Joint Audio-Visual Deepfake Detection

Yipin Zhou Ser-Nam Lim
Facebook AI
{yipinzhou, sernamlim}@fb.com

Abstract

Deepfakes ("deep learning" + "fake") are videos synthetically generated with AI algorithms. While they could be entertaining, they could also be misused for falsifying speeches and spreading misinformation. The process to create deepfakes involves both visual and auditory manipulations. Exploration on detecting visual deepfakes has produced a number of detection methods as well as datasets, while audio deepfakes (e.g. synthetic speech from text-to-speech or voice conversion systems) and the relationship between the video and audio modalities have been relatively neglected. In this work, we propose a novel visual / auditory deepfake joint detection task and show that exploiting the intrinsic synchronization between the visual and auditory modalities could benefit deepfake detection. Experiments demonstrate that the proposed joint detection framework outperforms independently trained models, and at the same time, yields superior generalization capability on unseen types of deepfakes.

1. Introduction

A convincing deepfake intentionally designed for delivering spurious information and fake news, e.g. a politician giving a speech or making a statement[1], usually requires meticulous manipulations of both the video and audio channels. In the given example, the video content has been modified with a technique known as lip sync [44, 43] while the voice was from an impersonator. With recent advances in text-to-speech (TTS) and voice conversion (VC) algorithms [58, 41, 22, 37, 10], synthesizing human speech will become even easier, paving a future where audio will play an equally important role as video in deepfake detection. Our work in this paper addresses the interplay between these two modalities, which can be critical towards detecting audiovisual deepfakes.

Recent work has mostly focus on identifying visual artifacts and 'fingerprints' from various generative frameworks [39, 56, 61, 11] or detecting local texture inconsistency caused by face swapping [30, 31]. Another branch of work utilizes biometric signals such as detecting specific facial motion patterns inherent in particular individuals [5, 3], but such ID-specific approach is limited by its ability to generalize to new identities. To achieve a more generalized approach, we observe that when humans speak, there is a strong correlation between the lip motions (viseme) and the pronounced syllables (phoneme) [32]. The synchronization breaks at some inconspicuous moments when any one of the modalities is fake, so for instance, in Fig. 1, the lip motions did not fit well with the syllables due to the artifacts introduced by face swapping or lip sync. Moreover, when the phonemes are created from TTS systems, they are often times not pronounced clearly to match mouth shapes, which is a good signal for detecting audiovisual deepfakes.

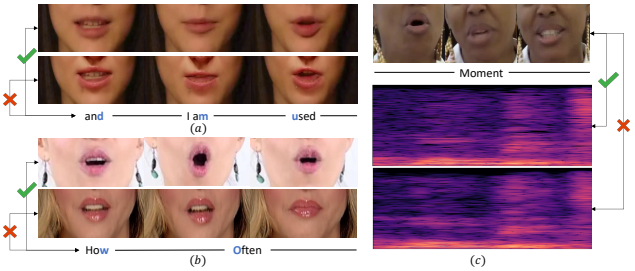


Figure 1: Examples to show that modified video or audio might violate the synchronization patterns. (a) The first row of video frames are unmodified while the second row has been faceswapped, and the words below are spoken by both videos. (b) Same as (a) just the bottom row has been lip-synced instead. Large discrepancies exist between the lip motions in the forged videos and the pronounced words. (c) The top row are authentic video frames, uttering the word "Moment". The corresponding Mel-spectrograms are in the second row, a TTS generated "Moment" Mel-spectrograms are in the third row which sounded more like "wow-mount". The audiovisual pair comprising the first and third row breaks the synchronization patterns maintained by the pair of the first and second row, which is what we hope to capture in this work.

[1] youtu.be/30NvDC1zcL8

Based on this intuition, we present a two-plus-one-stream model to jointly discriminate video / audio deepfakes. Existing multi-modal frameworks take paired inputs from different modalities (e.g. video frames and optical flow; or video and sound for Action Recognition task), and use a shared label on the fused representation, which can be based on a late fusion of two streams [42, 18, 25] or lateral connections [17]. For deepfakes, the labels for the audio and video streams may not necessarily be the same because it could be that either one of the modalities is modified. Learning a shared latent representation like this could thus be sub-optimal.

For this reason, we propose to model the video and audio stream separately with their own labels, as well as temporally aligning the coarse-to-fine representations from both streams. We refer to this as sync-stream, which itself is given a separate label that reflects whether any one of the modalities has been manipulated. By jointly training as we proposed, the network not only learns 'appearance' or texture artifacts but also benefits from the sync-steam that discriminates synchronization patterns of authentic audio-visual pairs from that of fake pairs. A limiting factor, however, is the lack of a proper dataset with both visual and auditory manipulations. To overcome this, we utilize existing video deepfake datasets containing unmodified audio channels, from which we extract the Mel-spectrograms [51]. By running these spectrograms through different vocoders [38, 52, 23, 33, 19, 28, 60] that are commonly used in TTS and VC tasks to mimic synthesized speech, we eventually curated a dataset similar in size to existing video deepfake datasets, but with manipulated audio channels.

Our contributions can be summarized as follows:

1. We present a joint audiovisual deepfake detection task that handles the case that either one (or both) of the visual or auditory modalities have been manipulated.

2. Further, we propose a sync-stream that models the synchronization patterns of two modalities. We show that with this additional signal, our model generalizes well as a result to unseen deepfakes.

3. Finally, we have built a deepfake dataset that contains both visual and auditory manipulations, with which we hope to encourage further research in the area of joint audiovisual deepfake detection.

2. Related Work

Video deepfake detection: [61, 56] demonstrate that there exists 'fingerprints' for different GAN frameworks that can be used to detected generated images. [39] presents a forensics dataset of video data manipulated by four existing methods and a XceptionNet [13] baseline for detecting deepfakes. [30, 31] propose more general face forgery detection methods, utilizing the discontinuity between modified and authentic regions. In [11], a patch-based detection framework is proposed to make local predictions then aggregate. [5, 3] learn personalized facial action patterns and make use of this biometric signal to detect face swaps. The approaches in [20, 4] are most relevant to this paper; both address lip motions while detecting deepfakes. [20] fine-tunes the model using a pre-trained lipreading network to learn embeddings that are more sensitive to mouth movements. However, unlike our proposal, there is no audio involved. [4] explicitly extracts phonemes and visemes from video and audio pairs to detect mis-matches. Our framework additionally exploits synchronization patterns of authentic pairs (versus modified pairs) via learning through an independent video and audio stream together with a sync-stream.

Audio deepfake detection: [49] releases a large-scale spoofed audio dataset comprising synthetic speech generated with state-of-the-art neural acoustic and waveform generation models as well as replayed attacking speech. Many existing spoofed speech detection frameworks [45, 12, 6, 8, 29, 16] rely on extracting acoustic representations like MFCC [34], STFT and CQCC [48] from raw waveform signals and applying classifiers such as SVM, Gaussian Mixture Model (GMM) or CNN to make predictions. [59] proposes an end to end ResNet-like framework to identify spoofed audios.

Video and audio cross-modeling: Visual and sound modalities are often intertwined, [36, 62, 9, 27] make use of the the concurrent property to provide supervision while training a network without annotation. Video and audio could also compliment each other by providing semantics from different perspectives. To this end, [62, 7, 35, 21] all train their models jointly with both modalities to learn richer representations. In this paper, we similarly exploit the concurrency property between authentic video (mouth movements) and human speech, and attempts to detect when one or both modalities are modified, causing the concurrency to be broken. Additionally, unlike prior work, where audio and video share the same label / latent space, our framework follows a multi-task setting with a separate video and audio stream that are given their own labels and linked by a sync-stream.

3. Methodology

Problem Formulation: We denote an input video containing human talking as $x = \{a, v\}$, where a, v are the respective audio and video channels and are sequences of sampled waveform digits and video frames. The network that makes prediction is denoted as $\mathcal{F}(x)$, which includes two parts: the feature extractor $\mathcal{F}_\theta$ maps input video or audio into a feature representation in $\mathbb{R}^{T\times d}$ with T and d respectively being the length of sequence and feature dimension; the classification layer $\mathcal{F}_\phi$ maps feature representations to labels.

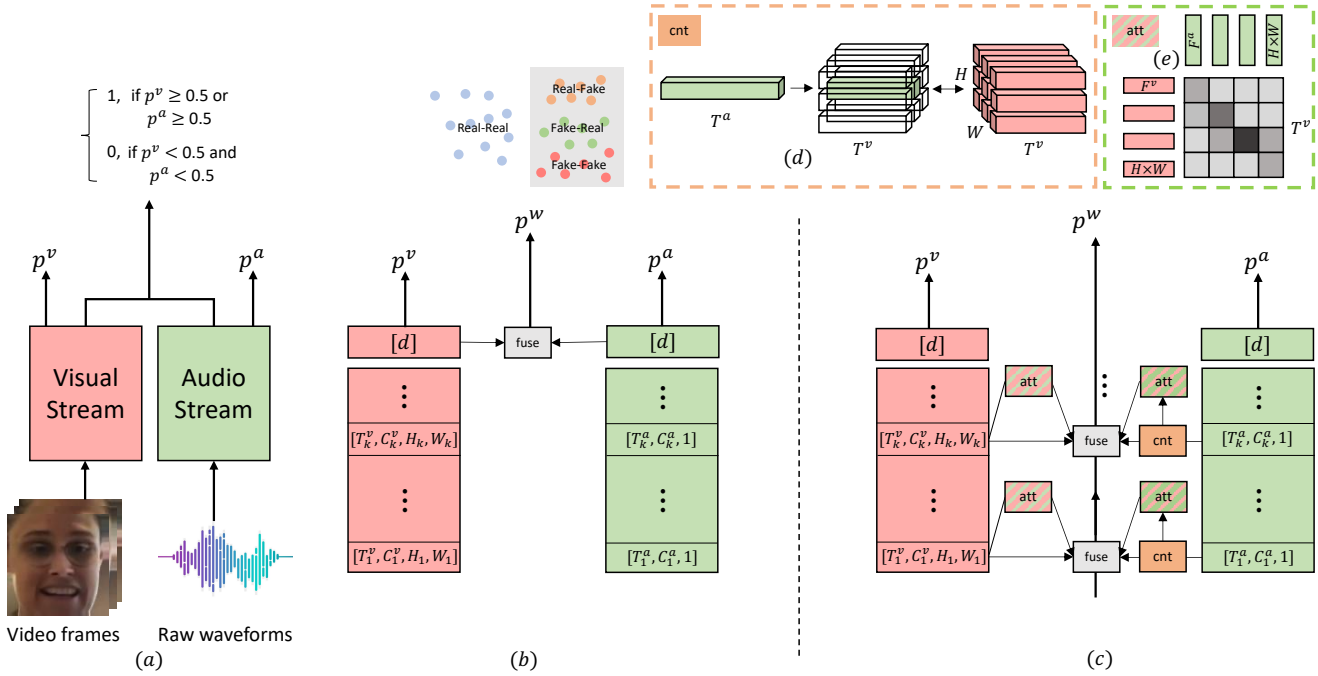


Figure 2: Overview of proposed frameworks. (a) Independently trained prediction framework. p^v and p^a are outputs for video and audio streams, representing the probability of the input being fake. (b) Late-fused joint prediction network with p^w denoting the probability of the whole video being modified and $[d]$ denoting last layers of networks where representations are spatiotemporally pooled. (c) Two-plus-one joint detection framework. The 'cnt' operation is illustrated in (d), where we use $conv$ and $tile$ to align the shapes of the visual and audio features. (e) shows the $inter$ attention mechanism as an example. C (channel dim) is weighted-pooled into 1 and H×W is flattened as the feature embedding. The attention is working on the temporal dimensions and the weight is a $T^v \times T^v$ matrix. The weighted representation will be unpooled / unflattened and added back to the original feature as a residual.

Let $D = \{a_i, v_i, y_i^a, y_i^v, y_i\}_{i=1}^N$ be our dataset, where $y \in \{0, 1\}$ is the label indicating whether the input is real or fake. y_i^v and y_i^a are labels for the video and audio channels respectively and are independent of each other. y_i is the label for the whole audiovisual sequence, and is defined as fake when either of the modalities has been modified. Deepfake detection is a binary classification task and the respective loss functions for the video and audio modality detection are defined as:

$$L_{cls}^v = \sum_{(v,y^v)\in D} \mathcal{C}[y^v, \mathcal{F}_\phi^v(\mathcal{F}_\theta^v(v))], \quad (1)$$

$$L_{cls}^a = \sum_{(a,y^a)\in D} \mathcal{C}[y^a, \mathcal{F}_\phi^a(\mathcal{F}_\theta^a(a))], \quad (2)$$

where $\mathcal{C}[.]$ is the cross-entropy loss. To combine the predictions from the video and audio streams, we apply the aggregation operation $\tilde{y} = \mathcal{G}(.)$ to obtain the final prediction. Various choices of $\mathcal{G}(.)$ are considered in this paper, including our proposal. We discuss them now in the following sections.

3.1. Independently trained video and audio streams

As a baseline, we have the video $\mathcal{F}_\phi^v(\mathcal{F}_\theta^v(.))$ and audio $\mathcal{F}_\phi^a(\mathcal{F}_\theta^a(.))$ streams trained and making predictions independently. Unlike two-stream framework [42] where class scores from two modalities are fused (averaged) after softmax as the final prediction, the label for the whole sequence is fake if either of the streams are classified as fake, otherwise it will be classified as real:

$$\tilde{y} = \mathcal{G}((\mathcal{F}_\phi^v(\mathcal{F}_\theta^v(v)) \geq 0.5), (\mathcal{F}_\phi^a(\mathcal{F}_\theta^a(a)) \geq 0.5)). \quad (3)$$

We use 0.5 as the threshold and $\mathcal{G}(.)$ represents the $\mathbb{OR}$ operation here as depicted in Fig. 2(a).

3.2. Late fusion of video and audio streams

Applying late fusion is a straight-forward operation to jointly learn with multiple modalities [24, 57]. As Fig. 2(b) shows, we simply extract latest feature representations from the audio and video streams (right before each classification head) and we use $\mathcal{F}_\theta^{(-1)}$ to represent this process. Features

from the two streams are then fused together before a prediction is made on whether the whole sequence has been modified. In this case, we have an additional loss for this prediction:

$$L_{cls} = \sum_{(v,a,y)\in D} \mathcal{C}[y, \mathcal{G}(\mathcal{F}_\theta^{v(-1)}(v), \mathcal{F}_\theta^{a(-1)}(a))]. \quad (4)$$

Here, $\mathcal{G}(.)$ represents an aggregation operation (e.g. addition) and classification head for the whole sequence. To jointly train the model, we have the loss:

$$\mathcal{L} = w^v L_{cls}^v + w^a L_{cls}^a + w L_{cls} \quad (5)$$

where w^v, w^a and w are the respective weights for each stream. As shown in the experimental section (Sec. 4.3), this approach performs worse than independently trained framework. The reasons can be two folds: 1) Based on the observation from [57], jointly training a multi-modality network could cause sub-par results because the different modalities tend to converge and generalize at different rates; 2) The spatial and temporal information are abstracted into high-level semantics in late fusion. This suggests that only fusing highly-abstracted feature representations of the video and audio networks is not enough to learn the subtleties of the intrinsic synchronization patterns.

3.3. Two-plus-one streams

Our proposed framework is to apply central connections to video and audio streams between low-level features, which encode spatial (for video frames) and temporal information, to higher-level semantic representations. Specifically we build a sync-stream by connecting video and audio network feature representations as shown in Fig. 2(c).

Central connections. We fuse video and audio streams into the sync-stream. Before temporal and spatial axes get pooled, there could be a mismatch between the sizes of the feature representations of the two modalities, so we apply $conv$ and $tile$ operations on the audio features. We denote the visual feature shape as $\{T^v, C^v, H, W\}$ and the audio feature as $\{T^a, C^a, 1\}$ (since we process raw waveform samples for audio channel, the feature dimension will be 1). We conduct strided 1D convolution on the audio features to pool the temporal length from T^a to T^v, after which we tile the features $H \times W$ times as Fig. 2(d) shows to align the temporal and spatial axes. By jointly training, the network will automatically learn the correspondence between the audio and corresponding visual regions over time. We show visualization of the learned correspondence in Sec. 4.5. At each layer, the audio and visual representation will be fused with the current layer of sync-stream and used as input to the fusion at the next layer.

Denoting the feature representation from the i^{th} layer as $\mathcal{F}_\theta^{(i)}$, prediction of the whole sequence can now be represented as:

$$\begin{aligned}\tilde{y} = \mathcal{G}(\dots \mathcal{F}_\theta^{v(i-k)}(v) \rightarrow \mathcal{F}_\theta^{v(i)}(v) \rightarrow \mathcal{F}_\theta^{v(i+k)}(v) \dots, \\ \dots \mathcal{F}_\theta^{a(i-k)}(a) \rightarrow \mathcal{F}_\theta^{a(i)}(a) \rightarrow \mathcal{F}_\theta^{a(i+k)}(a) \dots)\end{aligned} \quad (6)$$

where $\mathcal{G}(.)$ is now the central connections and the classification head and $\rightarrow$ indicates network forwarding. Sync-stream is jointly trained with video and audio streams and the supervision is done in the same way as the late-fusion framework.

Since the audio features are largely pooled over time to fit the length of the video sequence as we have described, a given audio representation might be better aligned with multiple video frames or vice versa, and there may be only specific moments that the network needs to pay attention to. To associate different positions and learn a better temporal alignment between the video and audio channels, we further apply intra (self-attention, [54]) and inter-attention mechanisms within and across the video and audio modalities. We experiment with the following attention mechanisms in sync-stream:

1) Inter-attention: In inter-attention, the weight is computed by involving both visual and audio representations.

$$InterAtt(v) = softmax(\frac{\mathcal{F}_\theta^{a(i)} \mathcal{F}_\theta^{v(i)T}}{\sqrt{d}}) \mathcal{F}_\theta^{v(i)} \quad (7)$$

$$InterAtt(a) = softmax(\frac{\mathcal{F}_\theta^{v(i)} \mathcal{F}_\theta^{a(i)T}}{\sqrt{d}}) \mathcal{F}_\theta^{a(i)} \quad (8)$$

where T is the transpose operation and d represents visual and audio feature dimensions.

2) Inter+intra-attention: Besides inter-attention we also apply self-attention on visual and audio representation respectively. We only show visual intra-attention below, but the same formulation applies to the audio counterpart.

$$IntraAtt(v) = softmax(\frac{\mathcal{F}_\theta^{v(i)} \mathcal{F}_\theta^{v(i)T}}{\sqrt{d}}) \mathcal{F}_\theta^{v(i)} \quad (9)$$

3) Joint-attention: Slightly different with inter-attention, in joint-attention, we apply the same attention weights on visual and audio representations:

$$JointAtt(a, v) = softmax(\frac{\mathcal{F}_\theta^{v(i)} \mathcal{F}_\theta^{a(i)T}}{\sqrt{d}})(\mathcal{F}_\theta^{v(i)} + \mathcal{F}_\theta^{a(i)}) \quad (10)$$

Before the attention operations are conducted, positional encoding [54] is applied and we add weighted representa-

tion back to the original feature as a residual (Fig. 2(c)).

During training, the sync-stream plays a role of enriching the video and audio representation with learned synchronization patterns. During inference, for all practical purposes, we would utilize the output of the sync-stream as a preliminary prediction. A positive prediction will be followed by examining the video and audio branch to determine the final prediction. We provide results supporting this approach in the supplementary material.

4. Experiments

4.1. Dataset

To the best of our knowledge, no large-scale dataset exists that provides high-quality visual and auditory deepfakes (with video and audio channels that are well-aligned and without clear mis-match that could be perceived by humans immediately). One straightforward way is to take existing video deepfake datasets with audio channels, extract the transcripts from the speech and apply TTS [58, 41] algorithms to convert to synthetic speech. The drawback of this approach is that it is hard to control the synchronization between the motion of the lip and the speech without additional constraints.

While the visual artifacts are easily 'borrowed' from existing high quality video deepfake datasets, a high quality audiovisual deepfake dataset would need to have synthesis artifacts in the audio channels without sacrificing the synchronization between the video and audio channels. To achieve this, we extract Mel-spectrogram from the audio channel of video deepfakes and apply various vocoders, which are commonly used by TTS and VC algorithms, including **Griffin-Lim** [19], **WORLD** [33] and CNN-based methods: **WaveNet** [52], **WaveRNN** [23], **Parallel-WaveGan** [60], **WaveGlow** [38] and **MelGAN** [28]. Extracting the Mel-spectrogram and converting it to a signal that human can perceive in this way preserves the synchronization, yet introduce the artifacts that comes from using the vocoders. We also noticed that TTS generates speech that tends to lack 'sharpness' (sometimes words are not clear). To mimic this, we apply random blurriness on the Mel-spectrograms. To prove the effectiveness of our audio synthesis, we train an audio deepfake classifier with the converted data to detect in-the-wild synthetic speeches[2] generated by unknown TTS algorithms. The model achieved an accuracy of 81.96% (89.55 AUC). More details are available in the supplementary materials.

The final dataset is curated from two video deepfake datasets in which audio channels are available.

FF [39]: Faceforensics++ is a deepfake benchmark dataset comprising 5000 video sequences, where the video channel is manipulated with 4 methods: Deepfakes [1], Face2Face [47], FaceSwap [2] and NeuralTexture [46]. We crawl the original videos from YouTube to recover the audio channel. Interestingly, while the majority of the speech in FF is non-English, our experiments show that the performance did not degrade, highlighting that our approach is actually language-agnostic.

DFDC [15]: The DeepFake Detection Challenge Dataset is currently the largest video deepfake dataset with more than 100,000 video and audio (in English) sequences. We remove those where the sound is from the cameraman instead of the actor(s) in order to ensure audiovisual synchronization.

We follow the splits for train / val / test from the original datasets, and randomly swap authentic audio with synthesized audio from one of the conversion methods mentioned. For testing, we keep the number of 'real-fake' (video is real and audio is fake), 'fake-real', 'real-real' and 'fake-fake' balanced.

[2] youtu.be/SK4vGKxm1PY

	Vid	*Aud*	*Sync*
Stem	k=$\begin{bmatrix}1\times7\times7\\3\times1\times1\end{bmatrix}$, c=64	-	-
Layer1	k=$\begin{bmatrix}1\times3\times3\\3\times1\times1\end{bmatrix}\times4$, c=64	[k=80, c=128, s=4] maxpool[k=4, s=4]	k=$\begin{bmatrix}1\times3\times3\\3\times1\times1\end{bmatrix}\times2$, c=64 att[d=56×56, h=4]
Layer2	k=$\begin{bmatrix}1\times3\times3\\3\times1\times1\end{bmatrix}\times4$, c=128	[k=3, c=128, s=1] maxpool[k=4, s=4]	k=$\begin{bmatrix}1\times3\times3\\3\times1\times1\end{bmatrix}\times2$, c=128 attd[d=28×28, h=1]
Layer3	k=$\begin{bmatrix}1\times3\times3\\3\times1\times1\end{bmatrix}\times4$, c=256	[k=3, c=256, s=1] maxpool[k=4, s=4]	k=$\begin{bmatrix}1\times3\times3\\3\times1\times1\end{bmatrix}\times2$, c=256 att[d=14×14, h=1]
Layer4	k=$\begin{bmatrix}1\times3\times3\\3\times1\times1\end{bmatrix}\times4$, c=512	[k=3, c=512, s=1] maxpool[4, 4]	k=$\begin{bmatrix}1\times3\times3\\3\times1\times1\end{bmatrix}\times2$, c=512 att[d=7×7, h=1]
FC	AdaAvgPool3D fc 512×2	AdaAvgPool1D fc 512×2	AdaAvgPool3D fc 512×2

Table 1: Network architecture. Notations: k is kernal size; c is number of channels; s is stride; d represents embedding size and h is number of heads. For layers in bold, Vid and Aud are central connected with sync-stream. In FC layer, $AdaAvgPool$ applies adaptive average spatiotemporal pooling that generates a 512-dimensional feature vector.

4.2. Implementation

The backbone architecture for the video stream is a R(2+1)D-18 [50] network. For the audio stream, we utilize a simple 1D convolutional network to process 1D raw waveform signals. Even though the structures of the two streams are very different, we show in the following sections that our sync-stream is able to boost performance effectively. In the sync-stream, 4 layers of feature representations are extracted from both streams and linked through central con-

	Att type	*Vid*	*Aud*	*Whole*
Indp	-	98.41	93.06	95.83 (-)
Late-fuse	-	99.21	92.26	94.25 (99.08)
+Fix-indp-audvid	-	-	-	94.84 (98.44)
{2+1}-streams	-	98.81	96.23	97.02 (99.41)
+Fix-indp-audvid	-	-	-	96.23 (99.29)
{2+1}-streams	Inter+intra	99.21	94.25	97.22 (99.29)
+Fix-indp-audvid	Inter+intra	-	-	96.03 (98.74)
{2+1}-streams	Joint	99.21	94.64	96.23 (99.25)
+Fix-indp-audvid	Joint	-	-	96.03 (99.18)
{2+1}-streams	Inter	98.81	96.03	**97.62 (99.65)**
+Fix-indp-audvid	Inter	-	-	**96.63 (99.32)**

	Vid	*Aud*	*Whole*
Indp	79.90	98.28	89.36 (-)
Late-fuse	80.65	98.07	89.15 (93.70)
+Fix-indp-audvid	-	-	89.81 (95.12)
{2+1}-streams	82.18	98.36	**91.01 (96.29)**
+Fix-indp-audvid	-	-	90.33 (**96.32**)
{2+1}-streams+att	82.39	98.95	90.84 (95.98)
+Fix-indp-audvid	-	-	**90.40** (96.29)

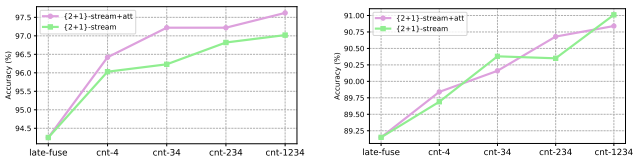
Table 2: Numerical evaluation on FF (left) and DFDC (right). We show classification accuracy (%) for video, audio streams as well as whole videos, and the numbers in parentheses are AUCs. **Indp** represents independently trained framework; **Late-fuse** is late-fused architecture; and **{2+1}-stream** represents joint detection network with sync-stream (with/without attention mechanisms). **+Fix-indp-audvid** share the same architectures with the above method while weights of video and audio streams are from **Indp** and fixed during training.

nections. We apply the same (2+1)D conv between layers of sync-stream like visual stack, and use addition as the fusion and aggregation operations. We show the network architecture including attention parameters in Table. 1.

The sampling rate is 10 FPS for the video sequence and 22.05 KHz for the audio sequence. The inputs to the networks are 3s long, which equals 30 video frames and 66150 audio samples. For sequences that are shorter than 3s, we pad them with empty frames / digits (0s). In the following experiments, we use the first 3s for video level classification. Our framework could also handle arbitrary lengths by running a sliding window then aggregating the probabilities, which might achieve more stable video-level predictions but the inference would be slower. We employ the face detector in [63] to detect, crop and align faces over time. During training, we utilize the Adam Stochastic Optimization [26] with a learning rate of 0.0002 and a minibatch size of 64 for all models. The weights in Eqn. 5 are set equally to $w^v = w^a = w = \frac{1}{3}$.

4.3. Numerical evaluation

In this section, we show quantitative results on the **FF** and **DFDC** datasets, following the original training, validation and testing splits. Referring to Table 2 (left), we show video-level classification accuracy on FF of (1) a model with independently trained video and audio streams (**Indp**); (2) a model jointly trained on the two modalities with late fusion (**Late-fuse**); and (3) our proposed model with sync-stream (**{2+1}-streams**). We also compare sync-streams with various types of attention. Further, in order to show the contribution of sync-stream in predicting manipulation of whole sequence, we initialize the video and audio streams in **Late-fuse** and **{2+1}-streams** with weights from independently trained models and fix the weights during training. We denote this experiment as **Fix-indp-audvid**.



(a) FF (b) DFDC

Figure 3: Sync-stream accuracy (%) of whole sequences for various architectures on FF and DFDC datasets.

Based on the results, we observe that **{2+1}-streams** consistently outperforms **Indp** and **Late-fuse** on the whole sequence detection (whether one or both modalities are modified) as well as single tasks of video and audio deepfake detection. We compare the results between different types of attention mechanisms and find that *Inter+intra* and *Joint-attention* perform worse than *Inter-attention*, especially on the whole sequence and audio stream prediction. This indicates that cross-modality attention contributes more towards the learning of the synchronization patterns than self-attention does. We will apply *Inter-attention* on the top of sync-stream for the rest of the experiments. In Table. 2 (right), we conduct experiments on **DFDC** dataset and similarly observe that sync-stream boosts the performance of whole / video / audio deepfake detections.

Ablation: To understand how different levels of spatiotemporal abstraction contribute to the performance of sync-stream, we compare frameworks with various sync-stream architectures. We use *cnt-1234* to denote centrally connected audio / visual features from layer 1 to 4 (Table. 1),

	Deepfakes+Griffin-Lim	*FaceSwap+WaveRNN*	*Face2Face+WORLD*	*NeuralTextures+WaveNet*
Xception[39]	93.90	51.20	86.80	79.70
CNN-aug[56]	87.50	56.30	80.10	67.80
Patch-based[11]	94.00	60.50	87.30	84.80
Face X-ray[30]	99.50	**93.20**	94.50	92.50
CNN-GRU[40]	97.60	47.60	85.80	86.60
LipForensics[20]	99.70	90.10	99.70	**99.10**
ResCNN+GRU[59]	80.93	83.51	87.21	89.37
Indp	98.78 / 67.07 / 83.94 (99.92 / 80.06 / -)	55.69 / **90.65** / 73.37 (73.06 / **96.37** / -)	95.93 / 78.46 / 86.99 (99.53 / 86.81 / -)	80.89 / **92.68** / 86.58 (96.40 / 96.81 / -)
{2+1}-streams+att	**99.19 / 77.85 / 87.40** (**99.99** / **84.66** / **94.36**)	**77.24** / 86.79 / **85.16** (90.48 / 96.01 / **89.39**)	**96.75 / 81.91 / 87.80** (**99.79** / **89.25** / **93.71**)	**82.92** / 92.28 / **88.01** (98.32 / **97.92** / **95.04**)

Table 3: Unseen category evaluation on FF. Each column represents the video and audio deepfake category that was left out for testing. For the last two rows, we show accuracy (%) on video/audio/whole stacks and corresponding AUC below in parenthesis. AUCs for video and audio deepfake detection are shown in pink and green respectively.

and *late-fuse* to denote that only features after spatiotemporal pooling are connected. We can observe from Fig. 3 that the accuracy of predicting on the whole sequence improves as we utilize more lower-level information, highlighting the importance of both low-level as well as highly-abstracted spatiotemporal information when learning the synchronization patterns between the video and audio streams.

4.4. Generalization on unseen categories

We expect sync-stream to provide better generalization by learning synchronization patterns that are agnostic to the type of manipulation, resulting in more robust audiovisual representations. In Table 3, we conduct experiments on unseen video and audio deepfake categories on **FF**. Specifically, we leave out one type of video deepfake as well as one type of vocoder for testing and use the remaining categories for training. We utilize two vocoders that applies hand-crafted features, **Griffin-Lim** and **WORLD**, and two deep neural network based methods, **WaveNet** and **WaveRNN**. Experiments show that our **{2+1}-stream** network is more robust on unseen data compared with independently trained models. We also compare with state of the art detection methods that work on video streams only (no audio involved). These are: (1) **Xception** [39], which proposes a frame-based deepfake detection framework; (2) **CNN-aug** [56], which applies pre- and post-processing and data augmentation to increase the robustness; (3) **Face X-ray** [30], which makes use of discontinuity between local regions; (4) **Patch-based** [11], which makes local predictions with small receptive fields; (5) **CNN-GRU** [40], which utilizes a video-based detection framework based on recurrent architectures; and (6) **LipForensics** [20], which fine-tunes on a lip reading network. We follow the same experimental settings as [20] and use the statistics reported in the paper. Our method shows competitive results with [20] and outperforms other methods, some of which are designed for generalizing on unseen data. [20] applies a pre-trained lip reading network as a strong prior, while we demonstrate that the synchronization could be automatically learned with audio involved. For audio only detection, we implement an end-to-end framework [59] for audio spoofing detection, which consists of a GRU [14] on the top of a residual CNN (**ResCNN+GRU**) with magnitude spectrogram as the input. We can see from Table. 3 that the audio stack of **{2+1}-stream** network is superior in terms of generalizing to unseen audio categories, even though it uses a simpler architecture. Please note that our joint detection framework supports any video and audio backbones, so it's performance could be further improved with deeper networks.

4.5. Analysis

Visualization: We visualize which regions the network focus on to make decisions with/without jointly training with audio modality. To do so, we apply ScoreCAM [55] on the last convolutional layer (best compromise between high-level semantics and spatialtemporal information) of the video stream. For each input video frame, a heat map is generated that shows the regions in the frame that are key to the final predictions. In Fig. 4, we show results from independently trained video deepfake detector as well as from the visual stack of the **{2+1}-stream+att** network. For independently trained networks, the model tends to focus on large areas of faces, eyes and mouth tips where artifacts might appear. This is consistent with the observation from [11]. For our joint detection framework, the majority of the attention falls on the mouth regions indicating that the network has indeed learned the synchronization between the visual and auditory signals.

Audio channel shuffling: We further demonstrate the importance of learning synchronization patterns between the video and audio channels. We randomly swap audio chan-

(a) (b) (c)

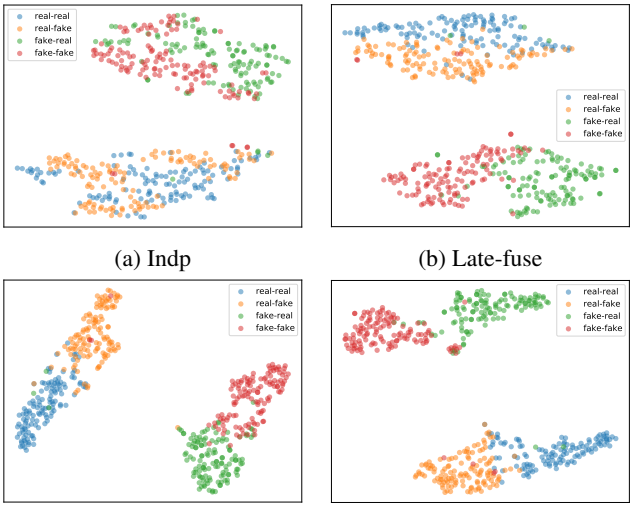
Figure 4: Visualization on where the network focus on while making predictions. (a) Frames from DFDC. (b) Frames from FF. For each set of results, the top row is from independently trained network and the bottom row is from joint detection framwork with sync-stream. (c) Visualization from the shuffling experiments. The hsv colormap shows the intensity.

	Vid	*Aud*	*Whole*
Indp	98.41	93.06	95.83 (-)
{2+1}-streams+att	98.81	96.03	97.62 (99.65)
{2+1}-shuffle	98.02	91.07	95.44 (98.65)
	0.39 | 0.79	1.99 | 4.96	0.39 | 2.18
{2+1}-att-shuffle	98.21	92.46	95.44 (98.90)
	0.20 | 0.60	0.60 | 3.57	0.39 | 2.18

Table 4: Experiments for shuffling audio channels to train joint-detection network with unpaired video and audio. Numbers in parenthesis are AUCs. Numbers in red are performance drop in accuracy (%) compared with first and second rows.

nels of different videos while training the same **{2+1}-stream** architecture on **FF**, meaning that the video and audio are not paired anymore. We present the results of such a shuffled framework in Table. 4. Compared with jointly trained models with paired data or even independently trained networks, we observe a performance drop as there is no concurrent patterns between viseme and phoneme anymore. We also visualize where the shuffled framework pays attention to for making predictions in Fig. 4(c), and we can see that they are mostly random regions like noses or even flickering backgrounds.

Feature representations: To understand the efficacy of the aggregated representations (**Indp**, **Late-fuse** and **{2+1}-stream+att**), we employ t-SNE [53] to visualize the clustering of 4 different groups, namely, 'real+fake' (video is real and audio is fake), 'fake+real', 'fake+fake' and 'real+real'. Specifically, we aggregate the last layer of the video and audio streams with addition operations. As Fig. 5 shows, aggregated representations produced by sync-stream are more discriminating than **Late-fuse** and **Indp**.



(a) Indp (b) Late-fuse

(c) {2+1}-stream (d) {2+1}-stream+att

Figure 5: t-SNE visualization on feature space of aggregated visual and audio representations.

5. Conclusion

In this paper, we propose a novel task on detecting deepfakes by jointly modeling video and audio modalities. This task is important because in practice, we do not have any prior knowledge about whether it's the video or the audio that have been manipulated. We show that utilizing learned intrinsic synchronization between video and audio boosted the performance of both video and audio based deepfake detection as well as whole sequence prediction. The learned synchronization patterns further help the model generalize to unseen deepfake categories. Finally, we provide a high-qualtiy audiovisual deepfake dataset that will be useful for future research in this direction.

References

[1] Deepfakes repo. https://github.com/deepfakes/faceswap.

[2] Faceswap repo. https://github.com/MarekKowalski/FaceSwap/.

[3] Shruti Agarwal, Hany Farid, Tarek El-Gaaly, and Ser-Nam Lim. Protecting world leaders against deep fakes. In *WIFS*, 2020.

[4] S. Agarwal, H. Farid, O. Fried, and M. Agrawala. Detecting deep-fake videos from phoneme-viseme mismatches. In *CVPRW*, 2020.

[5] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In *CVPRW*, 2019.

[6] K.N.R.K. Raju Alluri and Anil Kumar Vuppala. IIIT-H Spoofing Countermeasures for Automatic Speaker Verification Spoofing and Countermeasures Challenge 2019. In *INTERSPEECH*, 2019.

[7] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020.

[8] Moustafa Alzantot, Ziqi Wang, and Mani B. Srivastava. Deep residual neural networks for audio spoofing detection. *CoRR*, 2019.

[9] R. Arandjelović and A. Zisserman. Look, listen and learn. In *ICCV*, 2017.

[10] Sercan Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *NeurIPS*, 2018.

[11] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *ECCV*, 2020.

[12] Bhusan Chettri, Daniel Stoller, Veronica Morfi, Marco A. Martínez Ramírez, Emmanouil Benetos, and Bob L. Sturm. Ensemble models for spoofing detection in automatic speaker verification. In *INTERSPEECH*, 2019.

[13] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017.

[14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, 2014.

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

[16] S. Fatemifar, S. R. Arashloo, M. Awais, and J. Kittler. Spoofing attack detection by anomaly detection. In *ICASSP*, 2019.

[17] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *NeurIPS*, 2016.

[18] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.

[19] D. Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. In *ICASSP*, 1983.

[20] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. *CoRR*, 2020.

[21] D. Hu, F. Nie, and X. Li. Deep multimodal clustering for unsupervised audiovisual learning. In *CVPR*, 2019.

[22] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, and Yonghui Wu. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *NeurIPS*, 2018.

[23] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. In *ICML*, 2018.

[24] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[25] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[27] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018.

[28] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In *NeurIPS*, 2019.

[29] Galina Lavrentyeva, Sergey Novoselov, Andzhukaev Tseren, Marina Volkova, Artem Gorlanov, and Alexandr Kozlov. STC antispoofing systems for the asvspoof2019 challenge. In Gernot Kubin and Zdravko Kacic, editors, *INTERSPEECH*, 2019.

[30] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo. Face x-ray for more general face forgery detection. In *CVPR*, 2020.

[31] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. In *CVPRW*, 2019.

[32] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 1976.

[33] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. World: A vocoder-based high-quality speech synthesis system for real-time applications. In *IEICE Transactions on Information and Systems*, 2016.

[34] Lindasalwa Muda, Mumtaj Begam, and I. Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *CoRR*, 2010.

[35] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *ICML*, 2011.

[36] Andrew Owens, Jiajun Wu, Josh H. McDermott, William T. Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016.

[37] Wei Ping, Kainan Peng, and Jitong Chen. Clarinet: Parallel wave generation in end-to-end text-to-speech. In *ICLR*, 2019.

[38] R. Prenger, R. Valle, and B. Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP*, 2019.

[39] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *ICCV*, 2019.

[40] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. In *CVPRW*, 2019.

[41] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *ICASSP*, 2018.

[42] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014.

[43] Linsen Song, Wayne Wu, Chen Qian, Chen Qian, and Chen Change Loy. Everybody's talkin': Let me talk as you want. *CoRR*, 2020.

[44] Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: Learning lip sync from audio. *ACM Trans. Graph.*, 2017.

[45] Hemlata Tak, Jose Patino, Andreas Nautsch, Nicholas Evans, and Massimiliano Todisco. Spoofing attack detection using the non-linear fusion of sub-band classifiers. In *INTERSPEECH*, 2020.

[46] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph.*, 2019.

[47] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. *Commun. ACM*, 2018.

[48] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans. A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients. In *Odyssey*, 2016.

[49] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection. In *INTERSPEECH*, 2019.

[50] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[51] S. Umesh, L. Cohen, and D. Nelson. Fitting the mel scale. In *ICASSP*, 1999.

[52] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *CoRR*, 2016.

[53] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[55] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *CVPRW*, 2020.

[56] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, 2020.

[57] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *CVPR*, 2020.

[58] Yuxuan Wang, R.J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis. In *Interspeech*, 2017.

[59] Jee weon Jung, Hye jin Shim, Hee-Soo Heo, and Ha-Jin Yu. Replay attack detection with complementary high-resolution information using end-to-end dnn for the asvspoof 2019 challenge. *CoRR*, 2019.

[60] R. Yamamoto, E. Song, and J. Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP*, 2020.

[61] Ning Yu, Larry S. Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *ICCV*, 2019.

[62] Antonio Torralba Yusuf Aytar, Carl Vondrick. Learning sound representations from unlabeled video. In *NeurIPS*, 2016.

[63] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 2016.