsu Chae. Kodf: A large-scale korean deepfake detection dataset. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10744–10753, 2021. 2

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[14] 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. 2

[15] Trevine Oorloff, Surya Koppisetti, Nicolò Bonettini, Divyaraj Solanki, Ben Colman, Yaser Yacoob, Ali Shahriyari, and Gaurav Bharaj. Avff: Audio-visual feature fusion for video deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27102–27112, 2024. 2

[16] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM international conference on multimedia*, pages 484–492, 2020. 2

[17] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023. 2

[18] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pages 529–534. IEEE, 2011. 2

[19] Rui Zhang, Hongxia Wang, Mingshan Du, Hanqing Liu, Yang Zhou, and Qiang Zeng. Ummaformer: A universal multimodal-adaptive transformer framework for temporal forgery localization. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8749–8759, 2023. 1