A. Video samples

A webpage showing the video samples from EasyCom and from the four tasks on LRS3 (video-to-speech, audio-visual speech inpainting, audio-visual speech denoising, and audio-visuak source separation) are included in the supplementary material. Readers are highly recommended to watch the samples to better understand how the proposed model perform compared to the baselines.

B. EasyCom Dataset

Easy Communications (EasyCom) dataset [13] is designed to study the cocktail party problem in the conversational augmented reality setup. The dataset contains twelve 30 minute conversations, amounting to about six hours of raw recording. In each session, one participant wears the glasses that record ego-centric video and 6-channel audio clips. Other participants wear a single-channel close-talking microphone. After keeping only speech segments and removing utterances without the target speaker being visible throughout the entire segment (using the official time-aligned transcripts and lip bounding boxes), there are 1.64/0.59/0.35 hours remained for the train/valid/test splits.[4] The recordings from EasyCom and very different and much more challenging than those in LRS3. In EasyCom, there are motion blur in the video due to the head movement and barrel distortion from the camera. The distant microphone recordings often have interfering speech that is much louder than the target speech, coming from the subject who wears the glasses. The clips are recorded indoor with noise played by multiple speakers in the background. Hence, the microphone array on the glasses records substantial amount of noise and reverberation.

We use the close-talking microphone as the clean reference speech x_a and distant one as the noisy speech $\tilde{x}_a$ following [13]. As a standard practice in multichannel speech processing, we consider beamformed audio that is derived from the four non-binaural channels as input [5, 13, 18]. We use a maximally directive beamformer formulation that is optimized using a minimum-variance distortionless-response algorithm with a diffuse noise covariance and anechoic steering vector. The beamformer is steered towards the target's head location and the filter and sum is performed using weighted overlap add. We follow the exact formulation and implementation in [13]. We use a 64ms length FFT and filter length at 16kHz sampling rate, with an analysis and synthesis hanning window at 50% overlap. The diffuse noise covariance and target steering vector are obtained using the set of array transfer functions (ATFs) provided in the dataset and the distortionless response reference is microphone number two.

[4]Session 4 and 12 are used for validation, and 10 and 11 for testing.

For noisy speech $\tilde{x}_a$ used in training, we merge all six channels, beamformed audio as well as the audio from close-talking microphone to train ReVISE. Combining the multiple views of "same" speech can be regarded as one form of data augmentation, which has empirically improve performance a lot (see Section C.1). For evaluation, we test our model on both channel two (i.e., single-channel) and beamformed audio (i.e., multi-channel).

C. Additional Results

C.1. EasyCom visual features and training data

Table 10 shows the impact of fine-tuning data and visual inputs on the model performance in EasyCom.

Data For our main results, the pre-trained ReVISE model is fine-tuned on EasyCom only (1.6 hours). Though merging other audio-visual datasets and EasyCom can increase the size of training data by orders of magnitude, we do not observe gain brought by such practice (row (a) vs. row (b) in Table 10). This is potentially due to the severe domain mismatch in two datasets on multiple aspects such as types of audio (rehearsed speech vs. conversation) and noise (simulated vs. natural).

Input We also notice that using mouth regions as input is more effective in enhancing speech compared to directly feeding talking head into the model (row (a) vs. row (c) in Table 10). Mouth cropping helps remove unrelated visual background, thus bridging gap in visual domain between pretraining and fine-tuning.

Input	Data	WER (ch2)	WER (bf)
(a). Mouth	EasyCom	**50.3**%	**47.6**%
(b). Mouth	EasyCom+LRS3	54.1%	49.3%
(c). Head	EasyCom	56.2%	51.4%

Table 10. Impact of visual input and fine-tuning data on speech enhancement in EasyCom. Numbers are on development set.

C.2. Predicting units versus spectrogram

To perform video-to-speech synthesis, ReVISE differs from SVTS [36] in two main aspects. First, we predict SSL units while SVTS predicts Mel spectrogram. Second, we initialize the video-to-unit prediction module (P-AVSR) with AV-HuBERT, a self-supervised audio-visual speech model while SVTS trains the video-to-spectrogram model from scratch.

Tab. 11 studies how these two factors together contribute to the superior performance observed from ReVISE. "No-PT" and "PT" indicates where pre-trained weights of AV-HuBERT is loaded, and "Unit" and "Spec" indicates the prediction target of the P-AVSR module. We also train a

vocoder using LJSpeech to convert spectrogram to waveform for models that predict spectrograms. In terms of intelligibility, both pre-training and predicting units improve the performance, and pre-training is particularly important, similar to what prior studies observed on speech recognition [49]. Figure 3 shows the spectrograms of the three generated audios. Directly using spectrogram (Spec,PT) as prediction target leads to generation of more blurry speech compared to using units (Unit, PT).

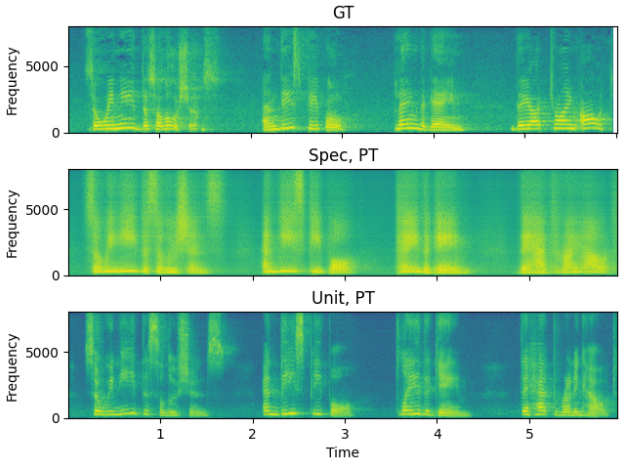

Figure 3. Spectrogram of ground truth (GT) and predicted audio generated by an ablated model that predicts spectrogram (Spec, PT) and the proposed ReVISE model (Unit, PT). We can observe that "Spec, PT" is more blurry, where harmonics (horizontal stripes) are missing.

	WER	
Target	No-PT	PT
Unit	78.6%	35.5%
Spec	96.9%	39.7%

Table 11. Ablation studies comparing prediction target (unit vs. spec) and effectiveness of pre-training on different targets.

num. of updates	15000
num. of frozen steps	0
tri-stage LR schedule	(33%, 0%, 67%)
peak learning rate	5e-5
audio masking prob	0
audio masking length	n/a
batch size / GPU	1000
num. of GPU	8
Adam (β_1, β_2)	(0.9, 0.98)

Table 12. EasyCom experiment hyperparameters.

D. Model Configurations

Tab. 13 and Tab. 12 detail the hyperparameters used for each experiment trained on LRS3 and EasyCom. Best checkpoints are selected based on the unit prediction accuracy on the validation set. Tri-stage learning rate schedules are used for all experiments, where the learning rate first ramps up linearly from 0 to the peak learning rate for the $t_1\%$ of the total updates, remains at the peak learning rate for the next $t_2\%$ of the total updates, and linearly decay to 5% of the peak learning rate during the rest of the updates. We use $(t_1, t_2, 1-t_1-t_2)$ to denote the learning rate schedule. Following [4], we freeze the pre-trained module for a number of updates (num. of frozen steps) and only fine-tune the new modules (upsampling and softmax layers) at the beginning of training. We also follow [4] to use SpecAug [42] for data augmentation and regularization, where random audio spans are dropped. The length of the spans and the ratio of frames dropped are labeled as "audio masking length" and "audio masking prob", respectively.

E. Mean Opinion Score Evaluation

We asked raters to rate the quality of the recordings without considering the identity of the speaker, we point the raters to focus on the fidelity of the generated output following the typical text-to-speech synthesis evaluation protocol for audio quality. The raters evaluated the quality of the speech recordings without access to the videos. The premise beyond the MOS test is to complete the rest of the reported metrics, to cover generation fidelity. This creates a full set of metrics concerning content evaluation using WER, prosody evaluation using VDE and FFE, video syncing evaluation using SyncNet, and synthesis quality evaluation using MOS.

	universal	video-to-speech	inpainting	speech denoising	source separation
num. of updates	45000	45000	45000	45000	45000
num. of frozen steps	10000	5000	30000	5000	5000
tri-stage LR schedule	(10%, 0%, 90%)	(10%, 20%, 70%)	(10%, 0%, 90%)	(10%, 0%, 90%)	(10%, 0%, 90%)
peak learning rate	1e-4	6e-5	1e-4	1e-4	1e-4
audio masking prob	35%	n/a	35%	35%	30%
audio masking length	1	n/a	1	1	1
batch size / GPU	1000	1000	1000	1000	1000
num. of GPU	8	8	8	8	8
Adam (β_1, β_2)	(0.9, 0.98)	(0.9, 0.98)	(0.9, 0.98)	(0.9, 0.98)	(0.9, 0.98)

Table 13. LRS3 experiment hyperparameters.