

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 under the assumption that the audio-video pairs in the dataset are perfectly synchronized. In this work, we show that the commonly used audio-visual datasets, such as GRID, TCD-TIMIT, and Lip2Wav, can have data asynchrony issues. Training lip-to-speech with such datasets may further cause the model asynchrony issue — that is, the generated speech and the input video are out of sync. To address these asynchrony issues, we propose a synchronized lip-to-speech (SLTS) model with an automatic synchronization mechanism (ASM) to correct data asynchrony and penalize model asynchrony. We further demonstrate the limitation of the commonly adopted evaluation metrics for LTS with asynchronous test data and introduce an audio alignment frontend before the metrics sensitive to time alignment for better evaluation. We compare our method with state-of-the-art approaches on conventional and time-aligned metrics to show the benefits of synchronization training.

1. Introduction

Lip-to-speech (LTS) is the task of reconstructing the speech audio of a speaker based on the lip movement in a silent video. With the development of deep learning, many data-driven deep network models have been proposed to solve the LTS task.

A common assumption is made in training LTS models: the time offset between the corresponding video and audio data is a small constant, or zero. In other words, the audio and video of the same speech are fairly synchronized in time. However, after analyzing the synchronization errors using the lip-sync model SyncNet [4], we find that there exist varying time offsets between audios and videos in the audio-visual datasets that are commonly used for training and evaluating LTS models. Some datasets, such as GRID [5] and TCD-TIMIT [8] have small offsets within ± 1 video frames, but others, such as Lip2Wav [21], may

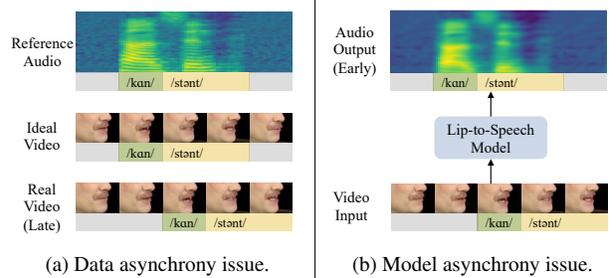


Figure 1. Illustration of the audio-visual asynchrony problem. The speaker is saying the word *constant*.

have larger offsets of multiple video frames. Moreover, large time offsets can also be introduced with careless data preprocessing (e.g. using FFmpeg [26] to segment a video file into smaller chunks¹). We call this *data asynchrony issue* (see Fig. 1a) since the synchronization error comes from the external dataset instead of the LTS model itself.

Although the synchronization errors are, most of the time, barely visible to the human naked eye, they can have a non-negligible impact on LTS model optimization. The training of LTS models usually utilizes time-wise learning objectives (e.g. MSE between the audio mel-spectrograms) that are sensitive to time offsets. The misalignments between videos and audios in the dataset can mislead the model to produce asynchronous output, resulting in the *model asynchrony issue* (see Fig. 1b). Besides, non-constant time offsets can cause training instability, making it difficult for the model to converge on large-scale datasets.

In the evaluation stage, the audio-visual asynchrony of the test dataset can make objective evaluation difficult as well. The commonly used objective speech intelligibility measures, such as STOI [25] and ESTOI [12], require the reference audio and the testing audio to be perfectly time-aligned to produce scores that precisely reflect the outcome of a listening test. When both model and data asynchrony

¹Such segmentation exists in the preprocessing pipeline of the Lip2Wav dataset: https://github.com/Rudrabha/Lip2Wav/blob/master/download_speaker.sh.

are present, misalignment between audio and video of the same speech can lead to inaccurate evaluation if they are not handled carefully.

In this work, we aim to solve the asynchrony issues in both the training and evaluation stage. For training, we introduce the synchronized lip-to-speech (SLTS) architecture, which consists of an automatic synchronization mechanism (ASM) that ensures the model and data synchronization in the training stage. Moreover, we propose an intrusive time-alignment frontend of the popular metrics during evaluation. The proposed frontend decouples the synchronization errors from conventional evaluation, ensuring reliable scoring despite data asynchrony in the test set.

In the experiment section, we perform extensive experiments on popular audio-visual datasets to show the effectiveness of the proposed automatic synchronization mechanism. The results show that the proposed synchronization method can handle both the long-term asynchrony that is visible to the human eye (*e.g.*, more than one video frame offsets) and subtle synchronization errors (*e.g.*, single-frame or sub-frame offsets). SLTS also outperforms existing SOTA models on various objective metrics and achieves high scores in the subjective listening test.

2. Related Works

2.1. Synchronization in Lip-to-Speech Models

Lip-to-speech models usually consist of components that provide a large temporal receptive field, such as 3D convolutional stacks [21], LSTM or GRU [1, 19, 21, 29], location sensitive attention [10, 21], and self-attention layers [14, 28]. The large receptive field potentially allows the model to generate offset audio. Kim *et al.* [14] point out that some existing LTS models do not explicitly process local visual features and may produce out-of-sync speech from the input video. They propose additional synchronization losses during training to handle the model asynchrony problem. However, their work only considers the model asynchrony but not the data asynchrony. Our work considers both types of asynchrony and proposes solutions to these issues.

2.2. Lip-Sync Models

The task of lip-sync aims to predict audio-visual offsets to correct lip-sync errors. Existing works, such as [4, 15], assume the audio-visual training data is synchronized and design different negative pairs to train the model with contrastive loss. Chung *et al.* [4] generate negative (off-sync) audio-video pairs by randomly shifting the audio and applies the contrastive loss from Siamese networks [3] to train their network. Kim *et al.* [15] instead adopt a softmax-based contrastive loss and treats the audio-visual features with different time steps as negative pairs. Our proposed data synchronization module (DSM) can also be used for lip-sync.

Compared to existing lip-sync models, DSM does not assume the training data to be synchronized. It processes a set of candidate pairs and discovers the positive and negative pairs in an unsupervised manner, driven by the lip-to-speech learning objective (*e.g.*, MSE loss between mel-spectrograms).

2.3. End-to-End Lip-to-Speech Models

Lip-to-speech models are often not designed to generate waveform end-to-end since more compact acoustic representations (*e.g.* mel-spectrogram) are usually sought to reduce the task difficulty. The compact acoustic representations are later converted to audio waveform by a vocoder, which can be either algorithm-based, such as Griffin-Lim used in [14, 21, 29], or a separately trained neural vocoder as in [10, 13, 18]. Building end-to-end LTS models [19, 28] that directly generate the audio waveform has recently attracted more attention as it produces speech with better quality than the algorithm-based vocoder and does not require separate training of a neural vocoder. In this work, we also investigate end-to-end modeling by jointly training a UnivNet vocoder [11] with the proposed model.

3. Synchronized Lip-to-Speech Synthesis

We first formulate the data and model asynchrony issues in Sec. 3.1, and then we describe the overall architecture of the proposed synchronized lip-to-speech (SLTS) model in Sec. 3.2. We will introduce our key contribution, the automatic synchronization mechanism (ASM), with a detailed description on its two components: the data synchronization module (DSM) and the self-synchronization module (SSM) in Sec. 3.3.

3.1. Problem Formulation

For simplicity, we first consider the silent lip video $x(t) \in \mathbb{R}^{H \times W \times 3}$ and the corresponding audio $y(t) \in \mathbb{R}$ as continuous functions of time $t \in \mathbb{R}$. There are two kinds of asynchrony issues that LTS task faces: the data asynchrony issue and the model asynchrony issue.

Ideally, a lip video $x(t)$ is expected to be accompanied by an audio $y(t)$ with a zero time offset. However, in real-world datasets, a lip video can have a non-constant time offset of o_d seconds from the audio (when $o_d > 0$, video lags behind the audio), resulting in an asynchronous video: $x^{o_d}(t) = x(t - o_d)$. We call this o_d -second data asynchrony issue as shown in Fig. 1a.

On the other hand, an LTS model can inject its own time offset to the reconstructed audio due to its exploitation of temporal context when the data used to train the model is off-sync. Given a video $x(t)$, the LTS model may instead reconstruct an audio $\hat{y}^{o_m}(t) = \hat{y}(t - o_m)$ with a time shift of o_m (seconds) from its ideal synchronized reconstruct-

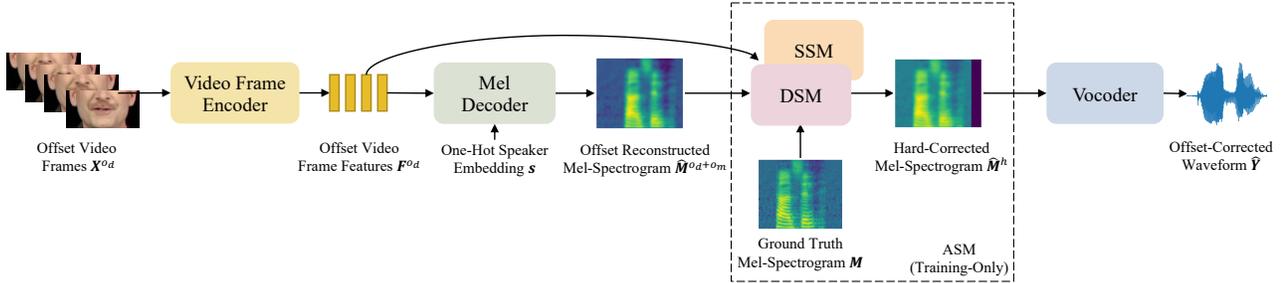


Figure 2. The overview of the proposed SLTS architecture. Where o_d is due to data asynchrony and o_m is due to model asynchrony; both are measured in seconds. The two asynchrony issues are handled by DSM and SSM respectively.

tion $\hat{y}(t)$. We call this the o_m -second model asynchrony as shown in Fig. 1b.

Usually, if there is no data asynchrony, there may not be model asynchrony neither as the synchronized reconstructed audio should be the optimal among other asynchronous proposals. On the other hand, data asynchrony will bring forth model asynchrony, especially when the audio-visual offsets vary from samples to samples.

3.2. Architecture Overview

Before delving into the solution to the asynchrony problems, we first introduce the architecture of our LTS model, which is shown in Fig. 2. In practice, the video and audio data are discrete signals in time. The silent RGB video data is represented by $\mathbf{X} \in \mathbb{R}^{T_v \times H \times W \times 3}$ where T_v, H, W are the number of video frames, frame height and width, respectively, and 3 is the number of color channels. The single-channel audio with T_a samples is represented by $\mathbf{Y} \in \mathbb{R}^{T_a \times 1}$. In our work, the video data have the frequency of 25 or 30 Hz depending on the dataset, and the audio frequency is fixed to 16 kHz.

The proposed synchronized lip-to-speech (SLTS) model aims to reconstruct an offset-corrected audio $\hat{\mathbf{Y}}$ from a given silent video \mathbf{X}^{o_d} which has an offset of o_d (seconds) during training to match the learning target (ground truth) audio \mathbf{Y} . During inference, the model can either produce an audio $\hat{\mathbf{Y}}^{o_d}$ that is aligned with the offset video \mathbf{X}^{o_d} without using ASM, or an ASM-corrected audio $\hat{\mathbf{Y}}$ aligned with the reference audio \mathbf{Y} . The latter is mainly used for evaluation which requires aligned audios.

SLTS consists of a video frame encoder, a decoder, two synchronization modules, namely DSM and SSM, and a vocoder. The frame encoder is based on ResNet18 [9], which produces D_f -dimensional features $\mathbf{F} \in \mathbb{R}^{T_v \times D_f}$ for each individual video frame. The decoder consists of a conformer [7] and a Conv1D-based post-net. The 25 Hz frame features \mathbf{F} are first concatenated with the speaker embedding and then sent to the conformer to generate compact acoustic representations based on local and global contexts. The compact acoustic representations are then linearly up-

sampled to 100 Hz and fed into the post-net to generate 100 Hz mel-spectrograms $\hat{\mathbf{M}}$.

Following the decoder is the ASM which consists of two modules: DSM and SSM, the key contributions of this work. The two modules learn and correct the data and model asynchrony respectively during training. DSM takes the 25 Hz video frame features \mathbf{F} , the ground-truth mel-spectrogram \mathbf{M} and the reconstructed mel-spectrogram $\hat{\mathbf{M}}$ as inputs to estimate the time offset for correcting the asynchrony in the audio-visual data. SSM, on the other hand, generates a self-synchronization loss based on the video frame features \mathbf{F} and the reconstructed mel-spectrogram $\hat{\mathbf{M}}$ to penalize the model asynchrony.

Finally, an UnivNet-based [11] vocoder is adopted to generate the audio waveform. Since vocoders are usually trained with audio segments shorter than 1.0 second, we perform 0.6-second random segmentation on pairs of offset-corrected mel-spectrogram and reference audio waveform. The vocoder is trained with the whole system using multi-resolution STFT loss and a differentiable STOI loss. We also allow the option of adopting the multi-resolution spectrogram discriminator (MRSD) and the multi-period waveform discriminator (MPWD) to further improve subjective speech quality at the cost of lower objective evaluation scores.

3.3. Automatic Synchronization Mechanism

The automatic synchronization mechanism consists of two components: DSM and SSM, both rely on a time offset predictor. We describe the time offset predictor first and then introduce the details of DSM and SSM.

3.3.1 Audio-visual Time Offset Predictor

The time offset predictor (shown in Fig. 3) generates a categorical distribution for the audio-visual offsets. The range of values in the categorical distribution of time offsets can be set manually, usually from 150–300 ms. Different from SyncNet [4] which predicts synchronization error in the number of video frames (at 25 Hz in our work), the pro-

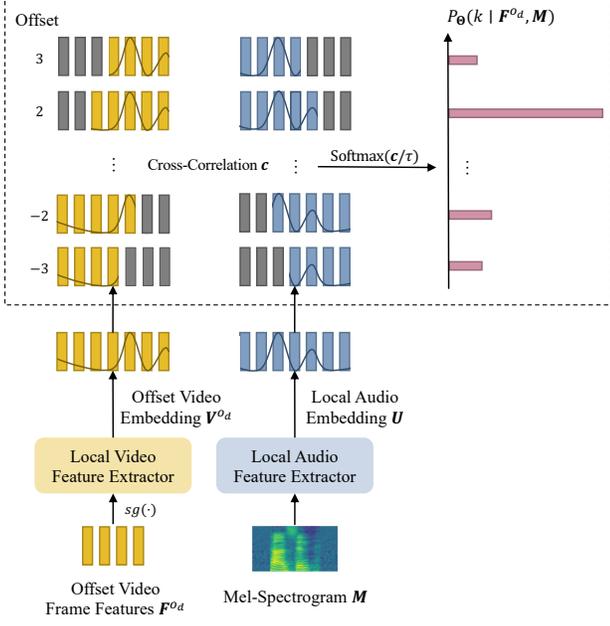


Figure 3. The audio-visual offset predictor.

posed offset predictor predicts offsets in the number of mel-spectrogram frames (at 100 Hz in our work) to achieve more precise synchronization.

The offset predictor contains two local feature extractors for video and audio, respectively. Each feature extractor contains two Conv1D-BN-GELU blocks, a fully-connected layer and an L2 normalization operation to generate normalized local embeddings. Only the first Conv1D has a kernel size of 3, whereas the other has a kernel size of 1. The receptive field is intentionally restricted to preserve the time precision of the embeddings, with a slight exploitation of temporal context to improve feature discriminability. The video feature extractor has an additional resampling operation that linearly upsamples the input video features from 25 Hz to 100 Hz before the first Conv1D, so as to match the sampling rate of mel-spectrograms.

After obtaining the sequence of local video embeddings $\mathbf{V}^{o_d} = (\mathbf{v}_0^{o_d}, \dots, \mathbf{v}_{T_m-1}^{o_d})$ and local audio embeddings $\mathbf{U} = (\mathbf{u}_0, \dots, \mathbf{u}_{T_m-1})$, cross-correlation $\mathbf{c} = (c_{-K}, \dots, c_K)$ between the two sequences of embeddings is computed for samples within a synchronization radius of $K \in \mathbb{N}^+$ (which is a hyper-parameter):

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

The cross-correlation is then normalized by the softmax function with a manually tuned temperature τ to produce the offset distribution:

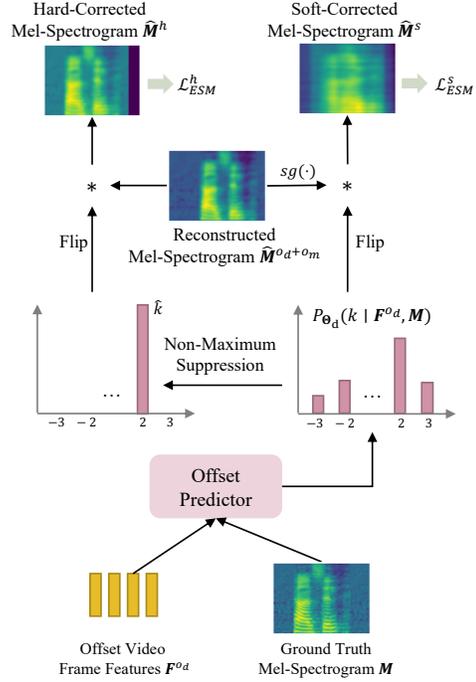


Figure 4. The data synchronization module.

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

where Θ is the parameters of the offset predictor (*i.e.*, the parameters of the local extractors).

3.3.2 Data Synchronization Module

As shown in Fig. 4, the data synchronization module (DSM) consists of an offset predictor that first generates a categorical distribution of the audio-visual offset, $P_{\Theta_d}(k | \mathbf{F}^{o_d}, \mathbf{M})$, based on the video features \mathbf{F}^{o_d} and the ground truth mel-spectrogram \mathbf{M} , with a set of DSM model parameters Θ_D .

The generated offset distribution is flipped along time to obtain a correction convolution kernel, which is used to correct the offset mel-spectrogram. A soft-corrected mel-spectrogram $\hat{\mathbf{M}}^s = (\hat{\mathbf{m}}_0^s, \dots, \hat{\mathbf{m}}_{T_m-1}^s)$ is produced by convolving the reconstructed mel-spectrogram $\hat{\mathbf{M}}^{o_d+o_m} = (\hat{\mathbf{m}}_0^{o_d+o_m}, \dots, \hat{\mathbf{m}}_{T_m-1}^{o_d+o_m})$ with the kernel:

$$\hat{\mathbf{m}}_i^s = \sum_{k=\max(-K, i-T+1)}^{\min(K, i)} P_{\Theta_e}(-k | \mathbf{F}^{o_d}, \mathbf{M}) sg(\hat{\mathbf{m}}_{i-k}^{o_d+o_m}), \quad (3)$$

where $sg(\cdot)$ is the gradient stopping operation. Then, a soft-DSM loss is computed between the ground-truth mel-

spectrogram and the soft-corrected mel-spectrogram:

$$\mathcal{L}_{DSM}^s(M, \hat{M}^s) := \|M - \hat{M}^s\|_2^2. \quad (4)$$

When generating the soft-corrected mel-spectrogram, the gradient stop operation on the reconstructed mel-spectrogram is critical. The soft-corrected mel-spectrogram is a combination of numerous offset proposals, which may include some wrong proposals. These wrong proposals may cause erroneous gradients backpropagated to the decoder, forcing it to learn several wrong targets at once, hence causing convergence problems.

To ensure that the decoder only learns from the most probable offset proposal, alongside the soft-corrected mel-spectrogram, a hard-corrected mel-spectrogram $\hat{M}^h = (\hat{m}_0^h, \dots, \hat{m}_{T-1}^h)$ is computed by convolving the reconstructed mel-spectrogram with another correction kernel that suppresses the less likely offsets, giving the following result:

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

where $\hat{k} = \arg \max_k P(k | F^{o_d}, \hat{M})$, and the out-of-bound frames $i < \hat{k}$ are set to zero and excluded in the loss computation. Similar to the soft-DSM loss, we adopt the MSE loss on the hard-corrected mel-spectrogram:

$$\mathcal{L}_{DSM}^h(M, \hat{M}^h) := \|M - \hat{M}^h\|_2^2. \quad (6)$$

Ideally, after the DSM is trained to convergence, a shift of $-\hat{k}$ frames on the reconstructed mel-spectrogram will correct the o_d -second data asynchrony.

3.3.3 Self-Synchronization Module

Besides the o_d -second data asynchrony, there can also be o_m -second model asynchrony due to the large receptive field of the audio decoder. We introduce a self-synchronization module (SSM) that tries to minimize the potential model asynchrony.

Similar to the DSM, SSM also contains an independent offset predictor parameterized by Θ_S to generate an offset distribution, $P_{\Theta_S}(k | F^{o_d}, \hat{M}^{o_d+o_m})$. Unlike DSM, SSM focuses on reducing the offsets between the video features F^{o_d} and the reconstructed mel-spectrogram $\hat{M}^{o_d+o_m}$ by minimizing the following SSM loss:

$$\mathcal{L}_{SSM}(F^{o_d}, \hat{M}^{o_d+o_m}) := -\log P_{\Theta_S}(k = 0 | F^{o_d}, \hat{M}^{o_d+o_m}). \quad (7)$$

Empirically, we find that SSM improves the training stability of DSM. Without SSM, DSM sometimes does not learn effective offsets, and the predicted offset collapses to a constant. We hypothesize that this may be attributed to the propagation of noisy gradients from the otherwise uncontrolled soft DSM loss.

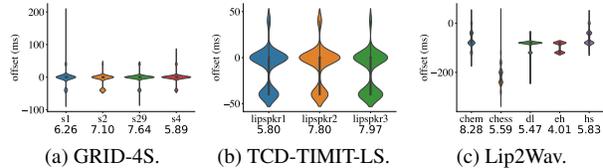


Figure 5. Offsets produced by SyncNet. The figures under speaker names are confidence scores produced by SyncNet. A higher score means SyncNet has greater confidence in its outputs. Only the offsets having confidence greater than 3.0 are counted.

Dataset	Speakers	Train / Val / Test Samples	Train / Val / Test Hours
GRID-4S [5]	4	3,600 / 200 / 200	2.98 / 0.17 / 0.17
TCD-TIMIT-LS [8]	3	1,017 / 57 / 57	1.64 / 0.09 / 0.09
Lip2Wav [21]	5	15,894 / 376 / 487	115.16 / 2.48 / 3.37

Table 1. Statistics for dataset splits used in our experiments. All speakers from the same dataset are present in all training, validation, and test splits.

4. Datasets, Metrics and Training Details

4.1. Datasets Overview

GRID-4S is a four-speaker subset of the GRID audio-visual corpus [5]. The subset consists of two male speakers ($s1, s2$) and two female speakers ($s4, s29$), and is commonly used in the literature [14, 21] to evaluate lip-to-speech models. The corpus is recorded in the laboratory condition. It has a small vocabulary and an artificial grammar.

TCD-TIMIT-LS [8] is another audio-visual corpus produced in the laboratory condition using real English sentences with a larger vocabulary. The original TCD-TIMIT dataset is produced by three professionally-trained lip speakers and 59 normal-speaking volunteers. Following the literature [14, 21], we adopt only the data from the three professionally-trained lip speakers.

Lip2Wav [21] is a large-scale audio-visual dataset collected from YouTube lecture videos. The dataset includes five different speakers, all of whom are used in our experiments.

4.1.1 Data Preparation

For GRID-4S and TCD-TIMIT-LS datasets, we follow the convention [17, 21, 27] and randomly select 90% of the data samples from each speaker for training, 5% for validation, and 5% for testing. For Lip2Wav, we adopt the official data split². We adopt S³FD [30] face detector to obtain the facial region of the videos for all three datasets. Before face detection, the long videos in the Lip2Wav datasets are segmented into chunks with a maximum duration of 30 seconds, fol-

²Official Lip2Wav splits: <https://github.com/Rudrabha/Lip2Wav/tree/master/Dataset>.

Dataset	Model	STOI \uparrow	ESTOI \uparrow	PESQ \uparrow	MCD \downarrow	<i>a</i> -STOI \uparrow	<i>a</i> -ESTOI \uparrow	<i>a</i> -PESQ \uparrow	<i>a</i> -MCD \downarrow	Offset- R^2 (F) \uparrow	Offset- R^2 (SN) \uparrow	<i>w</i> -WER (%) \downarrow	<i>k</i> -WER (%) \downarrow
GRID-4S	VCA-GAN	0.688	0.500	1.917	29.720	0.732	0.552	1.910	28.437	-0.002 [†]	-0.030 [†]	23.67	8.50
	SLTS w/o ASM	0.698	0.519	1.906	27.438	0.753	0.582	1.903	25.684	-0.001 [†]	-0.030 [†]	15.33	4.92
	SLTS	0.703	0.525	1.932	27.327	0.761	0.592	1.933	25.404	0.862	0.383	12.83	2.92
TCD-TIMIT-LS	VCA-GAN	0.577	0.398	1.373	33.450	0.593	0.412	1.376	33.175	-0.038 [†]	-0.164 [†]	79.96	-
	SLTS w/o ASM	0.622	0.460	1.480	30.334	0.650	0.496	1.482	29.667	-0.585 [†]	-0.164 [†]	50.40	-
	SLTS	0.606	0.445	1.480	30.818	0.664	0.511	1.480	29.430	0.796	0.525	38.06	-
Lip2Wav <i>chem</i>	VCA-GAN	0.543	0.364	1.363	37.827	0.659	0.477	1.365	34.600	-0.000 [†]	-2.725 [†]	48.20	-
	SLTS w/o ASM	0.603	0.445	1.478	34.104	0.736	0.578	1.481	30.291	-0.005 [†]	-2.725 [†]	33.03	-
	SLTS	0.215	0.049	1.520	49.481	0.760	0.616	1.515	29.130	0.982	0.704	24.69	-

Table 2. Comparison between VCA-GAN [14], SLTS without ASM during training, and SLTS. *a*-: metrics with the time alignment frontend. [†]: results computed with a dummy offset predictor (*i.e.*, always predicts 0). By default, the reference text used to compute WER is from the dataset, except for Lip2Wav, where the reference text is obtained by applying the Whisper ASR on the reference speech.

lowing the official Lip2Wav preprocessing pipeline. The statistics of the preprocessed datasets are shown in Tab. 1.

4.1.2 Asynchrony Analysis

To study the asynchrony in the datasets, we use a pretrained SyncNet³ [4] to estimate the degree of asynchrony on the three audio-visual datasets. SyncNet takes a 25 FPS video and 16 kHz audio as inputs and produces the audio-visual offsets with a resolution of 40 ms. The statistics of the SyncNet results are shown in Fig. 5. Except for a few outliers, the audio-visual offsets of the GRID-4S and TCD-TIMIT-LS data samples center around 0 ms, with some slightly off-sync by one video frame (*i.e.*, ± 40 ms). In the Lip2Wav dataset, offsets of the *chess* speaker center around -200~-250 ms, while offsets from other speakers center around -80 ms. The audio lag of Lip2Wav data is mainly caused by video segmentation during data preprocessing, except for those from the *chess* speaker, whose original videos are generally ahead of time.

4.2. Evaluation Metrics Overview

PESQ [23]: evaluates the perceptual quality of a generated speech compared to a clean reference speech. We follow [14, 21] to report the narrow-band MOS-LQO score of PESQ.

STOI [25] & **ESTOI** [12]: predicts the results of intelligibility listening tests based on the correlation of the short-time temporal envelopes between the generated and clean speech. Both metrics assume that the audios are time-aligned.

MCD: is another alignment-sensitive metric that measures the differences between two sequences of mel cepstra extracted from the generated audio and reference audio.

WER: counts the word errors in the transcriptions of the generated audios. We use the medium version of Whisper [22] to obtain the transcriptions, and the WER computed from them is denoted as *w*-WER. GRID-4S utterances are

³Implementation and model checkpoint of SyncNet obtained here: https://github.com/joonson/syncnet_python.

generated by an artificial grammar with a constrained vocabulary. Whisper, however, is a general large-vocabulary speech recognizer. It produces a lot of homophones (*e.g.*, ‘red’ \rightarrow ‘read’, ‘blue’ \rightarrow ‘blew’) on GRID-4S which are counted as errors, resulting in inaccurate evaluation. Thus, we train an ad-hoc Kaldi ASR model [20] with the GRID-4S training set to recognize the generated GRID-4S audios, and denote the resulting WER as *k*-WER.

Offset- R^2 : is the coefficient of determination between the offsets produced by DSM and another approach, such as the metrics frontend (see Sec. 4.3). It is denoted as Offset- R^2 (F) and Offset- R^2 (SN) if the other approach is the metrics frontend and SyncNet, respectively.

4.3. Time Alignment Metric Frontend

Alignment-sensitive metrics, such as STOI, ESTOI, and MCD, can produce inaccurate scores when the two input audios are not time-aligned (details discussed in Sec. 5.2). We propose a time alignment frontend to address the issue by first computing mel-spectrograms from both the generated and reference audios with a window size of 640 and a hop length of 160, and then normalizing them along the channel dimension. Sixty-one alignment proposals are then created by shifting the generated audio from -300 ms to 300 ms with a step size of 10 ms. The shift that produces the minimum mean squared error between the two normalized mel-spectrograms is selected to correct the generated audio before scoring. The negative of the optimal shift is called the front-end offset, denoted as o_f .

4.4. Implementation and Training Details

We limit the video clip length to a maximum of 3 seconds via random chunking and adopt a batch size of 32 to train our SLTS models. Adam optimizer [16] with a linear warm-up and cosine annealing learning rate is adopted, where the number of warm-up steps is 1k and the maximum learning rate of 5×10^{-4} . We choose conformer (S) for GRID and TCD-TIMIT models and conformer (M) for Lip2Wav models. All SLTS models are trained for a maximum of 50k iterations (each taking around 1 day on an RTX 2080-Ti). To fit the model into the VRAM, we adopt

Offset	STOI \uparrow	ESTOI \uparrow	MCD \downarrow
0 ms	1.000	1.000	0.000
4 ms	0.916	0.869	11.651
8 ms	0.770	0.708	15.605
12 ms	0.660	0.594	18.712

Table 3. The impact of offsets on the alignment-sensitive metrics. The results are computed on two copies of the same audio (bbaf2n from GRID-4S) with one shifted to simulate asynchrony.

the gradient checkpointing [2] on the frame encoder to reduce VRAM consumption. For comparison, we also train the SOTA VCA-GAN model [14] for a maximum number of 70k iterations with the Adam optimizer and a fixed learning rate of 1×10^{-4} . A smaller batch size of 24 is adopted due to the larger memory consumption of the model.

5. Experimental Results and Discussion

Unless otherwise stated, the reported results are obtained from models with the best time-aligned STOI scores on the validation set throughout training. The time-aligned STOI is computed after every 1k iterations for GRID-4S and TCD-TIMIT-LS, and 5k iterations for Lip2Wav. The results are computed on the original test set without applying an additional lip-sync method by default.

5.1. Effectiveness of Synchronization Training

Regardless of the severity of the asynchrony problems in the datasets, SLTS models score higher than its non-synchronized competing models (*i.e.*, VCA-GAN and SLTS without ASM) according to the time-aligned metrics (see Tab. 2). The results show that the synchronization training benefits the speech intelligibility, perceptual quality, and mel cepstra similarity of the reconstructed audios when appropriately evaluated. Moreover, the content correctness of the reconstructed audio is also improved with synchronization training, measured by WER. Compared to the GRID-4S and TCD-TIMIT-LS datasets, the Lip2Wav *chem* dataset, which has a more severe data asynchrony issue, achieves a more significant performance gain, especially on intelligibility and content correctness. This suggests that the severe asynchrony in the dataset does not only produce off-sync generated audios but also the quality of the generated speech.

5.2. Limitation of Non-aligned Metrics

Though models trained with the ASM achieve better intelligibility, perceptual quality, and content correctness as measured by the alignment insensitive metrics (*e.g.*, PESQ and WER) and the metrics with alignment frontend, they may not consistently score better on vanilla STOI, ESTOI, and MCD. For these metrics, we notice that a slight offset between the testing and reference audios can have a large

Method	a -STOI \uparrow	a -ESTOI \uparrow	a -PESQ \uparrow	a -MCD \downarrow	w-WER (%)	MOS (I) $\downarrow \uparrow$	MOS (N) \uparrow
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/ <i>dis</i>	0.738	0.583	1.405	31.856	26.55	4.483 ± 0.139	4.267 ± 0.153
Real Voice	1.000	1.000	4.549	0.000	0.00	4.808 ± 0.100	4.975 ± 0.028

Table 4. Results on Lip2Wav *chem. w/ dis*: trained with discriminators to generate audio waveforms. MOS scores are listed with their 95% confidence interval computed from their t-distribution.

negative impact (see Tab. 3). The problem is that our SLTS model is trained to correct data asynchrony in the training data, when it is used for testing, if the reference test audio and test video are off-sync, the audio reconstructed by our SLTS model from the test video (perhaps with a perfect zero offset) will also be off-sync with the reference test audio. As a result, the SLTS models score lower on STOI, ESTOI and MCD. This shows the limitation of the alignment-sensitive metrics. Without proper alignment, lower scores can be produced even for a better performing model.

5.3. Accuracy of the Data Synchronization Module

Since there are no available ground truths for the audio-visual offsets in the test set, we evaluate the accuracy of DSM by comparing the offsets produced by different approaches. R^2 scores between the offsets predicted by DSM and the metrics frontend or SyncNet are also shown in Tab. 2. The high R^2 scores between the offsets predicted by DSM (*i.e.*, \hat{o}_d) and the metrics frontend (*i.e.*, $o_f = o_d + o_m, o_m \approx 0$) show that DSM can predict the data asynchrony o_d accurately. On the other hand, the offsets produced by DSM can explain more variance of the SyncNet offsets than a dummy offset predictor that assumes all offsets to be 0. On the datasets with a more severe data asynchrony issue (*e.g.*, Lip2Wav *chem*), the R^2 score becomes more prominent due to the high total variance of the audio-visual offsets.

5.4. Impact of Discriminators on Vocoder

To demonstrate the superiority of the model trained with discriminators (*i.e.*, MRSD and MPWD) on audio generation, we conduct mean opinion score (MOS) tests by asking 12 volunteers to score 10 samples randomly selected from the Lip2Wav *chem* test set. Both intelligibility and naturalness are assessed. Each volunteer rates four versions (*i.e.*, VCA-GAN, SLTS, SLTS w/ *dis*, and real voice) of the 10 samples. Results in Tab. 4 show significant MOS gains on both intelligibility and naturalness after adopting the discriminators. However, we notice that the objective scores are lower on both training and test samples after adopting the discriminators. For instance, after including the discriminators, the a -STOI score drops from 0.760 to 0.738 on the test set of Lip2Wav *chem*, and from 0.855 to 0.825 on a training subset of 200 samples. We hypothesize that the lower objective scores are caused by the non-

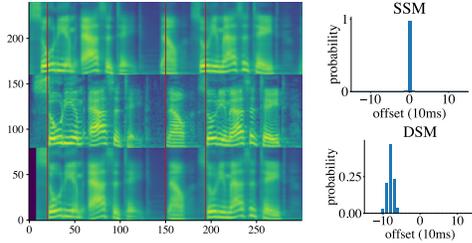


Figure 6. An example from Lip2Wav *chem* showing how ASM works. The left side shows the reconstructed, ground-truth and hard-corrected mel-spectrograms from top to bottom.

intrusive nature of GAN training. The discriminators encourage the generated audios to match a distribution of real audios rather than the corresponding target audios, rendering it harder to meet the intrusive learning objectives, such as MSE, multi-resolution STFT, and STOI, and resulting in lower scores on intrusive metrics.

5.5. Qualitative Study

To demonstrate how ASM works, we show a real training example in Fig. 6. In this example, the reconstructed audio is earlier than the reference mel-spectrogram by 80 ms. DSM assigns most of the probability mass to the offsets around -80 ms. After the reconstructed mel-spectrogram is convolved with the hard-correction kernel, the resulting mel-spectrogram precisely aligns with the ground-truth mel-spectrogram, allowing more accurate loss computation between the reconstructed and reference mel-spectrograms.

5.6. Comparison with Other SOTA Results

Tables 5 to 7 compare our results with SOTA results reported in existing work. Since SLTS models produce audio synchronized with the input video, they can have low scores on the vanilla metrics when data asynchrony in the test set is severe (e.g., the *chem*, as shown in Tab. 2). For reasonable comparisons, we report the results on the test set that is lip-synced by the DSM of the corresponding SLTS model. SLTS achieves similar or superior results compared to other SOTA works. On GRID-4S, SLTS has the best STOI and MCD, and outperforms other methods on all metrics except for the PESQ of work [24] on TCD-TIMIT-LS. For the majority of the speakers in Lip2Wav (i.e., *chem*, *chess* and *hs*), SLTS achieves much better intelligibility, and comparable (or superior) perceptual quality. On *dl* and *eh*, SLTS performs similarly or slightly worse than the SOTA work [10]. We notice that videos from *dl* and *eh* have relatively smaller mouth regions, making recognition of visemes difficult. This agrees with the SyncNet results (Fig. 5c) which also has lower confidence in its performance on *dl* and *eh*.

Method	STOI \uparrow	ESTOI \uparrow	PESQ \uparrow	MCD \downarrow
E2E-V2AResNet [24]	0.627	-	2.030	27.790
Yadav <i>et al.</i> [29]	0.724	0.540	1.932	-
VCA-GAN [14]	0.724	0.609	2.008	-
Lip2Wav [21]	0.731	0.535	1.722	-
Kim <i>et al.</i> [10, 13]	0.738	0.579	1.984	-
SLTS	0.757	0.588	1.931	25.491

Table 5. Comparison between SOTA results on GRID-4S dataset.

Method	STOI \uparrow	ESTOI \uparrow	PESQ \uparrow	MCD \downarrow
E2E-V2AResNet [24]	0.472	-	1.540	36.190
Ephrat <i>et al.</i> [6]	0.487	0.310	1.231	-
GAN-based [27]	0.511	0.321	1.218	-
Lip2Wav [21]	0.558	0.365	1.350	-
VCA-GAN [14]	0.584	0.401	1.425	-
SLTS	0.661	0.507	1.474	29.689

Table 6. Comparison with SOTA results on TCD-TIMIT-LS.

Speaker	Method	STOI \uparrow	ESTOI \uparrow	PESQ \uparrow
<i>chem</i>	Lip2Wav [21]	0.416	0.284	1.300
	Hong <i>et al.</i> [10]	0.566	0.429	1.529
	SLTS	0.757	0.612	1.514
<i>chess</i>	Lip2Wav [21]	0.418	0.290	1.400
	Hong <i>et al.</i> [10]	0.506	0.334	1.503
	SLTS	0.680	0.451	1.604
<i>dl</i>	Lip2Wav [21]	0.282	0.183	1.671
	Hong <i>et al.</i> [10]	0.576	0.402	1.612
	SLTS	0.565	0.320	1.513
<i>hs</i>	Lip2Wav [21]	0.446	0.311	1.290
	Hong <i>et al.</i> [10]	0.504	0.337	1.366
	SLTS	0.590	0.394	1.402
<i>eh</i>	Lip2Wav [21]	0.369	0.220	1.367
	Hong <i>et al.</i> [10]	0.463	0.304	1.362
	SLTS	0.482	0.268	1.428

Table 7. Comparison between the SOTA work and the proposed model on the Lip2Wav dataset. Unlike other datasets, we follow convention to train speaker-specific models for each speaker.

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 model (SLTS). During training, the SLTS actively learns audio-visual time offsets to correct data asynchrony via DSM. The model synchronization is also ensured using 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.

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] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021. 7
- [3] 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
- [4] J. S. Chung and A. Zisserman. Out of time: automated lip sync in the wild. In *Workshop on Multi-view Lip-reading, ACCV*, 2016. 1, 2, 3, 6
- [5] 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, 5
- [6] 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
- [7] 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. 3
- [8] Naomi Harte and Eoin Gillen. TCD-TIMIT: An audio-visual corpus of continuous speech. *IEEE Transactions on Multimedia*, 17(5):603–615, 2015. 1, 5
- [9] 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. 3
- [10] 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, 8
- [11] 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. 2, 3
- [12] 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, 6
- [13] 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. 2, 8
- [14] 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, 5, 6, 7, 8
- [15] 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
- [16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [17] Daniel Michelsanti, Olga Slizovskaia, Gloria Haro, Emilia Gómez, Zheng-Hua Tan, and Jesper Jensen. Vocoder-based speech synthesis from silent videos, 2020. 5
- [18] Rodrigo Mira, Alexandros Haliassos, Stavros Petridis, Björn W Schuller, and Maja Pantic. SVTS: Scalable video-to-speech synthesis. *arXiv preprint arXiv:2205.02058*, 2022. 2
- [19] 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
- [20] 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
- [21] 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, 5, 6, 8
- [22] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. Technical report, Tech. Rep., Technical report, OpenAI, 2022. 6
- [23] 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
- [24] Nasir Saleem, Jiechao Gao, Muhammad Irfan, Elena Verdu, and Javier Parra Fuente. E2E-V2SRsNet: Deep residual convolutional neural networks for end-to-end video driven speech synthesis. *Image and Vision Computing*, 119:104389, 2022. 8
- [25] 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, 6
- [26] Suramya Tomar. Converting video formats with ffmpeg. *Linux Journal*, 2006(146):10, 2006. 1
- [27] Konstantinos Vougioukas, Pingchuan Ma, Stavros Petridis, and Maja Pantic. Video-driven speech reconstruction

- using generative adversarial networks. *arXiv preprint arXiv:1906.06301*, 2019. [5](#), [8](#)
- [28] Yongqi Wang and Zhou Zhao. Fastlts: Non-autoregressive end-to-end unconstrained lip-to-speech synthesis. *arXiv preprint arXiv:2207.03800*, 2022. [2](#)
- [29] Ravindra Yadav, Ashish Sardana, Vinay P Nambodiri, 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](#), [8](#)
- [30] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. S³FD: Single shot scale-invariant face detector. In *Proceedings of the IEEE international conference on computer vision*, pages 192–201, 2017. [5](#)