On the Audio-visual Synchronization for Lip-to-Speech Synthesis

Zhe Niu and *Brian Mak*
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
{zniu,mak}@cse.ust.hk

Abstract

Most lip-to-speech (LTS) synthesis models are trained and evaluated with the assumption that the audio-video pairs in the dataset are well synchronized. In this work, we demonstrate that commonly used audiovisual datasets such as GRID, TCD-TIMIT, and Lip2Wav can, however, have the data asynchrony issue, which will lead to inaccurate evaluation with conventional time alignment-sensitive metrics such as STOI, ESTOI, and MCD. Moreover, training an LTS model with such datasets can result in model asynchrony, meaning that the generated speech and input video are out of sync. To address these problems, we first provide a time-alignment frontend for the commonly used metrics to ensure accurate evaluation. Then, we propose a synchronized lip-to-speech (SLTS) model with an automatic synchronization mechanism (ASM) that corrects data asynchrony and penalizes model asynchrony during training. We evaluated the effectiveness of our approach on both artificial and popular audiovisual datasets. Our proposed method outperforms existing SOTA models in a variety of evaluation metrics.

1. Introduction

Lip-to-speech (LTS) refers to the task of reconstructing spoken audio from a speaker's lip movements in a video that lacks sound. It is especially useful in circumstances where audio is missing due to various reasons, such as inadequate recording devices, ambient noise, transmission failures, *etc*. Deep learning has significantly advanced this field, with the development of various data-driven deep networks aimed at solving the LTS task.

During the training and evaluation of LTS models, it is often assumed that there is little or no time offset between the corresponding audio-video data pair, as audiovisual datasets are usually believed to be well synchronized. However, as shown in Fig. 2 and studies by others [26], commonly used datasets for training and evaluating LTS models have varying time offsets within their audio-video pairs, which we refer to as *data asynchrony*. While some datasets such as GRID [4] and TCD-TIMIT [7] have small offsets within ±1 video frame (*i.e.*, $\pm$40ms), others like Lip2Wav [20] can have larger offsets of multiple video frames.

Data asynchrony can significantly impact the evaluation of LTS models. Even a slight misalignment between the reference and the test audio can have a major impact on the vanilla STOI [24], ESTOI [11] and MCD [16] scores during evaluation. To address this issue, we propose a time-alignment frontend that precedes the computation of alignment-sensitive metrics. This frontend effectively miti-

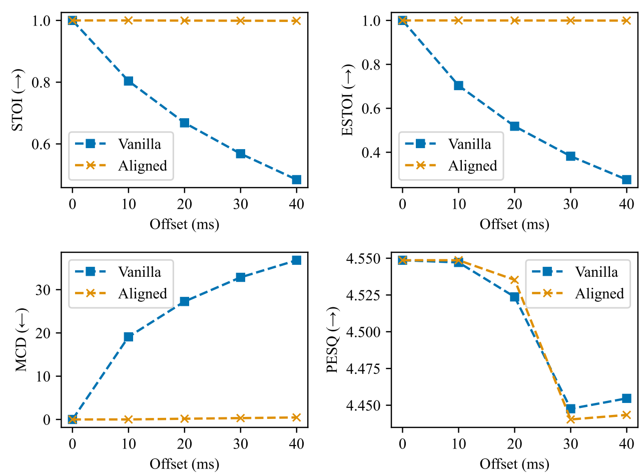
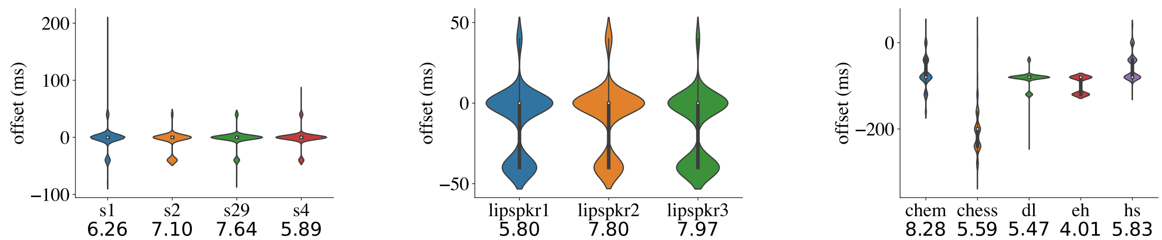


Figure 1: The impact of offsets on alignment-sensitive metrics such as STOI, ESTOI, and MCD can be significant. A 40ms offset, which is equivalent to a single video frame at 25 fps, can lead to severe degradation in the scores of the original versions of these metrics (*i.e*. Vanilla). PESQ is less affected due to its alignment mechanism. Our proposed solution, which is a time alignment frontend applied to each of the metrics (*i.e*. Aligned), ensures consistent scores regardless of the offsets. The results were obtained by computing the scores between the ground truth audio as the reference and its offset versions as the test audios on the test set of GRID-4S (as described in Section 4.1).



(a) GRID-4S. (b) TCD-TIMIT-LS. (c) Lip2Wav.

Figure 2: AV offsets produced by SyncNet on various datasets with the SyncNet confidence scores noted beneath the speaker ID. We only consider offsets with confidence scores greater than 3.0. Most of the samples in the GRID-4S and TCD-TIMIT-LS datasets exhibit zero or slight offsets of ± 40 ms. In contrast, the Lip2Wav dataset shows larger offsets, with the *chess* speaker exhibiting -200 to -250 ms and other speakers showing offsets around -80 ms.

gates the impact of synchronization errors on conventional time-alignment sensitive metrics, thus ensuring consistent scoring despite data asynchrony, as demonstrated in Fig. 1.

When a model is trained on a dataset with synchronization errors, it can generate offset audio against the input video during inference. We refer to this issue as *model asynchrony*. To train LTS models that can handle asynchrony in datasets and produce synchronized output, we propose the synchronized lip-to-speech (SLTS) model, which incorporates an automatic synchronization mechanism (ASM) to ensure synchronization from both the data and model perspectives during training. The ASM overcomes data asynchrony with a data synchronization module (DSM) and prevents model asynchrony using the self-synchronization module (SSM).

In our experiment section, we first validated the robustness of the proposed SLTS model on a small-scale artificial dataset with severe data asynchrony issue called GRID-4S-Async, which we created from GRID [4] by adding artificial audio-video offsets uniformly sampled from -150 to 150 ms. Then we tested the SLTS model on popular audiovisual datasets, including GRID-4S [4], TCD-TIMIT-LS [7], and Lip2Wav [20]. Our findings demonstrate that SLTS is effective on datasets with either obvious asynchrony that can be seen by the human eye (*e.g*. GRID-4S-Async and Lip2Wav [20]), or slight synchronization errors (*e.g*. single frame or sub-frame offsets, such as in GRID-4S and TCD-TIMIT-LS [7]).

In summary, this paper offers several contributions to the field of lip-to-speech (LTS) synthesis. We begin by identifying two types of asynchrony issues that arise during the development of LTS models: *data asynchrony* and *model asynchrony*. Next, we propose a time alignment frontend to enable consistent evaluation regardless of the time offsets in the audiovisual dataset. Following this, we introduce a novel synchronized lip-to-speech (SLTS) model, which incorporates an automatic synchronization mechanism (ASM) that actively learns audiovisual time offsets during training, aiding in the rectification of data asynchrony and alleviating model asynchrony. The SLTS model shows competitive and, in many cases, superior performance on various datasets, leading to high-quality and synchronized audio reconstruction.

2. Related works

Synchronization in lip-to-speech models. Lip-to-speech models often use components with large temporal receptive fields, such as 3D convolutional stacks [20], LSTM/GRU [1, 20, 28, 18], location-sensitive attention [20, 9], and self-attention [13, 27], which are vulnerable to model asynchrony. When the dataset exhibits data asynchrony, the large receptive field can cause the model to generate audio that is offset from the input video. Kim *et al*. [13] suggested the use of additional synchronization losses during training as a solution for model asynchrony. However, their method does not take into account the issue of data asynchrony.

Lip-sync models. The primary objective of lip-sync models is to accurately predict audiovisual offsets, thus rectifying any synchronization errors. Existing works, such as [3, 14], construct positive (*i.e*. in-sync) and negative (*i.e*. off-sync) audiovisual pairs to train the model with contrastive loss. Chung *et al*. [3] utilize the audiovisual samples in the training set as positive pairs. They generate negative pairs by randomly shifting the audio and train their network using contrastive loss from Siamese networks[2]. Kim *et al*. [14] adopt a softmax-based contrastive loss, treating audio and video features from the same time step as positive pairs and those from different time steps as negative pairs. Both methods assume that the audiovisual dataset is well synchronized to create the positive and negative pairs. In contrast, our proposed data synchronization module (DSM) does not assume that the dataset is synchronized. Instead, it processes a set of candidate pairs and discovers positive and negative pairs in an unsupervised manner, driven by the learn-

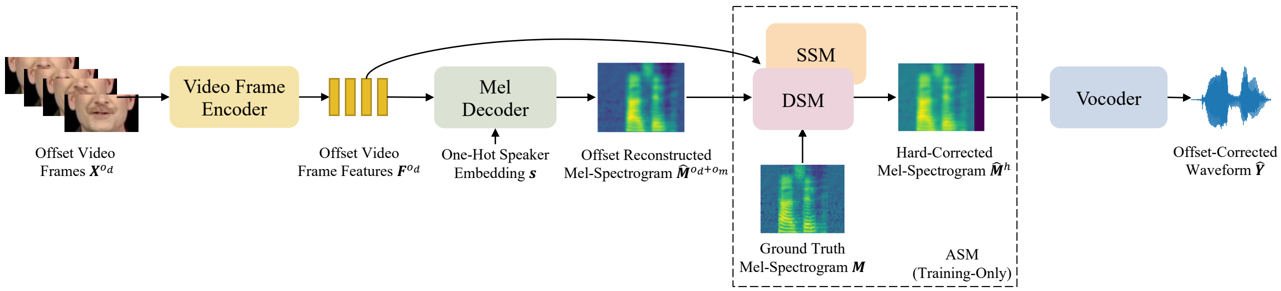


Figure 3: An overview of the proposed SLTS model architecture. Where o_d is due to data asynchrony and o_m is due to model asynchrony; both are measured in seconds. The two asynchrony issues are handled by DSM and SSM respectively.

ing objective of the lip-to-speech task (to be introduced in Sec. 3.5).

End-to-end lip-to-speech models. Lip-to-speech models typically represent waveforms using more compact acoustic features, such as mel-spectrograms. As a result, these models require a vocoder to convert these acoustic representations into audio waveforms. Commonly used vocoders include the algorithmic Griffin-Lim used in [20, 13, 28], and separately trained neural vocoders explored in [12, 9, 17]. Recently, there has been increasing interest in developing end-to-end lip-to-speech models that directly generate audio waveforms [18, 27]. In this work, we also delve into end-to-end modeling by jointly training a UnivNet vocoder [10] as part of the proposed model.

3. Synchronized lip-to-speech synthesis

In this section, we outline our synchronization methods for both evaluation and training. We start with the metric alignment frontend, a mechanism that guarantees consistent evaluation, irrespective of dataset asynchrony. Following this, we turn our attention to our synchronized lip-to-speech (SLTS) model, detailing the automatic synchronization mechanism and our end-to-end training objectives.

3.1. Metric alignment frontend

As shown in Fig. 1, alignment-sensitive metrics such as STOI [24], ESTOI [11] and MCD [16] can produce inaccurate scores when the two input audio signals are not time-aligned. To solve this issue, we propose a time alignment frontend to synchronize the *degraded* (*i.e.* generated) and *reference* (*i.e.* ground truth) audio signals.

The frontend initiates by identifying the optimal alignment, which is subsequently used to adjust the degraded audio signal. To accomplish this, we utilize a method based on grid search. First, we extract mel-spectrograms from both the degraded and reference 16-kHz audio signals using a window size of 40 ms and a hop length of 10 ms, and then L2-normalize these mel-spectrograms across the mel-frequency bands. Following this, we shift the normalized mel-spectrogram of the degraded audio frame by frame, generating a range from -30 to 30 frames, equivalent to -300 to 300 ms, resulting in 61 potential shift proposals. We then identify the shift proposal that yields the smallest mean squared error (MSE) when compared to the normalized reference mel-spectrogram, and select this proposal as the optimal one for adjusting the degraded audio. The adjusted degraded audio and original reference audio are subsequently used as input for conventional metrics to produce the final scoring.

During the aforementioned process, length discrepancies can arise due to the shifting operation. When searching for the optimal shift proposal, we truncate the reference mel-spectrogram, either from the start or end based on the direction of the shift, to ensure precise alignment. Once the best offset is identified, we instead correct the length mismatch by padding the degraded audio with silence (*i.e.*, a value of 0), either at the beginning or end as needed, while ensuring the reference audio remains unaltered before sending to the metrics.

In Fig. 1, we compare the commonly used metrics in their original form (*i.e.* Vanilla) and when they are used with our proposed time alignment frontend (*i.e.* Aligned). Our time alignment frontend consistently provides accurate scoring, regardless of the AV offsets.

3.2. SLTS model architecture overview

Once we have addressed the issues related to the evaluation metrics, our attention shifts towards improving the model architecture for training with asynchronous datasets. To this end, we introduce the synchronized lip-to-speech (SLTS) model, as depicted in Fig. 3. Without loss of generality, we assume that the audio in each audio-video pair is offset free and is referred to as *ground truth audio*, while the video in the pair, however, can be offset from the audio and is referred to as *offset video*. During training, SLTS reconstructs an offset-corrected audio waveform $\hat{\boldsymbol{Y}} \in \mathbb{R}^{T_w}$ from a silent offset video $\boldsymbol{X}^{o_d} \in \mathbb{R}^{T_v \times H \times W \times 3}$ that has an offset of o_d seconds to match ground truth audio $\boldsymbol{Y} \in \mathbb{R}^{T_w}$. Here, T_w is the length of the audio waveform, T_v is the

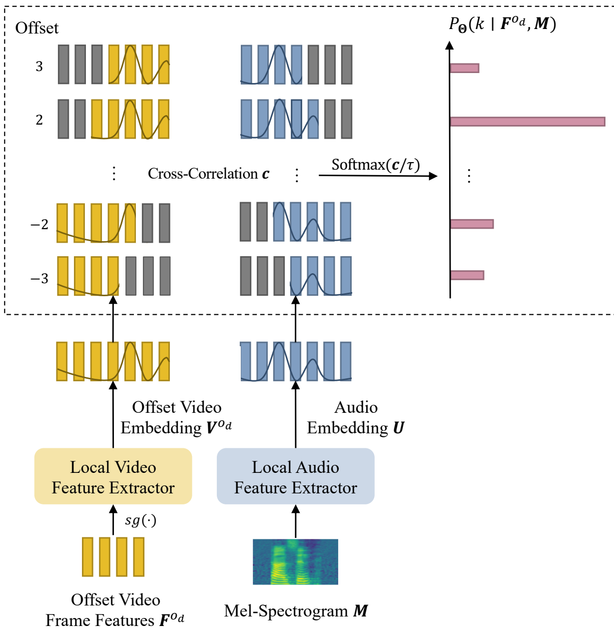


Figure 4: Proposed AV offset predictor architecture.

number of video frames, H, W, and 3 denote the height, width, and number of channels of the RGB video frame, respectively. During inference, the SLTS model operates in two modes. When a reference mel-spectrogram is given, the SLTS model generates a synchronized audio signal, denoted $\hat{\boldsymbol{Y}}$. If not, it creates an offset audio, $\hat{\boldsymbol{Y}}^{o_d}$, which aligns with the offset input video, $\boldsymbol{X}^{o_d}$. The former mode serves as an AV synchronizer, while the latter is used for the LTS task.

3.3. Lip-to-speech backbone

The proposed synchronized lip-to-speech (SLTS) model shares several typical components with conventional LTS models, including a video encoder, a mel-spectrogram decoder, and a vocoder. The video encoder used in this work is a framewise 2D-ResNet18 [8], which produces a sequence of D_f-dimensional vectors $\boldsymbol{F} = (f_0, \ldots, f_{T_v-1})$ at 25 Hz, where $f_t \in \mathbb{R}^{D_f}$. The mel decoder comprises a Conformer [6] and a Conv1d-based postnet. Frame features $\boldsymbol{F}$, obtained at a rate of 25 Hz, are first concatenated with the speaker embedding and then fed into the Conformer to generate compact acoustic representations. These representations capture both local and global contexts, which are then linearly upsampled to 100 Hz and passed into the postnet to reconstruct mel-spectrograms $\hat{\boldsymbol{M}}$ at 100 Hz. The reconstructed mel-spectrograms are then fed into a UnivNet vocoder [10] to produce 16 kHz audio waveform. To effectively train the vocoder, we randomly segment pairs of generated mel spectrograms and reference audio waveforms into 0.6-second segments, since vocoders are typically trained with shorter audio segments.

3.4. AV offset predictor

To tackle asynchrony issues, we develop an automatic synchronization mechanism (ASM) and incorporate it into the lip-to-speech (LTS) model, forming our synchronized lip-to-speech (SLTS) model. The ASM consists of two modules: a data synchronization module (DSM) and a self-synchronization module (SSM). Both modules have its time-offset predictor, which parameterizes a categorical distribution on the audiovisual offsets within a predefined range. Before exploring the details of the two synchronization modules, we first introduce the proposed offset predictor in this section.

As detailed in Fig. 4, the offset predictor first extracts two normalized feature embedding sequences: the linearly upsampled video frame embedding sequence, $\boldsymbol{V}^{o_d} = \left(\boldsymbol{v}_0^{o_d}, \ldots, \boldsymbol{v}_{T_m-1}^{o_d}\right)$, and the mel-spectrogram embedding sequence, $\boldsymbol{U} = (\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{T_m-1})$ of length T_m, both at a frequency of 100 Hz. A cross-correlation sequence, $\boldsymbol{c} = (c_{-K}, \ldots, c_K)$, where $K \in \mathbb{N}^+$ signifies the preset range in terms of the number of frames of the mel spectrogram, is then calculated as follows:

$$c_k = \sum_{i=\max(k,0)}^{\min(k,0)+T_m-1} \langle \boldsymbol{v}_i^{o_d}, \boldsymbol{u}_{i-k} \rangle. \tag{1}$$

The computed cross-correlation sequence is then passed through a softmax function with a manually adjusted temperature parameter (set at $\tau = 0.1$), to generate the offset distribution:

$$P_\Theta(k|\boldsymbol{F}^{o_d}, \boldsymbol{M}) = \frac{\exp(c_k/\tau)}{\sum_{i=-K}^{K} \exp(c_i/\tau)}, \tag{2}$$

where Θ represents the parameters of the offset predictor. We use the same architecture for audio and video feature extractors, consisting of two Conv1D-BN-GELU blocks followed by a fully connected layer. The first Conv1D has a kernel size of 3, and the other has a kernel size of 1. This design decision confines the receptive field to maintain the time precision of the embeddings, while leveraging certain temporal context, and thereby enhancing the discriminability of the embeddings.

3.5. Data synchronization module (DSM)

The proposed data synchronization module (DSM) incorporates an offset predictor that parameterizes a categorical distribution on audiovisual offsets, represented as $P_{\Theta_d}(k \mid \boldsymbol{F}^{o_d}, \boldsymbol{M})$. It does so by utilizing the ground-truth mel-spectrogram, $\boldsymbol{M}$, and the offset video features, $\boldsymbol{F}^{o_d}$, along with a specific set of parameters from the DSM model, Θ_d. As illustrated in Fig. 5, the produced offset distribution is later used to rectify the reconstructed mel spectrogram prior to the calculation of frame-level losses (*e.g.*

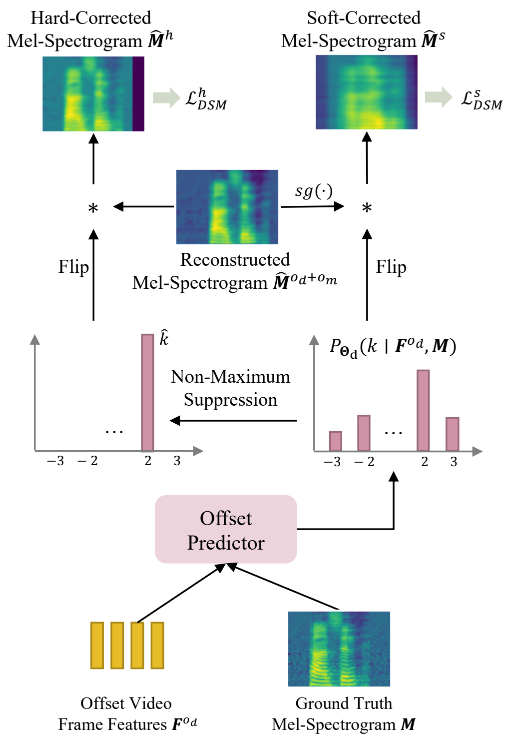


Figure 5: The data-synchronization module.

MSE and vocoder losses), through both soft and hard corrections.

First, we generate a soft-corrected mel-spectrogram, denoted as $\hat{\boldsymbol{M}}^s$. This process entails reversing the temporal direction of the offset distribution, followed by a convolution operation with the reconstructed mel-spectrogram: $\hat{\boldsymbol{M}}^{o_d+o_m} = \left(\hat{\boldsymbol{m}}_0^{o_d+o_m}, \ldots, \hat{\boldsymbol{m}}_{T_m-1}^{o_d+o_m}\right)$. As a result, we obtain a soft-corrected mel-spectrogram denoted as $\hat{\boldsymbol{M}}^s = \left(\hat{\boldsymbol{m}}_0^s, \ldots, \hat{\boldsymbol{m}}_{T_m-1}^s\right)$, where:

$$\hat{\boldsymbol{m}}_i^s = \sum_{k=\max(-K, i-T_m+1)}^{\min(K,i)} P_{\Theta_d}(-k \mid \boldsymbol{F}^{o_d}, \boldsymbol{M}) sg(\hat{\boldsymbol{m}}_{i-k}^{o_d+o_m}). \tag{3}$$

A soft-DSM loss is defined as:

$$\mathcal{L}_{\text{DSM}}^s := \|\boldsymbol{M} - \hat{\boldsymbol{M}}^s\|_2^2. \tag{4}$$

The loss aims to supervise the offset predictor to produce the most appropriate shift to correct the offset in the video, hence matching the generated audio with the target audio. In Eq. (3), the gradient stopping operation, denoted by $sg(\cdot)$, is essential to avoid potential optimization issues. The soft-corrected mel-spectrogram aggregates various offset proposals, some of which may be incorrect. Without the gradient stopping operation, these erroneous proposals could lead the decoder to attempt to learn from multiple incorrect targets simultaneously, thus hindering model convergence.

Given that the soft-DSM loss solely generates gradients for updating the offset predictor, and the decoder does not receive updates due to the gradient stopping operation, we also incorporate a hard-corrected reconstructed mel-spectrogram. This is used to supervise the decoder based on the most probable offset proposal. The hard-corrected mel-spectrogram $\hat{\boldsymbol{M}}^h = (\hat{\boldsymbol{m}}_0^h, \ldots, \hat{\boldsymbol{m}}_{T_m-1}^h)$ is computed by convolving the reconstructed mel-spectrogram with another correction convolution kernel that suppresses all offsets but the most probable one, as shown in the left branch in Fig. 5. This is equivalent to the following mel-spectrogram shifting operation:

$$\hat{\boldsymbol{m}}_i^h = \begin{cases} \hat{\boldsymbol{m}}_{i-\hat{k}}^{o_d+o_m}, & i \geq \hat{k} \\ \boldsymbol{0}, & i < \hat{k} \end{cases}, \tag{5}$$

where $\hat{k} = \arg\max_k P_{\Theta_d}(k \mid \boldsymbol{F}^{o_d}, \hat{\boldsymbol{M}})$. The out-of-bound frames, *i.e.*, $i < \hat{k}$, are set to zero and excluded from the loss computation. Note that there is no $sg(\cdot)$ operation applied to the mel-spectrogram this time as we want the loss to supervise the decoder. Similarly, we apply the MSE loss on the hard-corrected mel-spectrogram:

$$\mathcal{L}_{\text{DSM}}^h := \|\boldsymbol{M} - \hat{\boldsymbol{M}}^h\|_2^2, \tag{6}$$

which we name as the hard-DSM loss.

3.6. Self-synchronization module (SSM)

During our initial experiments, we observed that the model trained solely on DSM tended to generate mel-spectrograms with a large consistent shift, which was likely due to the model's asynchrony caused by the Conformer's extensive receptive field. To alleviate this, we propose the self-synchronization module (SSM). SSM includes an offset predictor parameterized by Θ_s to generate an offset distribution, $P_{\Theta_s}(k \mid \boldsymbol{F}^{o_d}, \hat{\boldsymbol{M}}^{o_d+o_m})$. Based on this offset distribution, we compute the SSM loss:

$$\mathcal{L}_{\text{SSM}} := -\log P_{\Theta_s}(k = 0 \mid \boldsymbol{F}^{o_d}, \hat{\boldsymbol{M}}^{o_d+o_m}). \tag{7}$$

This loss function aims to encourage similarity between the features of the generated audio and video at the same timestep, while penalizing similarity between features at different time steps.

3.7. End-to-end training of the SLTS model

So far, we have introduced several losses from the ASM, including the soft-DSM to train the offset predictor, the hard-DSM to guide the generation of the mel spectrogram, and the SSM loss to prevent the model from producing internal offsets. Since our model is an end-to-end system that aims to generate waveforms, the vocoder is also trained from scratch as part of the model. We supervise the vocoded waveform using the a spectral convergence loss $\mathcal{L}_{\text{sc}}$, a log-STFT magnitude loss $\mathcal{L}_{\text{mag}}$, and a negative STOI loss $\mathcal{L}_{\text{neg-stoi}}$. Moreover, we optionally adopt the GAN objectives by employing a multi-resolution spectrogram discriminator (MRSD) and a multi-period waveform discriminator (MPWD) to improve speech quality. Our final training loss is given by:

$$\mathcal{L} := \mathcal{L}_{\text{DSM}}^{h} + \mathcal{L}_{\text{DSM}}^{s} + \mathcal{L}_{\text{SSM}} + \mathcal{L}_{\text{voc}}, \tag{8}$$

where $\mathcal{L}_{\text{voc}} := \mathcal{L}_{\text{sc}} + \mathcal{L}_{\text{mag}} + \mathcal{L}_{\text{neg-stoi}} + \lambda_G \mathcal{L}_G$. When training with GAN objectives, we set $\lambda_G > 0$ and adopt an additional loss, $\mathcal{L}_D$, to update the discriminator. Details on vocoder training and GAN objectives (*i.e.*, $\mathcal{L}_{\text{sc}}$, $\mathcal{L}_{\text{mag}}$, $\mathcal{L}_G$, and $\mathcal{L}_D$) can be found in [10].

4. Experimental settings

In this section, we will cover the specifics of our experimental setup. We first discuss the datasets used for our study, then outline our evaluation metrics and the time alignment frontend, and finish with our model training details.

4.1. Datasets

GRID-4S and GRID-4S-Async are subsets of the audiovisual corpus GRID [4] that consist of four speakers: two males (*s1* and *s2*) and two females (*s4* and *s29*). These subsets are commonly used to evaluate lip-to-speech models [20, 13]. The corpus, recorded in a controlled laboratory environment, features a limited vocabulary and employs an artificial grammar. We follow the convention of dividing the 4,000 samples (approximately 3.3 hours) into 90% for training, 5% for validation, and 5% for testing (namely the 90-5-5 rule). We refer to the original dataset as GRID-4S, and then create another artificial dataset called GRID-4S-Async by adding random AV offsets uniformly sampled from -150 to 150 ms to each sample. This asynchronous artificial dataset was designed to demonstrate the robustness of the proposed SLTS model.

TCD-TIMIT-LS [7] is an audiovisual corpus produced under laboratory conditions using real English sentences and a larger vocabulary. The original TCD-TIMIT dataset was produced by three professionally trained lip speakers and 59 normal-speaking volunteers. Following the literature [20, 13], we only included the data from the three professionally trained lip speakers. The three-speaker subset consists of 1,131 samples, for a total of roughly 1.82 hours of data. We split this subset using the 90-5-5 rule.

Lip2Wav [20] is a large-scale audiovisual dataset collected from YouTube lecture videos. It contains a total of 16K samples and more than 120 hours of data, including five different speakers. We used the official data split for this dataset.

4.2. Evaluation metrics

PESQ [22] evaluates the perceptual quality of a generated speech compared to a clean reference speech. We follow [20, 13] to report the narrowband MOS-LQO score.

STOI [24] & ESTOI [11] predict the intelligibility of generated speech by measuring the correlation of short-time temporal envelopes with clean speech. These metrics assume that the audio signals are time-aligned.

MCD [16] measures speech quality by computing the differences between two sequences of mel cepstra, which are extracted from the generated audio and reference audio.

WER is a metric used to evaluate the word accuracy of the generated audio compared to its ground-truth transcription. Since LTS does not directly generate text, we use Whisper medium [21] to obtain transcriptions of the generated speech, with the resulting WER denoted as *w*-WER. Since Whisper is trained with general English sentences, it is not well suited for recognizing the artificial grammar used in GRID. Instead, we train an ad hoc Kaldi ASR model [19] on GRID-4S training data to recognize the generated audio, with the resulting WER denoted as *k*-WER.

4.3. Time alignment frontend

We apply the time alignment frontend introduced in Sec. 3.1 to alignment-sensitive metrics such as STOI, ESTOI, and MCD. This helps to achieve stable evaluation on datasets with severe data asynchrony. The results obtained with the frontend are denoted with the prefix *a*-.

4.4. Implementation details

To obtain the facial region of the videos for the three datasets, we use the S^{3}FD [29] face detector. Before face detection, the long videos in the Lip2Wav dataset are segmented into chunks with a maximum duration of 30 seconds, following the official Lip2Wav pre-processing pipeline.

We limit the length of the video clips to a maximum of three seconds using random clipping during training. To train our SLTS models, we use a batch size of 32 and the Adam optimizer [15] with linear warm-up and cosine annealing learning rates. We use 1k warm-up steps and a maximum learning rate of 5×10^{-4}. For the GRID and TCD-TIMIT models, we choose the conformer (S) architecture and set the offset predictor range to ± 150 ms. For the Lip2Wav models, we choose the conformer (M) architecture and set the offset predictor range to ± 300 ms. All SLTS models are trained for a maximum of 50k iterations, each taking around one day on an NVIDIA RTX 2080 Ti GPU. For comparison, we also train the state-of-the-art VCA-GAN [13] for a maximum of 70k iterations using the Adam optimizer with a fixed learning rate of 1×10^{-4}. To fit the model within the same amount of VRAM required by SLTS, we reduce its batch size to 24.

Dataset	Model	STOI ↑	ESTOI ↑	PESQ ↑	MCD ↓	*a*-STOI ↑	*a*-ESTOI ↑	*a*-PESQ ↑	*a*-MCD ↓	WER (%) ↓
GRID-4S-Async	VCA-GAN	0.325	0.056	1.811	**44.499**	0.700	0.492	1.793	30.328	14.67†
	SLTS w/o ASM	**0.348**	**0.095**	1.827	44.508	0.735	0.557	1.820	27.723	8.67†
	SLTS	0.337	0.079	**1.924**	45.152	**0.752**	**0.585**	**1.909**	**26.670**	**4.25**†
GRID-4S	VCA-GAN	0.688	0.500	1.917	29.720	0.732	0.552	1.910	28.437	8.50†
	SLTS w/o ASM	0.698	0.519	1.906	27.438	0.753	0.582	1.903	25.684	4.92†
	SLTS	**0.703**	**0.525**	**1.932**	**27.327**	**0.761**	**0.592**	**1.933**	**25.404**	**2.92**†
TCD-TIMIT-LS	VCA-GAN	0.577	0.398	1.373	33.450	0.593	0.412	1.376	33.175	79.96
	SLTS w/o ASM	**0.622**	**0.460**	**1.480**	**30.334**	0.650	0.496	**1.482**	29.667	50.40
	SLTS	0.606	0.445	**1.480**	30.818	**0.664**	**0.511**	1.480	**29.430**	**38.06**
Lip2Wav *chem*	VCA-GAN	0.543	0.364	1.363	37.827	0.659	0.477	1.365	34.600	48.20
	SLTS w/o ASM	**0.603**	**0.445**	1.478	**34.104**	0.736	0.578	1.481	30.291	33.03
	SLTS	0.215	0.049	**1.520**	49.481	**0.760**	**0.616**	**1.515**	**29.130**	**24.69**

Table 1: Comparison between VCA-GAN [13], SLTS without ASM during training, and SLTS, on the GRID-4S-Async, GRID-4S, TCD-TIMIT-LS, Lip2Wav *chem*. *a*- denotes metrics with the time alignment frontend. The WER indicated by † is *k*-WER, while the others are *w*-WER, as described in Sec. 4.2. The ground-truth texts used to compute WER come from the dataset by default, except for Lip2Wav as the dataset contains no transcriptions. For Lip2Wav, we obtained the ground truth transcription by applying Whisper on the ground-truth speech.

Unless otherwise stated, the results presented are from models with the best time-aligned STOI (*i.e*. *a*-STOI) on the validation set throughout the training. The *a*-STOI is computed after every 1k iterations for GRID-4S and TCD-TIMIT-LS and 5k iterations for Lip2Wav. By default, we do not apply GAN objectives, except for the SLTS model with GAN in Tab. 2.

5. Results and discussion

5.1. Effectiveness of synchronization training

We begin by demonstrating the robustness of our proposed SLTS model with the GRID-4S-Async dataset, as shown in Tab. 1. The GRID-4S-Async dataset exhibits a severe data asynchrony issue that causes vanilla time-alignment sensitive metrics, such as STOI, ESTOI, and MCD, to fail. In contrast, the metrics with the proposed time alignment front-end show more reasonable scores when considering the GRID-4S scores as a reference. According to the scores of time-alignment insensitive metrics, such as PESQ and *k*-WER, as well as aligned metrics, such as *a*-STOI, *a*-ESTOI, and *a*-MCD scores, our proposed SLTS model equipped with ASM achieves the best results. This shows the effectiveness of the proposed ASM model when dealing with a dataset with severe data asynchrony.

The results on multiple conventional datasets are also shown in Tab. 1, including two small-scale datasets with less data asynchrony (GRID-4S and TCD-TIMIT-LS) and a large-scale dataset with significant inherent data asynchrony (Lip2Wav *chem*). SLTS models outperform baselines (*i.e*., VCA-GAN and SLTS without ASM) according to time-aligned metrics, regardless of the severity of asynchrony in the datasets.

When comparing GRID-4S-Async and GRID-4S, it can be seen that asynchrony in the data set has a negative effect, since the results of GRID-4S-Async are generally worse than those of GRID-4S. However, when trained on a dataset with more severe asynchrony (*i.e*., GRID-4S-Async), SLTS with ASM achieves more significant performance gains compared to other baselines. This can also be observed when comparing GRID-4S and TCD-TIMIT-LS with Lip2Wav *chem*. The former two datasets exhibit less severe asynchrony, while the Lip2Wav *chem* dataset has a more serious issue. Notably, when employing SLTS, the Lip2Wav *chem* dataset shows a more significant performance improvement, particularly in aspects of intelligibility and content correctness, compared to the GRID-4S and TCD-TIMIT-LS datasets.

5.2. Limitations of vanilla metrics

As shown in Fig. 1, we observed that even a slight offset between the test and the reference audio can have a significant negative impact. Therefore, while models trained with ASM achieve better intelligibility, perceptual quality, and content correctness as measured by aligned metrics (*e.g*., a-STOI, a-ESTOI, a-MCD) and also alignment-insensitive metrics (*e.g*., PESQ and WER), they can score worse on vanilla STOI, ESTOI and MCD due to data asynchrony in the test set. This is demonstrated in Tab. 1. These results highlight the limitations of alignment-sensitive metrics, since even a better-performing model can produce lower scores without proper alignment.

5.3. Impact of GAN training on vocoder

As part of our study on building an end-to-end LTS model, we also tried to improve audio quality by incor-

Method	a-STOI ↑	a-ESTOI ↑	a-PESQ ↑	a-MCD ↓	w-WER (%) ↓	MOS (I) ↑	MOS (N) ↑
VCA-GAN	0.659	0.477	1.365	34.600	48.20	3.250 ± 0.225	2.042 ± 0.179
SLTS	**0.760**	**0.616**	**1.515**	**29.130**	**24.69**	3.633 ± 0.228	1.858 ± 0.171
SLTS w/ GAN	0.738	0.583	1.405	31.856	26.55	$\mathbf{4.483 \pm 0.139}$	$\mathbf{4.267 \pm 0.153}$
Real Voice	1.000	1.000	4.549	0.000	0.00	4.808 ± 0.100	4.975 ± 0.028

Table 2: Results on Lip2Wav *chem*. *w/ GAN*: vocoder trained with GAN objectives. MOS scores are listed with their 95% confidence interval computed from their t-distribution.

porating GAN objectives with discriminators (*i.e.*, MRSD and MPWD) for joint training of the vocoder as in [10]. It is observed that the application of GAN objectives during training led to a notable improvement in performance according to human evaluation. However, this improvement comes with a trade-off, as it results in a decline in objective evaluation metrics, despite proper handling of alignment.

In Table 2, we present the mean opinion scores (MOS) obtained from 12 volunteers who evaluated the intelligibility (I) and naturalness (N) of 10 randomly selected samples from the Lip2Wav *chem* test set. Each volunteer rated four versions (*i.e.*, VCA-GAN, SLTS, SLTS w/ GAN, and real voice) of the 10 samples. The incorporation of the discriminators led to a substantial increase in the MOS results, enhancing both intelligibility and naturalness. However, we noticed a decline in the objective scores with the GAN training objectives. For example, the inclusion of discriminators led to a reduction in the a-STOI from 0.760 to 0.738 on the Lip2Wav *chem* test set. We observed that this decrease in the objective scores also had an impact on the training set. On a training subset consisting of 200 samples, a drop in the a-STOI from 0.855 to 0.825 was observed.

We hypothesize that the lower objective scores are a consequence of the nonintrusive nature of GAN training. By including discriminators in the training process, the generated audio is encouraged to match the distribution of real audio, rather than strictly align with the corresponding target audio. As a result, lower scores on intrusive metrics may be observed.

5.4. Synchronization qualitative study

To qualitatively understand how ASM works, we present a concrete training example in Fig. 6. In this example, the generated audio precedes the reference one by 80 ms. DSM assigns a significant portion of the probability mass to offsets around -80 ms. Once the reconstructed mel-spectrogram is convolved with the hard-correction kernel, the resulting mel-spectrogram aligns precisely with the ground-truth mel-spectrogram, enabling a more accurate timestep-level loss computation between the reconstructed and reference mel-spectrograms.

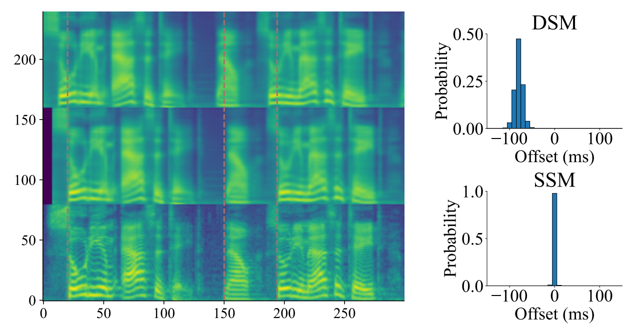


Figure 6: An example from Lip2Wav *chem*. On the left side, from top to bottom, are the reconstructed mel-spectrogram, hard-corrected reconstructed mel-spectrogram, and ground truth mel-spectrogram. On the right side, SSM concentrates on offset 0 as expected, and DSM accurately predicts the offset between the video and reference audio (*i.e.*, -80ms).

Method	STOI ↑	ESTOI ↑	PESQ ↑	MCD ↓
E2E-V2AResNet [23]	0.627	-	2.030	27.790
Yadav *et al.* [28]	0.724	0.540	1.932	-
VCA-GAN [13]	0.724	**0.609**	**2.008**	-
Lip2Wav [20]	0.731	0.535	1.722	-
Kim *et al.* [12, 9]	0.738	0.579	1.984	-
SLTS	**0.757**	0.588	1.931	**25.491**

Table 3: Comparison of SOTA results on GRID-4S.

Method	STOI ↑	ESTOI ↑	PESQ ↑	MCD ↓
E2E-V2AResNet [23]	0.472	-	**1.540**	36.190
Ephrat *et al.* [5]	0.487	0.310	1.231	-
GAN-based [25]	0.511	0.321	1.218	-
Lip2Wav [20]	0.558	0.365	1.350	-
VCA-GAN [13]	0.584	0.401	1.425	-
SLTS	**0.661**	**0.507**	1.474	**29.689**

Table 4: Comparison of SOTA results on TCD-TIMIT-LS.

5.5. Comparison with SOTA results

We compare our results with those of SOTA works reported in the literature with conventional non-aligned met-

Speaker	Method	STOI ↑	ESTOI ↑	PESQ ↑
chem	Lip2Wav [20]	0.416	0.284	1.300
	Hong *et al.* [9]	0.566	0.429	**1.529**
	SLTS	**0.757**	**0.612**	1.514
chess	Lip2Wav [20]	0.418	0.290	1.400
	Hong *et al.* [9]	0.506	0.334	1.503
	SLTS	**0.680**	**0.451**	**1.604**
dl	Lip2Wav [20]	0.282	0.183	**1.671**
	Hong *et al.* [9]	**0.576**	**0.402**	1.612
	SLTS	0.565	0.320	1.513
hs	Lip2Wav [20]	0.446	0.311	1.290
	Hong *et al.* [9]	0.504	0.337	1.366
	SLTS	**0.590**	**0.394**	**1.402**
eh	Lip2Wav [20]	0.369	0.220	1.367
	Hong *et al.* [9]	0.463	**0.304**	1.362
	SLTS	**0.482**	0.268	**1.428**

Table 5: Comparison of SOTA results on Lip2Wav. Speaker-specific models are trained for each speaker, following the convention.

rics. As SLTS models produce audio that is synchronized with the input video, severe data asynchrony in the test set can result in low scores with vanilla metrics (*e.g.*, *chem* in Tab. 1). To ensure fair comparisons, the results presented here are derived from the test set which has been synchronized using the offsets predicted by the DSM module. In GRID-4S Tab. 3, the SLTS model performs best in terms of STOI and MCD, with slightly lower scores in ESTOI and PESQ compared to the best result reported by [13]. In TCD-TIMIT-LS Tab. 4, except for the PESQ of [23], SLTS demonstrates the best scores in all metrics. For most speakers in Lip2Wav (*i.e.*, *chem*, *chess*, and *hs*), SLTS achieves much better intelligibility and comparable (or superior) perceptual quality. However, on *dl* and *eh*, SLTS performs similarly or slightly worse than [9]. We notice that videos from *dl* and *eh* have relatively smaller mouth regions, making the recognition of visemes difficult. This observation is consistent with the SyncNet results presented in Fig. 2c, which also show a lower confidence in its performance on *dl* and *eh*. In general, the performance of SLTS, based on various metrics, is either comparable to, or surpasses that of other state-of-the-art works.

6. Conclusion

In this work, we have identified two types of asynchronies that occur during lip-to-speech synthesis training: data asynchrony and model asynchrony. To address these asynchronies, we propose a synchronized lip-to-speech (SLTS) model. During training, the SLTS actively learns audiovisual time offsets to correct data asynchrony through a data synchronization module (DSM). The model synchronization is also ensured by using a self-synchronization module (SSM). In addition, we have introduced a time alignment frontend that separates the evaluation of synchronization and audio quality from conventional time-alignment sensitive metrics, such as STOI, ESTOI, and MCD. We have conducted extensive experiments using these new metrics to demonstrate the advantages of the proposed model. Our method achieves comparable or superior results across multiple tasks compared to existing state-of-the-art works.

7. Acknowledgements

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. T45-407/19N-1).

References

[1] Hassan Akbari, Himani Arora, Liangliang Cao, and Nima Mesgarani. Lip2AudSpec: Speech reconstruction from silent lip movements video. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2516–2520. IEEE, 2018. 2

[2] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005. 2

[3] J. S. Chung and A. Zisserman. Out of time: automated lip sync in the wild. In *Workshop on Multi-view Lip-reading, ACCV*, 2016. 2

[4] Martin Cooke, Jon Barker, Stuart Cunningham, and Xu Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424, 2006. 1, 2, 6

[5] Ariel Ephrat, Tavi Halperin, and Shmuel Peleg. Improved speech reconstruction from silent video. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 455–462, 2017. 8

[6] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020. 4

[7] Naomi Harte and Eoin Gillen. TCD-TIMIT: An audio-visual corpus of continuous speech. *IEEE Transactions on Multimedia*, 17(5):603–615, 2015. 1, 2, 6

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[9] Joanna Hong, Minsu Kim, Se Jin Park, and Yong Man Ro. Speech reconstruction with reminiscent sound via visual voice memory. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3654–3667, 2021. 2, 3, 8, 9

[10] Won Jang, Dan Lim, Jaesam Yoon, Bongwan Kim, and Juntae Kim. UnivNet: A neural vocoder with multi-resolution spectrogram discriminators for high-fidelity waveform generation. *arXiv preprint arXiv:2106.07889*, 2021. 3, 4, 6, 8

[11] Jesper Jensen and Cees H Taal. An algorithm for predicting the intelligibility of speech masked by modulated noise maskers. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):2009–2022, 2016. 1, 3, 6

[12] Minsu Kim, Joanna Hong, Se Jin Park, and Yong Man Ro. Multi-modality associative bridging through memory: Speech sound recollected from face video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 296–306, 2021. 3, 8

[13] Minsu Kim, Joanna Hong, and Yong Man Ro. Lip to speech synthesis with visual context attentional GAN. *Advances in Neural Information Processing Systems*, 34:2758–2770, 2021. 2, 3, 6, 7, 8, 9

[14] You Jin Kim, Hee Soo Heo, Soo-Whan Chung, and Bong-Jin Lee. End-to-end lip synchronisation based on pattern classification. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 598–605. IEEE, 2021. 2

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[16] Robert Kubichek. Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE pacific rim conference on communications computers and signal processing*, volume 1, pages 125–128. IEEE, 1993. 1, 3, 6

[17] Rodrigo Mira, Alexandros Haliassos, Stavros Petridis, Björn W Schuller, and Maja Pantic. SVTS: Scalable video-to-speech synthesis. *arXiv preprint arXiv:2205.02058*, 2022. 3

[18] Rodrigo Mira, Konstantinos Vougioukas, Pingchuan Ma, Stavros Petridis, Björn W Schuller, and Maja Pantic. End-to-end video-to-speech synthesis using generative adversarial networks. *IEEE Transactions on Cybernetics*, 2022. 2, 3

[19] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The Kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011. IEEE Catalog No.: CFP11SRW-USB. 6

[20] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. Learning individual speaking styles for accurate lip to speech synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13796–13805, 2020. 1, 2, 3, 6, 8, 9

[21] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. Technical report, OpenAI, 2022. 6

[22] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 749–752. IEEE, 2001. 6

[23] Nasir Saleem, Jiechao Gao, Muhammad Irfan, Elena Verdu, and Javier Parra Fuente. E2E-V2SResNet: Deep residual convolutional neural networks for end-to-end video driven speech synthesis. *Image and Vision Computing*, 119:104389, 2022. 8, 9

[24] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE international conference on acoustics, speech and signal processing*, pages 4214–4217. IEEE, 2010. 1, 3, 6

[25] Konstantinos Vougioukas, Pingchuan Ma, Stavros Petridis, and Maja Pantic. Video-driven speech reconstruction using generative adversarial networks. *arXiv preprint arXiv:1906.06301*, 2019. 8

[26] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Realistic speech-driven facial animation with GANs. *Int. J. Comput. Vis.*, 128(5):1398–1413, 2020. 1

[27] Yongqi Wang and Zhou Zhao. Fastlts: Non-autoregressive end-to-end unconstrained lip-to-speech synthesis. *arXiv preprint arXiv:2207.03800*, 2022. 2, 3

[28] Ravindra Yadav, Ashish Sardana, Vinay P Namboodiri, and Rajesh M Hegde. Speech prediction in silent videos using variational autoencoders. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7048–7052. IEEE, 2021. 2, 3, 8

[29] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. S^3FD: Single shot scale-invariant face detector. In *Proceedings of the IEEE international conference on computer vision*, pages 192–201, 2017. 6