Efficient Streaming Audio-Visual Target Speaker Extraction for Real-World Acoustic Scenes

Wendi Sang[1,2,3,*] **Kai Li**[2,*] **Yifan Li**[1,3]
Jianqiang Huang[1,3,†] **Xiaolin Hu**[2,4,†]
[1]School of Computer Technology and Application, Qinghai University, Xining 810016, China
[2]Department of Computer Science and Technology, Institute for AI, BNRist, IDG/McGovern Institute for Brain Research, Tsinghua University, Beijing 100084, China
[3]Intelligent Computing and Application Laboratory of Qinghai Province, Qinghai University, Xining 810016, China
[4]Chinese Institute for Brain Research (CIBR), Beijing 100010, China

Abstract

Streaming audio-visual target speaker extraction (AVTSE) is essential for latency-sensitive applications. However, current causal systems face two coupled gaps. First, standard benchmarks fix the number of speakers at two and ignore the near-/far-field interference, reverberation, and sudden noise of real scenes. Second, high-performance causal architectures are too heavy for edge deployment. Lightweight alternatives try to reclaim capacity by repeatedly invoking a small separator, which offsets the savings. To address the first gap, we release RealSSA, a realistic AVTSE benchmark consisting of RealSSA-Sim and RealSSA-Real. RealSSA-Sim provides controllable simulated scenes with 1–5 dynamic near-field interfering speakers; scenes with far-field interference additionally include 10–15 far-field interfering speakers. RealSSA-Real provides a held-out real-recorded evaluation set. To address the second gap, we propose Falcon, a causal AVTSE method. Its Separator performs multi-scale time-frequency separation in a single encoder–decoder pass. We train Falcon with an encoder-space speaker contrastive loss. This loss suppresses near-field leakage at zero inference cost and transfers across backbones. Falcon achieves state-of-the-art extraction quality on RealSSA-Sim, LRS2-2Mix, LRS3-2Mix, and VoxCeleb2-2Mix. Compared with the causal AV-TFGridNet baseline, it cuts parameters by 91.9%, computation by 20.2×, and GPU inference latency by 11.1×. Code and demo are available at `https://cslikai.cn/Falcon/`.

1 Introduction

Audio-visual target speaker extraction (AVTSE) leverages a speaker's visual stream, typically lip movements, as enrollment cues to separate the target speaker's voice from a multi-speaker mixture [1, 2]. Although this setting has often been described in prior work as audio-visual speech separation [3–6], these methods all condition on the visual stream of a single target speaker, which by definition is target speaker extraction; therefore, we use the term AVTSE throughout this paper. Many latency-sensitive applications now demand *causal* AVTSE, such as video conferencing [7] and hearing-assistive devices [8]. However, the current causal AVTSE methods fall short along two axes: the data used to train and evaluate causal models, and the architecture of the causal model itself.

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Small Rooms (b) Coffee Shops (c) Shopping Malls (d) Outdoor Streets

Figure 1: Real-world AVTSE scenarios motivating RealSSA. The four scene types, ranging from small rooms to outdoor streets, cover varying number of speakers, near-/far-field interference, reverberation, and sudden environmental noise beyond standard fixed-speaker benchmarks.

Existing AVTSE benchmarks do not reflect the acoustic conditions of deployment. Some benchmarks built on LRS2 [9], LRS3 [10], and VoxCeleb2 [11] fix the number of speakers at two, draw mixtures at controlled SNRs, and hold room acoustics and background noise constant, leaving out the near-/far-field geometry, reverberation, and sudden noise events that characterize everyday recordings (Figure 1) [12, 13]. None of them are paired with a real-recorded evaluation set, and reported gains offer little evidence that a causal model will hold up once deployed. Therefore, the AVTSE task requires a benchmark that can simulate simultaneous changes in the number of speakers, spatial geometry, and noise, while also providing real recordings to evaluate real-world deployment performance.

On the model side, current causal AVTSE faces an efficiency gap. Causal variants of strong non-causal architectures preserve the extraction quality. Some methods include AV-TFGridNet [14], AV-Mossformer2 [15], and RTFSNet [16], obtained under the protocol of SwiftNet [17]. However, their parameter counts and per-frame FLOPs push inference latency beyond real-time budgets on edge hardware. Lightweight alternatives such as SwiftNet [17] recover capacity by repeatedly invoking a small separator after audio-visual fusion. Yet each additional cycle multiplies per-frame computation and erodes the latency savings of the smaller module. On the training side, SI-SNR penalizes deviation from the clean target [18], but does not explicitly push the estimate away from near-field speakers. This leaves residual leakage from spatially close talkers, and the leakage grows with the number of simultaneously active near-field speakers in realistic scenes [1, 19].

To close this data gap, we release *RealSSA*, a realistic AVTSE benchmark consisting of RealSSA-Sim and RealSSA-Real. RealSSA-Sim provides controllable simulated scenes with 1–5 dynamic near-field interfering speakers across four scene types; scenes with far-field interference additionally include 10–15 far-field interfering speakers. RealSSA-Real provides a held-out real-recorded evaluation set. We jointly sample room reverberation, ambient noise, and sudden noise events, so that the number of interfering speakers, spatial geometry, and environmental acoustics vary together rather than one axis at a time. RealSSA-Real further allows causal models trained on simulation to be evaluated on deployment-side distributions without adaptation.

To close the model-side gap, we propose *Falcon*, a single-pass causal AVTSE method. At its core, the *Separator* applies an encoder–decoder trunk to the audio-visual fused feature. In a single decoder traversal, it performs frequency-axis, time-axis, and joint time-frequency modeling, assigned to different decoder levels from coarse to fine scales. We further introduce an *encoder-space speaker contrastive loss*, computed in the separator's STFT encoder space against the clean target and each near-field interferer. This loss suppresses residual leakage at zero inference cost and transfers across AVTSE backbones. On RealSSA-Sim, LRS2-2Mix [9], LRS3-2Mix [10], and VoxCeleb2-2Mix [11], Falcon matched or exceeded the strongest causal baselines [14, 15, 17, 20] in separation quality (Table 3). On RealSSA-Real, it achieved competitive perceptual quality, WER, and target-speaker similarity. Meanwhile, it reduced parameters, computational cost, and GPU inference latency by 91.9%, 20.2×, and 11.1× relative to the causal AV-TFGridNet [14].

2 Related Works

2.1 AVTSE Benchmarks

AVTSE benchmarks can be organized along three axes: speaker configuration, acoustic conditions, and evaluation realism (Table 1). Along the first axis, LRS2-2Mix [9], LRS3-2Mix [10], and VoxCeleb2-2Mix [11] fix two near-field speakers per mixture, and therefore do not expose models to varying counts of near- and far-field interferers. These three corpora are used by most AVTSE models [4, 6, 5, 16, 20] and their causal variants [17]. Along the second axis, these corpora inherit acoustic conditions from the source videos, and offer no controlled reverberation or sudden noise events. The LRS3+WHAM! noisy protocol, reused by [21–23], adds WHAM! [12] background noise at SNR roughly in $[-6, +3]$ dB, but still keeps two speakers, anechoic mixing, and no sudden noise. The NTCD-TIMIT-based 2-speaker+noise protocol [21, 24] likewise fixes two speakers, and varies only the noise SNR in $[-5, 20]$ dB. Along the third axis, evaluation relies on held-out splits of the same source videos rather than independent real recordings. The gap between simulated mixtures and true on-device capture therefore remains undiagnosed. As summarized in Table 1, RealSSA is the only AVTSE benchmark that satisfies all three axes jointly.

Table 1: AVTSE benchmarks organized along three axes: speaker configuration, acoustic conditions, and evaluation realism. "NF" and "FF" denote near- and far-field sources; "Env. noise" means an external noise corpus is mixed in.

Benchmark	NF speakers	FF speakers	Reverb control	Env. noise	Sudden noise	Real-recorded eval
NTCD-TIMIT+noise [2017]	2	✗	✗	✓	✗	✗
LRS2-2Mix [2018]	2	✗	✗	✗	✗	✗
LRS3-2Mix [2018]	2	✗	✗	✗	✗	✗
VoxCeleb2-2Mix [2018]	2	✗	✗	✗	✗	✗
LRS3+WHAM! [2023]	2	✗	✗	✓	✗	✗
RealSSA (ours)	**2–6**	✓	✓	✓	✓	✓

2.2 Causal Audio-Visual Target Speaker Extraction

Causal AVTSE methods mainly differ in two aspects: separator architecture and training objective. On the architecture side, causal evaluation largely follows the protocol of SwiftNet [17], which provides a recipe for converting non-causal AVTSE backbones into their causal counterparts by restricting temporal operations to past and current frames; earlier time-domain separators [25–27, 6] and recent time-frequency separators are causalized under this protocol. Among the time-frequency family, AV-TFGridNet [14] and AV-Mossformer2 [15] retain strong extraction quality at the cost of large parameter counts and heavy per-frame computation, while SwiftNet and the causalized RTFSNet [16] and AVLiT [21] variants reuse a single compact block across multiple iterative cycles, trading parameters for per-frame FLOPs. Falcon instead employs level-specific time-frequency blocks within a single encoder–decoder forward pass.

On the training side, AVTSE systems were typically optimized with waveform reconstruction losses such as SI-SNR [18], and prior attempts to suppress residual interferers either adjust the training mixture distribution or add routing-level gates over the separator output [19, 14] rather than explicitly separating the estimate from each interfering speaker. Falcon instead adds an encoder-space speaker contrastive loss, discarded at inference time.

3 The RealSSA Dataset

RealSSA consists of two parts: RealSSA-Sim, a controllable simulated-scene suite used for training, validation, and testing, and RealSSA-Real, a held-out real-recorded evaluation set used to assess generalization under non-simulated acoustic conditions. RealSSA-Sim provides systematic control over speaker configuration, source distance, reverberation, environmental noise, and sudden noise events, while RealSSA-Real reflects raw on-device recordings in everyday acoustic environments.

RealSSA-Sim: Simulated Scene Suite. RealSSA-Sim defines four acoustic scene types that span a wide range of room sizes, reverberation levels, and noise conditions (Table 2). Small meeting rooms have short RT60 and only near-field interferers, representing close-range conversations. Medium indoor spaces introduce far-field speakers and moderate reverberation. Large public spaces

further increase RT60 and the number of interferers, reflecting scenarios such as lecture halls. Outdoor environments have minimal reverberation but high ambient noise levels. Across the four scenes, the number of near-field interfering speakers varies from 1 to 5; scenes with far-field interference additionally include 10–15 far-field interfering speakers. Each scene independently samples RT60, near-/far-field geometry, environmental noise, and sudden noise events. We classify an interferer as *near-field* when it lies within $3\,\mathrm{m}$ of the microphone and as *far-field* otherwise; this threshold approximates the critical distance in typical indoor rooms, beyond which reverberant energy dominates the direct path [28]. Per-scene spatial layouts are illustrated in Figure 5 (Appendix A.1).

Table 2: Acoustic scene configurations in RealSSA-Sim, covering four room types with controlled reverberation, speaker distance, and noise conditions. Near speakers and Far speakers denote interfering speakers and exclude the target.

Scene	Space size	RT60	Near speakers	Far speakers	Env. noise SIR	Sudden noise
Small meeting	$4\text{-}6 \times 3\text{-}5 \times 2.5\text{-}3\,\mathrm{m}$	0.3–0.6 s	1–5, 0.5–1.5 m	none	5–15 dB	30%, 1 event
Medium indoor	$8\text{-}15 \times 8\text{-}12 \times 3\text{-}4\,\mathrm{m}$	0.5–0.9 s	1–3, 1.0–2.5 m	10–15, 3–15 m	5–10 dB	40%, 1–2 events
Large public	$20\text{-}40 \times 15\text{-}30 \times 4\text{-}8\,\mathrm{m}$	1.0–2.0 s	1–4, 1.0–3.0 m	10–15, 5–20 m	2–8 dB	50%, 1–3 events
Outdoor	$50 \times 50 \times 20\,\mathrm{m}$	0.05–0.1 s	1–4, 1.0–4.0 m	10–15, 5–25 m	-2–5 dB	40%, 1–3 events

Target audio-visual data are sampled from LRS3 [10], while near-field and far-field interfering speakers come from LibriSpeech [29]. Non-speech noise is drawn from WHAM! [12] for continuous backgrounds and Hive [30] for sudden noise events. The generation pipeline proceeds in three stages. **(1) Room simulation.** For each sample, we randomly select a scene type and sample the room geometry and RT60 from the corresponding ranges in Table 2. Room impulse responses are computed with gpuRIR [31]. **(2) Source placement.** The target speaker is placed at a random position within the near-field range of the selected scene. Interfering speakers are placed according to the scene-specific near-field and far-field distance ranges. **(3) Mixing.** The target speech is normalized to -23 LUFS following the EBU R128 broadcast loudness standard [32]. Interfering speakers and environmental noise are added at scene-specific SIR ranges, and the final mixture is peak-limited to $-1\,\mathrm{dBFS}$. All audio is single-channel at $16\,\mathrm{kHz}$. RealSSA-Sim contains $100{,}826$ training, $9{,}133$ validation, and $9{,}169$ test samples, totaling approximately 153 hours. The four scene types are uniformly sampled, with an average of 2.5 near-field interfering speakers per sample. Source corpora and scene-design rationale are detailed in Appendix A.1, the full mixing pipeline in Appendix A.2, and per-split distribution statistics in Appendix A.3. We evaluated the generalization from RealSSA-Sim training to RealSSA-Real in Section 6.3.

RealSSA-Real: Real-World Evaluation Set. In addition to the simulated suite, we collect a real-recorded evaluation set, denoted as RealSSA-Real, to assess generalization under non-simulated acoustic conditions. RealSSA-Real covers everyday indoor, public, and outdoor environments with natural background noise, room reverberation, and uncontrolled interfering speech. It is recorded with a single mobile device and preserves the raw acoustic conditions of real on-device capture. The recordings are post-processed only to match the input format used by the AVTSE pipeline, including audio resampling to 16 kHz, video conversion to 25 fps, and extraction of 96×96 grayscale mouth regions. RealSSA-Real is used exclusively for evaluation and is never used for training or validation. We assess perceptual quality with DNSMOS and NISQA, speech recognizability with WER, and target-speaker consistency with speaker similarity. Detailed recording protocol, environment composition, and privacy processing are provided in Appendix A.4.

4 Methods

4.1 Overall Pipeline

Given a mixture waveform $\mathbf{x} \in \mathbb{R}^L$ of length L and a mouth video $\mathbf{m} \in \mathbb{R}^{H \times W \times T_\mathrm{v}}$ of the target speaker with spatial resolution $H \times W$ and T_v frames, Falcon aims to extract the target speech $\mathbf{s} \in \mathbb{R}^L$ as captured at the microphone. The mixture is modeled as $\mathbf{x} = \mathbf{s} + \sum_{k=1}^{K} \mathbf{b}_k + \mathbf{f} + \mathbf{n}$, where $\mathbf{s}, \mathbf{b}_k, \mathbf{f} \in \mathbb{R}^L$ denote the target speech, the k-th near-field interfering speech signal, and the sum of far-field interfering speech signals, respectively, all captured at the microphone, including room reverberation. The term $\mathbf{n} \in \mathbb{R}^L$ denotes additive environmental noise, and $\mathbf{f} = \mathbf{0}$ for scenes without far-field interferers.

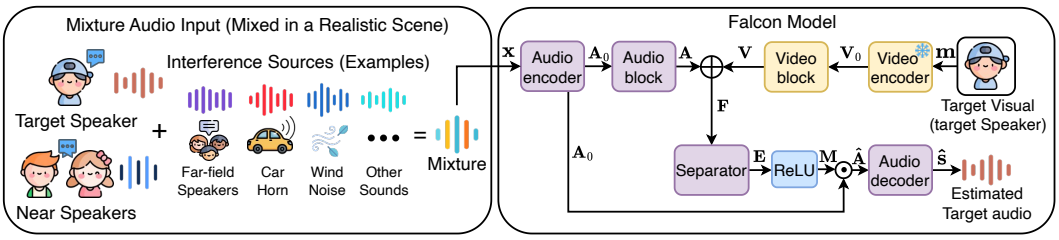


Figure 2: Overall pipeline of Falcon for audio-visual target speaker extraction.

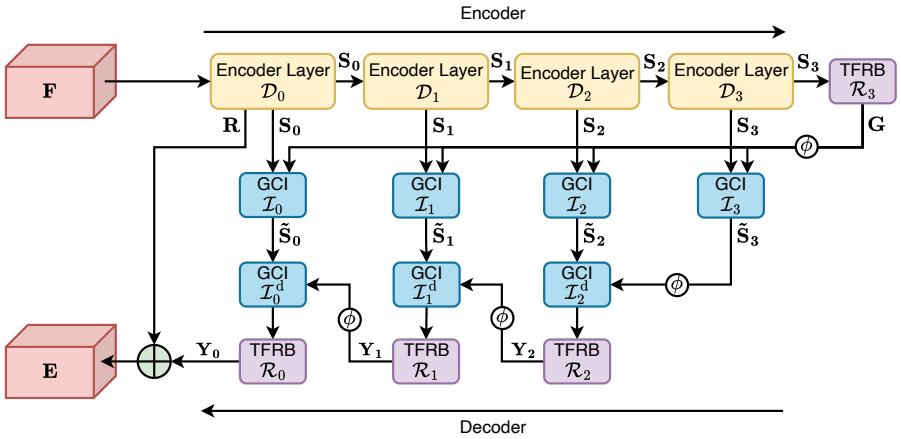


Figure 3: Single-Pass Multi-Scale Separator. ϕ denotes the upsampling operation.

As illustrated in Figure 2, the FRCNN visual backbone used in CTCNet [6], frozen during training, extracts mouth features from $\mathbf{m}$ and maps them through a visual bottleneck to obtain $\mathbf{V}_0 \in \mathbb{R}^{C_\mathrm{v} \times T_\mathrm{v}}$, where C_v is the visual feature dimension. The audio encoder $E_\mathrm{a} : \mathbb{R}^L \to \mathbb{R}^{C_\mathrm{e} \times T \times N_\mathrm{f}}$ applies STFT to its input, concatenates the real part, imaginary part, and magnitude spectrum as a three-channel tensor, and processes them through a 2D convolutional layer; we write $\mathbf{A}_0 = E_\mathrm{a}(\mathbf{x})$, where T and N_f denote the number of STFT frames and frequency bins, and C_e is the audio encoder output dimension. An audio bottleneck then projects $\mathbf{A}_0$ to $\mathbf{A} \in \mathbb{R}^{C \times T \times N_\mathrm{f}}$.

We refine each modality before fusion. The audio feature $\mathbf{A}$ is processed by an AudioBlock $\mathcal{B}_\mathrm{a}$, whose structure is the same as the separator detailed in Section 4.2, yielding $\tilde{\mathbf{A}} = \mathcal{B}_\mathrm{a}(\mathbf{A}) \in \mathbb{R}^{C \times T \times N_\mathrm{f}}$. The visual feature $\mathbf{V}_0$ is processed by the VideoBlock $\mathcal{B}_\mathrm{v}$ of SwiftNet [17], which aggregates temporal context with a causal LSTM and upsamples to the audio frame rate, yielding $\tilde{\mathbf{V}} = \mathcal{B}_\mathrm{v}(\mathbf{V}_0) \in \mathbb{R}^{C \times T}$. $\tilde{\mathbf{V}}$ is expanded along the frequency axis and added to $\tilde{\mathbf{A}}$, producing $\mathbf{F} \in \mathbb{R}^{C \times T \times N_\mathrm{f}}$. Then, the fused feature $\mathbf{F}$ is processed by the separator $\mathcal{B}_\mathrm{a}$ to obtain the target-speaker feature $\mathbf{E} \in \mathbb{R}^{C \times T \times N_\mathrm{f}}$. Each application of $\mathcal{B}_\mathrm{a}$ completes multi-scale time-frequency modeling in one encoder–decoder forward pass and is not unrolled along the time axis. Then, the mask generator predicts a time-frequency mask $\mathbf{M} \in \mathbb{R}^{C_\mathrm{e} \times T \times N_\mathrm{f}}$ from $\mathbf{E}$. We split the channel dimensions of $\mathbf{M}$ and $\mathbf{A}_0$ evenly into real and imaginary sub-bands, denoted $\mathbf{M} = [\mathbf{M}_\mathrm{r}; \mathbf{M}_\mathrm{i}]$ and $\mathbf{A}_0 = [\mathbf{A}_{0,\mathrm{r}}; \mathbf{A}_{0,\mathrm{i}}]$, and obtain the masked embedding by complex multiplication,

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{M}_\mathrm{r} \odot \mathbf{A}_{0,\mathrm{r}} - \mathbf{M}_\mathrm{i} \odot \mathbf{A}_{0,\mathrm{i}} \\ \mathbf{M}_\mathrm{r} \odot \mathbf{A}_{0,\mathrm{i}} + \mathbf{M}_\mathrm{i} \odot \mathbf{A}_{0,\mathrm{r}} \end{bmatrix} \in \mathbb{R}^{C_\mathrm{e} \times T \times N_\mathrm{f}}, \tag{1}$$

where $\odot$ is element-wise multiplication. The audio decoder finally maps $\hat{\mathbf{A}}$ back to real and imaginary spectral components and reconstructs the target speech $\hat{\mathbf{s}} \in \mathbb{R}^L$ via iSTFT.

Separator plays a central role in Falcon, since it is responsible for extracting the $\mathbf{F}$ into the target speaker features $\mathbf{E}$. We next introduce our single-pass multi-scale design for Separator.

4.2 Single-Pass Multi-Scale Separator

The Separator is an asymmetric U-Net (Figure 3): time-frequency modeling is concentrated at the encoder bottleneck to form a single coarse scene context, while a level-specific TF block at

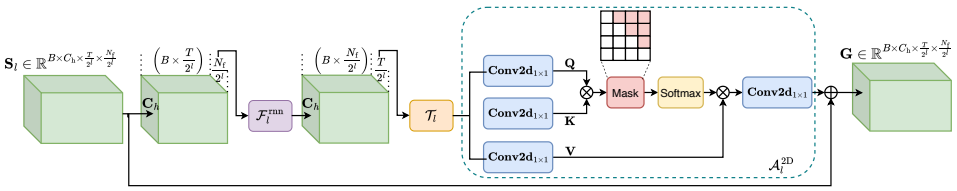


Figure 4: Time-Frequency Refinement Block.

every decoder resolution recovers fine spectral structure step by step. A single encoder–decoder pass therefore already yields four distinct TF scales, replacing the multi-iteration recurrence of SwiftNet [17].

The input feature $\mathbf{F}$ first passes through a depthwise 1×1 convolution with PReLU to form a residual branch $\mathbf{R} \in \mathbb{R}^{C \times T \times N_\mathrm{f}}$, which serves both as the entry to the hidden path below and as a global skip to the separator output. A pointwise 1×1 convolution with PReLU then projects $\mathbf{R}$ to the hidden channel dimension C_h, and a streaming-causal layer normalization $\mathrm{LN_c}$ produces $\mathbf{X}_0 \in \mathbb{R}^{C_\mathrm{h} \times T \times N_\mathrm{f}}$. For any input $\mathbf{Z}$, $\mathrm{LN_c}$ is defined per time-frequency position as

$$[\mathrm{LN_c}(\mathbf{Z})]_{c,t,f} = \gamma_c \cdot \frac{Z_{c,t,f} - \mu_{t,f}}{\sqrt{\sigma^2_{t,f} + \epsilon}} + \beta_c, \qquad \mu_{t,f} = \frac{1}{C_\mathrm{h}} \sum_c Z_{c,t,f}, \tag{2}$$

with $\sigma^2_{t,f}$ defined analogously, $\gamma, \beta \in \mathbb{R}^{C_\mathrm{h}}$ learnable, and a small ϵ for stability. $\mathrm{LN_c}$ is frame-local and cache-free, and is reused throughout the blocks below.

Encoder. The encoder has $L{=}4$ levels. $\mathcal{D}_0$ is a stride-1 depthwise $3{\times}3$ convolution; $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ are stride-2 depthwise $5{\times}5$ convolutions that halve both axes. All convolutions use left-aligned padding along time and symmetric padding along frequency, adding no temporal lookahead. Starting from $\mathbf{S}_0 = \mathcal{D}_0(\mathbf{X}_0)$, subsequent levels are produced in order $l = 1, 2, 3$ by $\mathbf{S}_l = \mathcal{D}_l(\mathbf{S}_{l-1})$, giving

$$\mathbf{S}_l \in \mathbb{R}^{C_\mathrm{h} \times \frac{T}{2^l} \times \frac{N_\mathrm{f}}{2^l}}, \quad l = 0, \ldots, 3. \tag{3}$$

Each $\mathbf{S}_l$ is kept as a skip. Time-frequency modeling is applied only at the bottleneck,

$$\mathbf{G} = \mathcal{R}_3(\mathbf{S}_3) \in \mathbb{R}^{C_\mathrm{h} \times \frac{T}{8} \times \frac{N_\mathrm{f}}{8}}, \tag{4}$$

where $\mathcal{R}_l(\cdot)$ is the Time-Frequency Refinement Block (TFRB) at level l (Figure 4); we identify $\mathbf{S}_3 \equiv \mathbf{G}$ in what follows.

Each TFRB is a cascade of three stages applied in order: first a bidirectional LSTM along frequency $\mathcal{F}^{\mathrm{rnn}}_l$, then a unidirectional LSTM along time $\mathcal{T}_l$, and finally a 2D self-attention $\mathcal{A}^{\mathrm{2D}}_l$,

$$\mathcal{R}_l(\mathbf{X}) = \mathcal{A}^{\mathrm{2D}}_l\left(\mathcal{T}_l\left(\mathcal{F}^{\mathrm{rnn}}_l(\mathbf{X})\right)\right), \tag{5}$$

where $\mathcal{A}^{\mathrm{2D}}_l$ operates on the flattened time-frequency plane with a block-causal mask: a query at (t, f) attends to every key (t', f') with $t' \le t$. Each stage is wrapped in a pre-norm residual block with $\mathrm{LN_c}$.

Global context injection (GCI) module. For $l = 0, \ldots, 3$, we inject $\mathbf{G}$ into each skip via

$$\tilde{\mathbf{S}}_l = \mathcal{I}_l(\mathbf{S}_l, \mathbf{G}) = \phi_{\mathrm{loc}}(\mathbf{S}_l) \odot \sigma(\phi_{\mathrm{gate}}(\mathbf{G})) + \phi_{\mathrm{glb}}(\mathbf{G}), \tag{6}$$

where $\mathbf{G}$ is first upsampled to the resolution of $\mathbf{S}_l$ by nearest-neighbor interpolation when needed, and $\phi_{\mathrm{loc}}, \phi_{\mathrm{glb}}, \phi_{\mathrm{gate}}$ are depthwise convolutions with $\mathrm{LN_c}$. The gate $\sigma(\phi_{\mathrm{gate}}(\mathbf{G}))$ lets the global context emphasize or suppress frequency bands of the local skip.

Decoder. The decoder restores resolution coarse-to-fine: for $l = 2, 1, 0$ in that order,

$$\mathbf{Y}_l = \mathcal{R}_l\left(\mathcal{I}^{\mathrm{d}}_l(\tilde{\mathbf{S}}_l, \mathbf{Y}_{l+1}) + \mathbf{S}_l\right), \tag{7}$$

with $\mathbf{Y}_3 \equiv \tilde{\mathbf{S}}_3$ and $\mathcal{I}^{\mathrm{d}}_l(\cdot)$ the decoder-side GCI module whose coarser argument is first upsampled. The $+\mathbf{S}_l$ term is an identity residual that preserves gradient flow and a raw spectral reference when the gate in $\mathcal{I}^{\mathrm{d}}_l$ collapses $\tilde{\mathbf{S}}_l$ toward zero. Since each $\mathcal{R}_l$ operates at a different resolution, the four TFRBs span four TF scales in one forward pass.

The decoder output $\mathbf{Y}_0 \in \mathbb{R}^{C_\mathrm{h} \times T \times N_\mathrm{f}}$ is projected to C channels and added to the residual branch, $\mathbf{E} = \mathcal{O}(\mathbf{Y}_0) + \mathbf{R}$, where $\mathcal{O}(\cdot)$ is a pointwise projection.

4.3 Encoder-Space Speaker Contrastive Loss

Standard reconstruction losses such as SI-SNR minimize the distance between the estimate and the clean target but do not explicitly push the estimate away from near-field interferers. We add a contrastive loss that reuses the audio encoder E_{a} for both the query and the references, so no separate encoder is introduced.

The query is the estimated target in encoder space, $\hat{\mathbf{A}} \in \mathbb{R}^{C_{\mathrm{e}} \times T \times N_{\mathrm{f}}}$, obtained from the complex masking of Section 4.1. The positive and negative references are the clean target $\mathbf{s}$ and each near-field interferer $\mathbf{b}_k$, both captured at the microphone and therefore carrying the same room response as the mixture, passed through E_{a} with stop-gradient on the reference path:

$$\mathbf{A}^{+} = \mathrm{sg}(E_{\mathrm{a}}(\mathbf{s})), \qquad \mathbf{A}_k^{-} = \mathrm{sg}(E_{\mathrm{a}}(\mathbf{b}_k)), \quad k = 1, \ldots, K. \tag{8}$$

Gradients flow into E_{a} only through the query $\hat{\mathbf{A}}$, not through the references. Far-field interferers are excluded from the negative set: both distance attenuation and, in reverberant scenes, late reverberation drive their residual below the SI-SNR error floor, so adding them as negatives yields no additional signal. A lightweight speaker pooling function $\mathcal{P}_{\text{spk}}(\cdot)$ then maps each encoder-space feature to a unit-norm speaker embedding by averaging over frequency, adaptively pooling over time, and projecting it to a fixed-dimensional vector:

$$\mathbf{q} = \mathcal{P}_{\text{spk}}(\hat{\mathbf{A}}), \qquad \mathbf{p} = \mathcal{P}_{\text{spk}}(\mathbf{A}^{+}), \qquad \mathbf{e}_k^{-} = \mathcal{P}_{\text{spk}}(\mathbf{A}_k^{-}). \tag{9}$$

Writing $s_k \triangleq \cos(\mathbf{q}, \mathbf{e}_k^{-})$, the contrastive loss combines a positive attraction term with a softmax-weighted, margin-based repulsion over negatives:

$$\mathcal{L}_{\text{spk}} = \big(1 - \cos(\mathbf{q}, \mathbf{p})\big) + \sum_{k=1}^{K} \frac{\exp(\tau\, s_k)}{\sum_{j=1}^{K} \exp(\tau\, s_j)}\ \mathrm{ReLU}(s_k - m), \tag{10}$$

where m=0.5 (a 60° angular margin) deactivates the repulsion once $\mathbf{q}$ is sufficiently far from $\mathbf{e}_k^{-}$, leaving SI-SNR to handle the remaining residual. The softmax weight acts via an *inverse* temperature τ: τ=0 gives uniform weighting across negatives, and larger τ sharpens the distribution toward the hardest negative. We linearly anneal τ from 0 to 1 during training.

The final training objective is $\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{spk}}\, \mathcal{L}_{\text{spk}}$ with $\mathcal{L}_{\text{recon}} = -\,\text{SI-SNR}(\hat{\mathbf{s}}, \mathbf{s})$ [18] and $\lambda_{\text{spk}} = 1.0$. Since query and reference signals share the same audio encoder, this loss can be readily attached to any audio encoder AVTSE backbone with no additional architectural changes required.

5 Experiment Configurations

5.1 Dataset Setup

We evaluated on three public AVTSE benchmarks (LRS2 [9], LRS3 [10], VoxCeleb2 [11]), RealSSA-Sim (Section 3) and RealSSA-Real (Appendix A.4). Mixtures on the three public benchmarks followed the protocol of prior works [6, 17]. All clips were 16 kHz audio paired with 96×96 grayscale mouth crops at 25 fps, and dataset splits followed the official partitions of each corpus. RealSSA-Real was used exclusively for evaluation. Further preprocessing details are given in Appendix B.

5.2 Model Configurations and Evaluation Metrics

The separator hidden dimension C_{h} was set to 64, and the full width assignment across C_{e}, C, C_{v}, and $\dim(\mathbf{q})$ is tabulated in Appendix C. The audio encoder E_{a} used an STFT with window size 256 and hop 128 (8 ms hop at 16 kHz), and fed the real, imaginary, and magnitude components as a three-channel input, following Section 4.1. For all models trained in this work, we used AdamW [33] (learning rate 1×10^{-3}, weight decay 0.01), with the learning rate halved whenever the validation loss plateaued for 15 consecutive epochs and early stopping applied after a patience of 30 epochs. Full training hyperparameters are provided in Appendix D. We evaluated separation quality with SI-SNRi and SDRi [34] on the simulated benchmarks, and non-intrusive perceptual quality on RealSSA-Real with DNSMOS [35] and NISQA [36]. On RealSSA-Sim and RealSSA-Real, we additionally

measured word error rate (WER) with Mega-ASR [37] and target-speaker similarity with a pretrained ResNet34 speaker encoder from WeSpeaker [38]. We also reported the number of parameters, MACs, CPU and GPU inference latency, and memory usage to assess computational efficiency, all measured on a 2 s, 16 kHz input clip paired with 50 mouth-crop frames at 25 fps. Metric definitions and the inference-latency measurement protocol are detailed in Appendix E.

6 Results and Analysis

6.1 Main Results

We retrained each baseline separately on RealSSA-Sim and VoxCeleb2-2Mix. Within each dataset, all models used the same training, validation, and test splits, preprocessing pipeline, optimizer, and learning-rate schedule. We then evaluated the models trained on RealSSA-Sim on RealSSA-Real. Dolphin [20] and AV-Mossformer2 [15] were converted to causal variants (Appendix F). Baseline results on LRS2-2Mix and LRS3-2Mix are from Swift-Net [17].

Signal-level separation quality. Table 3 reports SI-SNRi and SDRi on the four simulated benchmarks. Falcon achieved the highest SI-SNRi and SDRi on all four. Its SI-SNRi gain over Swift-Net-12 was largest on VoxCeleb2-2Mix, reaching 1.1 dB. On RealSSA-Sim, which includes dynamic speaker counts, near-/far-field interference, and reverberation, Falcon achieved 7.7 dB SI-SNRi compared with 7.6 dB for AV-TF-GridNet. Appendix G provides results by scene type and near-field speaker count, and Appendix H presents qualitative spectrogram comparisons.

Table 3: Performance of causal AVTSE methods. LRS2-2Mix and LRS3-2Mix baseline results are from Swift-Net; other baseline results use our retrained models. Bold and underline mark the best and second-best results.

Method	RealSSA-Real	RealSSA-Sim		LRS2-2Mix		LRS3-2Mix		VoxCeleb2-2Mix	
	DNSMOS ↑	SI-SNRi ↑	SDRi ↑	SI-SNRi ↑	SDRi ↑	SI-SNRi ↑	SDRi ↑	SI-SNRi ↑	SDRi ↑
AV-ConvTasNet [2021]	2.580	1.2	2.6	9.8	10.3	12.5	12.8	6.9	8.2
AV-Sepformer [2023]	2.813	5.5	6.3	10.6	11.1	13.4	13.7	6.9	8.3
AV-TF-GridNet [2023]	3.030	<u>7.6</u>	<u>8.1</u>	12.7	13.1	14.3	14.8	5.8	7.7
RTFSNet-12 [2024]	<u>3.066</u>	4.9	5.7	12.4	12.8	14.0	14.4	8.4	10.7
CTCNet [2024]	2.840	5.4	6.2	8.0	8.8	9.4	10.2	2.5	4.7
AV-DPRNN [2025]	2.901	5.4	6.2	10.2	10.7	11.9	12.3	7.8	9.0
AV-Mossformer2 [2025]	2.912	6.0	6.7	13.2	13.9	<u>16.2</u>	<u>16.6</u>	7.7	9.1
Swift-Net-12 [2025]	**3.276**	7.3	7.8	13.9	14.1	16.0	16.3	<u>9.4</u>	10.8
Dolphin [2026]	2.960	6.7	7.3	<u>14.0</u>	<u>14.6</u>	14.3	14.9	9.2	<u>11.1</u>
Falcon (Ours)	**3.276**	**7.7**	**8.2**	**14.1**	**14.7**	**16.4**	**16.7**	**10.5**	**11.8**

Speech quality and target-speaker consistency. On RealSSA-Real, Falcon achieved the highest DNSMOS of 3.276, tying Swift-Net-12 and exceeding AV-TF-GridNet (3.030) and Dolphin (2.960) (Table 3). Table 4 complements this perceptual evaluation with WER and target-speaker similarity. Among extraction systems, Falcon achieved the lowest RealSSA-Sim WER at 25.47%, improving on AV-TF-GridNet by 1.84 percentage points. Its RealSSA-Real WER of 32.74% ranked second, 0.74 points behind AV-Mossformer2. Falcon also achieved the highest mean target-speaker similarity on both sets: 43.05 and 42.23, exceeding Swift-Net-12 by 1.26 and 2.19 points, respectively. Further perceptual evaluation and the effect of training data are analyzed in Section 6.3. The demo page[3] provides examples from the four RealSSA-Real scenarios.

6.2 Computational Efficiency

We evaluated Falcon's computational efficiency by comparing it with the four strongest causal baselines on RealSSA-Sim. Table 5 reports parameter counts, MACs, CPU and GPU inference latency, and memory usage under the common measurement protocol described in Appendix E.

Falcon and Swift-Net-12 both had 0.5 M parameters. At this model size, Falcon required 10.6 G MACs compared with 45.3 G for Swift-Net-12, a reduction of 76.6%, while GPU inference latency decreased from 172.1 ms to 51.3 ms (70.2%). Compared with AV-TF-GridNet, the strongest baseline on RealSSA-Sim, Falcon achieved higher separation quality with 91.9% fewer parameters, 95.0% fewer MACs, and 91.0% lower GPU latency. Compared with Dolphin, Falcon reduced parameter

[3] https://cslikai.cn/Falcon/

Table 4: WER (%, ↓) and target-speaker similarity (cosine similarity $\times 100$, ↑). Ground truth evaluates the target waveform using the same protocol; "–" denotes unavailable isolated target waveforms. Bold and underline mark the best and second-best extraction systems, excluding ground truth.

Method	RealSSA-Sim		RealSSA-Real	
	WER (%) ↓	Similarity ↑	WER (%) ↓	Similarity ↑
AV-Sepformer [2023]	32.64	38.69	34.61	38.25
AV-TF-GridNet [2023]	<u>27.31</u>	40.44	34.95	39.88
RTFSNet-12 [2024]	31.91	36.51	35.56	35.78
CTCNet [2024]	34.44	37.08	38.81	35.33
AV-Mossformer2 [2025]	32.44	37.34	**32.00**	36.42
Swift-Net-12 [2025]	28.59	<u>41.79</u>	33.72	<u>40.04</u>
Dolphin [2026]	49.48	24.98	39.06	35.95
Ground truth	11.83	64.69	–	–
Falcon (Ours)	**25.47**	**43.05**	<u>32.74</u>	**42.23**

count, MACs, and GPU latency by 91.1%, 41.4%, and 41.6%, respectively. Falcon also had the lowest memory footprint among the evaluated methods, using 125.3 MB of CPU RAM and 364.7 MB of GPU memory. Together with the separation results in Table 3, these comparisons show that Falcon combined high extraction quality with low computational and memory costs.

Table 5: Efficiency comparison among Falcon and the four best-performing causal AVTSE baselines from Table 3. Lower is better for all metrics.

Method	Params (M) ↓	MACs (G) ↓	Latency (ms) ↓		Memory (MB) ↓	
			CPU	GPU	RAM	GPU Inf.
AV-TF-GridNet [2023]	6.2	213.8	5646.3	567.8	<u>136.3</u>	1015.8
AV-Mossformer2 [2025]	56.7	224.2	16670.7	114.9	330.9	539.8
Swift-Net-12 [2025]	**0.5**	45.3	4163.8	172.1	482.3	411.7
Dolphin [2026]	<u>5.6</u>	<u>18.1</u>	<u>2411.0</u>	<u>87.9</u>	560.2	<u>390.3</u>
Falcon (Ours)	**0.5**	**10.6**	**1361.2**	**51.3**	**125.3**	**364.7**

6.3 Generalization to Real-World Scenes

We compared training on RealSSA-Sim and VoxCeleb2, the largest existing AVTSE corpus, using non-intrusive perceptual quality on the held-out RealSSA-Real.

Table 6: Evaluation on RealSSA-Real when models were trained on VoxCeleb2 versus RealSSA.

Method	VoxCeleb2		RealSSA-Sim (Ours)	
	NISQA	DNSMOS	NISQA	DNSMOS
AV-TF-GridNet	**2.495**	2.757	2.413	**3.030**
AV-Mossformer2	1.995	2.715	**2.256**	**2.912**
Swift-Net-12	2.481	3.064	**3.484**	**3.276**
Dolphin	1.593	2.608	**2.590**	**2.960**
Falcon (Ours)	2.431	2.878	**3.615**	**3.276**

Table 6 reports NISQA [36] and DNSMOS [35] for five causal architectures. RealSSA-Sim training improved DNSMOS for all five and NISQA for four (Falcon: $2.431 \rightarrow 3.615$ NISQA, $2.878 \rightarrow 3.276$ DNSMOS). Whereas VoxCeleb2 does not systematically vary near-/far-field interference, environmental and sudden noise, or reverberation, RealSSA-Sim varies these factors jointly. Gains across architectures support systematic acoustic coverage as a basis for real-world generalization.

6.4 Ablation Study

Number of downsampling stages. The depth of the single-pass separator controls how many time-frequency scales are exploited. Increasing depth from one to three stages improved SI-SNRi by 0.6 dB (Table 7): the deeper encoder captures broader spectro-temporal context while skip connections progressively recover fine detail.

Three stages is also the maximum depth permitted by streaming causality, since the visual branch imposes a fixed algorithmic lookahead of 120 ms and the audio hop is 8 ms, leaving the audio branch at most 15 frames of lookahead; a fourth stage would exceed this budget and break audio-visual alignment. We used LSTM as the recurrent unit inside each TFRB based on a comparison with RNN, GRU, and SRU reported in Appendix I.

Table 7: Effect of downsampling depth.

Stages	SI-SNRi ↑	SDRi ↑
1	7.1	7.7
2	<u>7.4</u>	<u>7.9</u>
3 (Ours)	**7.7**	**8.2**

Encoder-space speaker contrastive loss. We assessed the proposed loss by removing it from Falcon and by applying it to two recent causal AVTSE models with different separator designs. The contrastive loss improved Falcon by 0.4 dB SI-SNRi, SwiftNet-12 by 0.2 dB, and Dolphin by 0.3 dB (Table 8). Consistent gains across three architectures suggest that pulling the estimate toward the clean reference and pushing it away from near-field interferers in the encoder feature space provides an architecture-independent regularization signal. The contrastive branch is active only during training, so these gains come at zero inference cost. We further visualized the learned encoder-space speaker embeddings in Appendix J, showing that the proposed loss yields more speaker-discriminative representations.

Table 8: Encoder-space speaker contrastive loss. Gains are consistent across three architectures with different separator designs.

Model	Loss	SI-SNRi ↑	SDRi ↑
Falcon	Without	7.3	7.8
Falcon	+ Ours	**7.7**	**8.2**
SwiftNet-12	Baseline	7.3	7.8
SwiftNet-12	+ Ours	<u>7.5</u>	<u>8.0</u>
Dolphin	Baseline	6.7	7.3
Dolphin	+ Ours	7.0	7.6

We also examined the annealing schedule of the negative-weighting temperature τ (Table 9). Annealing τ from 0 to 1 outperformed both fixed schedules and a more aggressive $0 \to 2$ anneal. Uniform initial weighting lets the embedding space stabilize before focus shifts toward hard negatives; fixing τ=1 concentrates on a few similar interferers too early, and annealing to 2 over-concentrates late in training and weakens constraints on moderately similar speakers.

Table 9: Sensitivity to the τ schedule. Moderate annealing ($0 \to 1$) balances early exploration and late hard-negative focus.

τ Strategy	SI-SNRi ↑	SDRi ↑
Fixed τ=0	<u>7.5</u>	<u>8.1</u>
Fixed τ=1	7.4	7.9
Anneal $0 \to 2$	<u>7.5</u>	8.0
Anneal $0 \to 1$ (Ours)	**7.7**	**8.2**

Additional experiments examined output parameterization (Appendix K), speech-overlap ratio (Appendix L), and missing visual information (Appendix M).

7 Conclusions

RealSSA combines RealSSA-Sim and RealSSA-Real to benchmark AVTSE under variable speaker counts, near-/far-field geometry, reverberation, and sudden noise, with held-out real recordings for evaluating generalization. Falcon performs multi-scale time-frequency separation in a single encoder–decoder pass and uses an encoder-space speaker contrastive loss that suppresses near-field leakage, transfers across backbones, and adds no inference cost. Across RealSSA-Sim, RealSSA-Real, LRS2-2Mix, LRS3-2Mix, and VoxCeleb2-2Mix, Falcon matches or surpasses strong causal baselines. Relative to causal AV-TFGridNet, it reduces parameters by 91.9%, computation by 20.2×, and GPU inference latency by 11.1×, supporting progress toward low-latency on-device AVTSE.

Limitations. Both the separator and the encoder-space speaker contrastive loss in Falcon operate on a single channel, and we leave the four-element array simulated in Appendix A.2 to future multi-channel extensions. RealSSA-Real covers approximately 84 minutes over 10 participants and eight environments, which is sufficient to reveal cross-architecture trends and can be scaled up in follow-up releases. The contrastive loss currently relies on clean per-source waveforms of near-field interferers available in simulation, and adapting it to unlabeled real recordings is a natural direction for future work.

Acknowledgments and Disclosure of Funding

This work is supported by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM504), the National Natural Science Foundation of China (No. 62576187), the Major Science and Technology Projects in Qinghai Province (2024-GX-A3), and the National Key R&D Program of China under Grant Nos. 2024YFB2808800 and 2024YFB2808803.

References

[1] Kai Li, Guo Chen, Wendi Sang, Yi Luo, Zhuo Chen, Shuai Wang, Shulin He, Zhong-Qiu Wang, Andong Li, Zhiyong Wu, et al. Advances in speech separation: Techniques, challenges, and future trends. *arXiv preprint arXiv:2508.10830*, 2025.

[2] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. In *ACM SIGGRAPH*, pages 1–11, 2018.

[3] Ruohan Gao and Kristen Grauman. Visualvoice: Audio-visual speech separation with cross-modal consistency. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15490–15500. IEEE, 2021.

[4] Jiyoung Lee, Soo-Whan Chung, Sunok Kim, Hong-Goo Kang, and Kwanghoon Sohn. Looking into your speech: Learning cross-modal affinity for audio-visual speech separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1336–1345, 2021.

[5] Kai Li, Runxuan Yang, Fuchun Sun, and Xiaolin Hu. IIANet: An intra- and inter-modality attention network for audio-visual speech separation. In *Forty-first International Conference on Machine Learning*, 2024.

[6] Kai Li, Fenghua Xie, Hang Chen, Kexin Yuan, and Xiaolin Hu. An audio-visual speech separation model inspired by cortico-thalamo-cortical circuits. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6637–6651, 2024.

[7] Elio Gruttadauria, Mathieu Fontaine, and Slim Essid. Online speaker diarization of meetings guided by speech separation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11356–11360. IEEE, 2024.

[8] Amandeep Singh Dhanjal and Williamjeet Singh. Tools and techniques of assistive technology for hearing impaired people. In *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pages 205–210. IEEE, 2019.

[9] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):8717–8727, 2018.

[10] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. Lrs3-ted: a large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496*, 2018.

[11] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2018, pages 1086–1090, 2018.

[12] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. Wham!: Extending speech separation to noisy environments. *Interspeech 2019*, 2019.

[13] Bing Yang, Changsheng Quan, Ya-Bo Wang, Pengyu Wang, Yujie Yang, Ying Fang, Nian Shao, Hui Bu, Xin Xu, and Xiaofei Li. Realman: A real-recorded and annotated microphone array dataset for dynamic speech enhancement and localization. *Advances in Neural Information Processing Systems*, 37, 2024.

[14] Zexu Pan, Gordon Wichern, Yoshiki Masuyama, François G Germain, Sameer Khurana, Chiori Hori, and Jonathan Le Roux. Scenario-aware audio-visual tf-gridnet for target speech extraction. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE, 2023.

[15] Shengkui Zhao, Zexu Pan, and Bin Ma. Clearervoice-studio: Bridging advanced speech processing research and practical deployment. In *Proc. Interspeech 2025*, pages 2980–2984, 2025.

[16] Samuel Pegg, Kai Li, and Xiaolin Hu. RTFS-net: Recurrent time-frequency modelling for efficient audio-visual speech separation. In *The Twelfth International Conference on Learning Representations*, 2024.

[17] Wendi Sang, Kai Li, Run Yang, Jianqiang Huang, and Xiaolin Hu. A fast and lightweight model for causal audio-visual speech separation. In *European Conference on Artificial Intelligence*, 2025.

[18] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. Sdr–half-baked or well done? In *2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 626–630, 2019.

[19] Yun Liu, Xuechen Liu, and Junichi Yamagishi. Improving curriculum learning for target speaker extraction with synthetic speakers. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 524–531, 2024.

[20] Kai Li, Gao Kejun, and Xiaolin Hu. Efficient audio-visual speech separation with discrete lip semantics and multi-scale global-local attention. In *The Fourteenth International Conference on Learning Representations*, 2026.

[21] Héctor Martel, Julius Richter, Kai Li, Xiaolin Hu, and Timo Gerkmann. Audio-visual speech separation in noisy environments with a lightweight iterative model. In *Proc. Interspeech 2023*, 2023.

[22] Ke Xue, Rongfei Fan, Dawei Zhao, Chao Zhu, Han Hu, et al. From coarse to fine: Recursive audio-visual semantic enhancement for speech separation. *arXiv preprint arXiv:2509.22425*, 2025.

[23] Sidong Zhang, Shiv Shankar, Trang Nguyen, Andrea Fanelli, and Madalina Fiterau. Audio-visual speech separation via bottleneck iterative network. *arXiv preprint arXiv:2507.07270*, 2025.

[24] Ahmed Hussen Abdelaziz. NTCD-TIMIT: A new database and baseline for noise-robust audio-visual speech recognition. In *Proc. Interspeech 2017*, pages 3752–3756, 2017.

[25] Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27 (8):1256–1266, 2019.

[26] Ruijie Tao, Xinyuan Qian, Yidi Jiang, Junjie Li, Jiadong Wang, and Haizhou Li. Audio-visual target speaker extraction with reverse selective auditory attention. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 2025.

[27] Jiuxin Lin, Xinyu Cai, Heinrich Dinkel, Jun Chen, Zhiyong Yan, Yongqing Wang, Junbo Zhang, Zhiyong Wu, Yujun Wang, and Helen Meng. Av-sepformer: Cross-attention sepformer for audio-visual target speaker extraction. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[28] Heinrich Kuttruff. *Room Acoustics*. CRC Press, 6th edition, 2016.

[29] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[30] Kai Li, Jintao Cheng, Chang Zeng, Zijun Yan, Helin Wang, Zixiong Su, Bo Zheng, and Xiaolin Hu. A semantically consistent dataset for data-efficient query-based universal sound separation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

[31] David Diaz-Guerra, Antonio Miguel, and Jose R Beltran. gpurir: A python library for room impulse response simulation with gpu acceleration. *Multimedia Tools and Applications*, 80(4): 5653–5671, 2021.

[32] European Broadcasting Union. EBU R 128: Loudness normalisation and permitted maximum level of audio signals. Technical report, EBU, 2020.

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[34] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4): 1462–1469, 2006.

[35] Chandan KA Reddy, Vishak Gopal, and Ross Cutler. Dnsmos p. 835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 886–890. IEEE, 2022.

[36] Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller. Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. *Interspeech 2021*, 2021.

[37] Zhifei Xie, Kaiyu Pang, Haobin Zhang, Deheng Ye, Xiaobin Hu, Shuicheng Yan, and Chunyan Miao. Mega-asr: Towards in-the-wild^ 2 speech recognition via scaling up real-world acoustic simulation. *arXiv preprint arXiv:2605.19833*, 2026.

[38] Hongji Wang, Chengdong Liang, Shuai Wang, Zhengyang Chen, Binbin Zhang, Xu Xiang, Yanlei Deng, and Yanmin Qian. Wespeaker: A research and production oriented speaker embedding learning toolkit. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[39] Peng Zhang, Jiaming Xu, Yunzhe Hao, and Bo Xu. Online audio-visual speech separation with generative adversarial training. In *Proceedings of the 2021 7th International Conference on Computing and Artificial Intelligence*, pages 379–385, 2021.

A Detailed Dataset Construction of RealSSA

This appendix provides the construction details of RealSSA that complement Section 3. We describe the source corpora and the rationale behind the four scene types (Appendix A.1), the spatial simulation and mixing pipeline (Appendix A.2), the resulting distribution statistics (Appendix A.3), and RealSSA-Real (Appendix A.4). All design choices are made to make the benchmark reproducible from public corpora and to expose AVTSE models to acoustic conditions that fixed two-speaker datasets do not contain.

A.1 Source Data and Scene Design

Source corpora. RealSSA-Sim is built from four publicly available corpora that play complementary roles in the mixture (Table 10). LRS3 [10] provides target audio together with the temporally aligned mouth video that AVTSE models rely on, and is the only corpus among the four that supplies paired audio-visual data at scale. LibriSpeech [29] contributes near-field and far-field interfering speakers; its large speaker inventory and clean read speech allow us to add interferers without re-introducing the noise already present in the target corpus. WHAM! [12] supplies continuous environmental backgrounds recorded in real urban scenes, which we use to populate the always-on noise floor of each mixture. Hive [30] is a recently released sound-event corpus whose semantic label hierarchy lets us draw scene-appropriate environmental and sudden-noise events; this hierarchy is the property that distinguishes RealSSA-Sim from prior simulated AVTSE sets that mix WHAM! noise

without semantic control. LRS3, LibriSpeech, and WHAM! are natively at $16\,\mathrm{kHz}$, while Hive is provided at $44.1\,\mathrm{kHz}$ and is resampled to $16\,\mathrm{kHz}$ before mixing.

Table 10: Source corpora used to construct RealSSA-Sim. "Role" summarizes the function of each corpus in the mixture. "Hours" denotes the approximate total speech duration. "Speakers" indicates the number of distinct speakers (marked "–" if not applicable). "Native SR" is the original sampling rate, and "Used SR" is the resampled rate used in RealSSA-Sim.

Corpus	Role	Hours	Speakers	Native SR	Used SR
LRS3 [10]	Target (audio–visual)	∼438	∼9,545	16 kHz	16 kHz
LibriSpeech [29]	Near-/far-field interferers	∼1000	∼2,484	16 kHz	16 kHz
WHAM! [12]	Continuous environmental noise	∼80	–	16 kHz	16 kHz
Hive [30]	Semantic environmental and sudden events	∼544	–	44.1 kHz	16 kHz

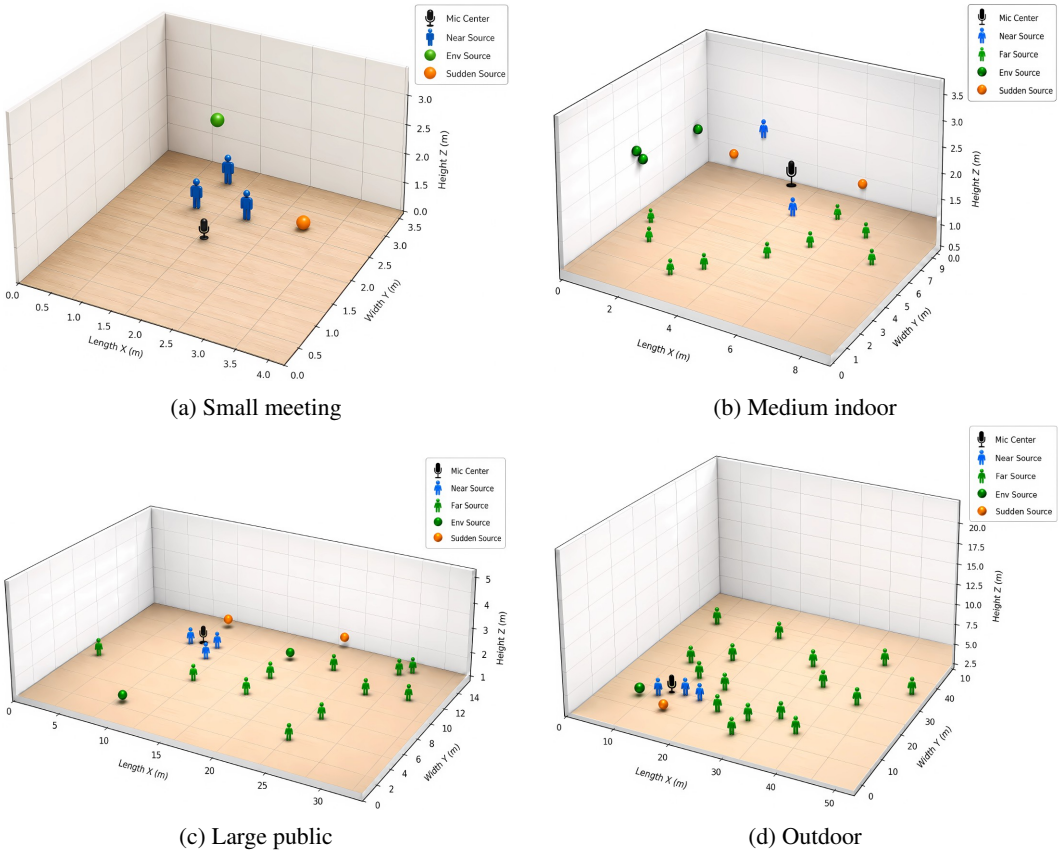
Scene design rationale. The four scene types in Table 2 are not arbitrary. They are chosen to cover an acoustic gradient that separators must traverse in deployment, with three controlled axes: room volume, reverberation time, and the presence of far-field interferers. Small meeting rooms have the shortest RT60 (0.3–0.6 s), match the scale of two-person to small-group offices, and contain only near-field interferers because no point in the room is beyond the 3 m near-/far-field threshold; this makes them the "clean" end of the gradient. Medium indoor spaces (e.g., kitchens, living rooms) introduce moderate reverberation (0.5–0.9 s) together with the first far-field talkers, since the room becomes large enough for some speakers to fall outside the critical-distance neighborhood of the microphone [28]. Large public spaces such as cafeterias and lecture halls extend RT60 to 1.0–2.0 s, which is the regime in which reverberant energy dominates the direct path and in which far-field speech becomes the harder interferer to suppress. Outdoor environments invert the indoor pattern: very low RT60 (0.05–0.1 s) but high environmental SIR difficulty, so that the stress test moves from reverberation to wide-band ambient noise. Together the four scenes span the regime in which residential and on-the-go AVTSE systems are deployed, and the RT60 ranges fall within the values reported for occupied rooms of comparable size in standard architectural-acoustics references [28].

Near-field overlap design. The overlap-ratio range of 10%–40% used in small-meeting samples is grounded in the speaker-overlap statistics measured in conversational corpora, where the fraction of dyadic-conversation time with simultaneous speech typically falls in this band. We therefore sample partial-overlap intervals only in the small-meeting scene, where short conversational turn-taking is plausible, and apply full-length alignment in larger scenes where the dominant interference pattern is co-occurring background talkers rather than turn-taking. The extended scene-parameter table is given in Table 11; it reproduces the four columns of the in-text Table 2 and additionally specifies the per-scene Near SIR ranges and the applied far-field SIR range of 20–$35\,\mathrm{dB}$ for scenes with far-field interferers.

Table 11: Extended acoustic-scene configuration for RealSSA-Sim. Near speakers and Far speakers denote interfering speakers and exclude the target. Near SIR specifies the scene-dependent range for near-field interferers. Applied Far SIR is 20–$35\,\mathrm{dB}$ in all scenes with far-field interferers; small-meeting scenes contain none. The mixing procedure is detailed in Appendix A.2.

Scene	Space size	RT60	Near speakers	Far speakers	Near SIR	Env. noise SIR	Applied Far SIR	Sudden noise
Small meeting	4-6 × 3-5 × 2.5-3 m	0.3–0.6 s	1–5, 0.5–1.5 m	none	2–5 dB	5–15 dB	–	30%, 1 event
Medium indoor	8-15 × 8-12 × 3-4 m	0.5–0.9 s	1–3, 1.0–2.5 m	10–15, 3–15 m	2–8 dB	5–10 dB	20–35 dB	40%, 1–2 events
Large public	20-40 × 15-30 × 4-8 m	1.0–2.0 s	1–4, 1.0–3.0 m	10–15, 5–20 m	2–8 dB	2–8 dB	20–35 dB	50%, 1–3 events
Outdoor	50 × 50 × 20 m	0.05–0.1 s	1–4, 1.0–4.0 m	10–15, 5–25 m	2–15 dB	−2 to 5 dB	20–35 dB	40%, 1–3 events

Semantic noise selection from Hive. Hive provides a label hierarchy over sound events; we use it to ensure that the noise pool of each scene is semantically consistent with the scene type rather than uniformly drawn. Table 12 lists the environmental and sudden-noise labels assigned to each scene. The mapping follows three rules: continuous mechanical noise (air conditioning, refrigerator, microwave) is restricted to indoor scenes whose physical setting plausibly contains such sources; pedestrian and traffic events (footsteps, sirens, vehicle pass-bys) are restricted to large-public and outdoor scenes; and a small pool of generic transient events (key jangling, coin dropping, slam, bang) is shared across all scenes since they occur ubiquitously regardless of room type. The label-to-scene



(a) Small meeting

(b) Medium indoor

(c) Large public

(d) Outdoor

Figure 5: Spatial layouts of the four RealSSA-Sim acoustic scene types. The blue human icons indicate the positions of near-field speakers, while the green human icons indicate the positions of far-field speakers. The green spheres denote environmental sound sources, the orange spheres denote sudden noise sources, and the black microphone denotes the microphone position. Since different sound sources may have different spatial locations, the visualization icons can appear at different heights.

assignment is fixed in the released metadata and is the only step in the pipeline that draws on human knowledge of the sound-event taxonomy.

A.2 Generation Pipeline

Spatial simulation. Room impulse responses (RIRs) are computed with gpuRIR [31] using the image-source method. For every sample, we draw a room geometry and an RT60 target from the ranges in Table 11, then convert the RT60 target into image-source attenuation parameters using Sabine's estimator at the gpuRIR cut-offs of $15\,\mathrm{dB}$ and $60\,\mathrm{dB}$, which controls when image sources stop contributing direct and reverberant energy. The microphone is modelled as a four-element linear array with $2.8\,\mathrm{cm}$ spacing whose array centre is placed at a height of 1.2–$1.8\,\mathrm{m}$ above the floor; this geometry mirrors the form factor of speakerphones and laptop microphones used in conferencing devices. Although the array is generated to enable future multi-channel extensions, all experiments in the main paper use the first-channel signal only, so RealSSA in its current form is a single-channel benchmark and the released waveforms are the channel-1 streams of the simulated array. Speaker heights are sampled from 1.0–$1.9\,\mathrm{m}$ to span the range of seated and standing posture.

Outdoor simulation. Outdoor scenes are approximated by a $50 \times 50 \times 20\,\mathrm{m}$ enclosure with wall absorption coefficient 0.99, which yields RT60 in 0.05–$0.1\,\mathrm{s}$ and produces an essentially anechoic propagation profile in which the direct path dominates. We adopt this approximation because gpuRIR's image-source model is constructed for closed rooms and does not natively model open-field

Table 12: Hive label assignment per RealSSA-Sim scene. "All scenes (shared)" lists generic transient events that are added to the sudden-noise pool of every scene.

Scene	Environmental noise labels	Sudden-noise labels
Small meeting	air conditioning, fan, clock, typing, printer, keyboard, tick, hiss	writing, clicking, crumple, rustle, knock, telephone, door, drawer, squeak, paper
Medium indoor	microwave, blender, dishes, pots/pans, refrigerator	cutlery, glass, cupboard, frying, cash register, gurgling, steam, kettle, sneeze, cough, laughter
Large public	footsteps, shuffle, speech, babble, crowd	sliding door, siren, child speech, crying, roll, alarm, buzzer, announcement
Outdoor	wind, rain, traffic, stream, city ambience	car, bus, truck, motorcycle, car passing, skidding, reversing beep, horn, police/ambulance/fire siren, dog, bird, cricket, jackhammer, hammer, power tool
All scenes (shared)	–	key jangling, coin dropping, slam, bang, thump, crash, squeal

propagation. The simplification preserves the inverse-distance amplitude law and the geometric arrival times of all sources, which are the dominant cues that an AVTSE model uses, and removes only late reverberation, which is in any case negligible in true outdoor scenes. We make this assumption explicit so that downstream users of RealSSA can interpret outdoor results accordingly.

Mixing pipeline. Algorithm 1 formalizes the per-sample mixing procedure. The target speech is first convolved with the target RIR and loudness-normalized to $-23\,\mathrm{LUFS}$ following the EBU R128 broadcast loudness recommendation [32], which is the standard reference loudness for speech-centric content and removes the residual loudness variability of LRS3. Near-field interferers are added at the per-scene Near SIR ranges in Table 11. For medium-indoor, large-public, and outdoor scenes, far-field interferers are scaled to an applied SIR of 20–35 dB, as listed in the Applied Far SIR column. Small-meeting scenes contain no far-field interferers. Continuous environmental noise is the sum of one WHAM! stream and one Hive ambient stream; when both are present we attenuate each by $3\,\mathrm{dB}$ before summation so that the total noise SIR matches the per-scene environmental-noise SIR specification rather than doubling. Sudden-noise events are inserted at uniformly-sampled non-overlapping positions according to the per-scene event count, with each event scaled to the same SIR distribution as the environmental noise. The final mixture is peak-limited at $-1\,\mathrm{dBFS}$ to prevent digital clipping, and per-source gains are clipped to the range $[-40, +40]\,\mathrm{dB}$ to suppress the rare numerical instability that can arise when an exceptionally quiet source is paired with a high target SIR.

Reproducibility. All randomness is controlled by a fixed master seed of 42, with disjoint seed offsets for the training, validation, and test partitions so that re-running the pipeline regenerates the released splits bit-for-bit. The released metadata records the sampled room geometry, RT60, per-source positions, RIR hashes, SIR values, and Hive label identifiers for each sample. The pipeline depends on `gpuRIR`, `pyloudnorm`, `torchaudio`, and `soundfile`; on a single NVIDIA RTX 3090 the full 100,826-sample training set takes approximately 48h, GPU-hours to generate, which is dominated by the gpuRIR convolution stage. We will release the generation scripts and the metadata together with the dataset.

A.3 Dataset Statistics

Table 13 summarizes the realized statistics of the three splits. The four scene types are sampled in approximately balanced proportions, with each accounting for nearly one quarter of every split, which is reflected in the scene-distribution panel of Figure 6. Figure 6 additionally presents aggregate per-mixture distributions of the total number of speakers, the realized near-field and far-field SIR, and

Algorithm 1 RealSSA-Sim per-sample mixture generation.

Require: Target utterance $\mathbf{s}$ with paired mouth video $\mathbf{m}$ from LRS3
Require: Scene type $c \in \{\text{small_meeting}, \text{medium_indoor}, \text{large_public}, \text{outdoor}\}$
1: Sample room geometry, RT60, and source/microphone positions from the ranges of scene c
2: Compute RIRs $\{\mathbf{h}_\text{s}, \mathbf{h}_1, \ldots, \mathbf{h}_K\}$ with gpuRIR and Sabine attenuation
3: Convolve target with $\mathbf{h}_\text{s}$ and normalize the result to $-23\,\text{LUFS}$
4: Add near-field LibriSpeech interferers at Near SIR; partial overlap of 10%–40% for small_meeting, full-length alignment otherwise
5: For scenes with far-field speakers, add LibriSpeech interferers at an applied SIR of 20–35 dB
6: Add WHAM! continuous noise and one Hive ambient stream at the scene-specific environmental SIR (each attenuated by 3 dB when both are present)
7: Insert per-scene Hive sudden-noise events at random non-overlapping positions
8: Apply per-source gain clipping to $[-40, +40]\,\text{dB}$
9: Peak-limit the mixture at $-1\,\text{dBFS}$
10: **return** mixture waveform, clean target, mouth video, and source-/scene-level metadata

Table 13: Per-split summary statistics of RealSSA-Sim. Total number of speakers includes the target and all near-/far-field interferers. Number of speakers, near-/far-field SIR, and RT60 are sampled per mixture from the ranges of Table 11; the values reported here are realized averages over the released metadata.

Split	Samples	Total hours	Avg. duration	Avg. speakers	Avg. Near SIR	Avg. Applied Far SIR
Train	100,826	129.2	4.61 s	13.21	8.51 dB	27.49 dB
Val	9,133	11.7	4.61 s	13.29	8.51 dB	27.47 dB
Test	9,169	11.7	4.61 s	13.23	8.49 dB	27.50 dB

the RT60 of the simulated rooms. The RT60 histogram exhibits the four-mode pattern induced by the four scene RT60 bands, with the outdoor mode tightly concentrated near 0.05 s and the large-public mode spread between 1.0 and 2.0 s.

A.4 RealSSA-Real: Real-World Evaluation Set

Recording protocol. RealSSA-Real complements the simulated suite and is collected separately, so that no participant or location is reused between the two. Each session is captured by a single Apple iPhone 17 Pro, whose rear camera and built-in microphones acquire video and audio in one device pass at 60 fps and 48 kHz, respectively. The audio is then downsampled to 16 kHz and the video to 25 fps to match the AVTSE pipeline used during training. The iPhone is placed on a table at conversational height in indoor scenes and held at chest level in outdoor scenes, which keeps the device-to-source geometry within the range used in the simulated suite of Appendix A.2. We deliberately do not denoise, dereverberate, or volume-normalize the recordings, so that the evaluation reflects the raw on-device signal that a deployed AVTSE system would receive.

Scenes and contents. The set contains 54 sessions totalling 5,035 s (approximately 84 minutes) of overlapping speech across eight environments that span quiet indoor rooms, crowded public venues, and open streets (Figure 7): a study room, a classroom, an internet cafe, a billiard hall, a shopping mall, a supermarket, and two street scenes captured in daytime and at night. Per-environment session counts and durations are reported in Table 14. Recordings involve 10 participants (5 male and 5 female), with 2–5 participants present in each session. Every participant provided written consent for the recording and for its subsequent academic release, and all participant faces are pixelated prior to release. Each environment carries its own non-stationary noise signature: mall and supermarket sessions combine crowd chatter with venue ambient noise, including noise from nearby diners in the mall and from passing shoppers in the supermarket; billiard-hall sessions add intermittent ball-collision impacts together with background music playing in the venue; internet-cafe sessions contain venue ambient noise layered with crowd chatter; the two street scenes contain vehicular traffic, pedestrian chatter, and occasional music from nearby storefronts; classroom sessions add desk-tapping and footstep events; study-room sessions add coughing and door-opening events. The classroom and study-room sessions each contain 2 near-field talkers, whereas the remaining public

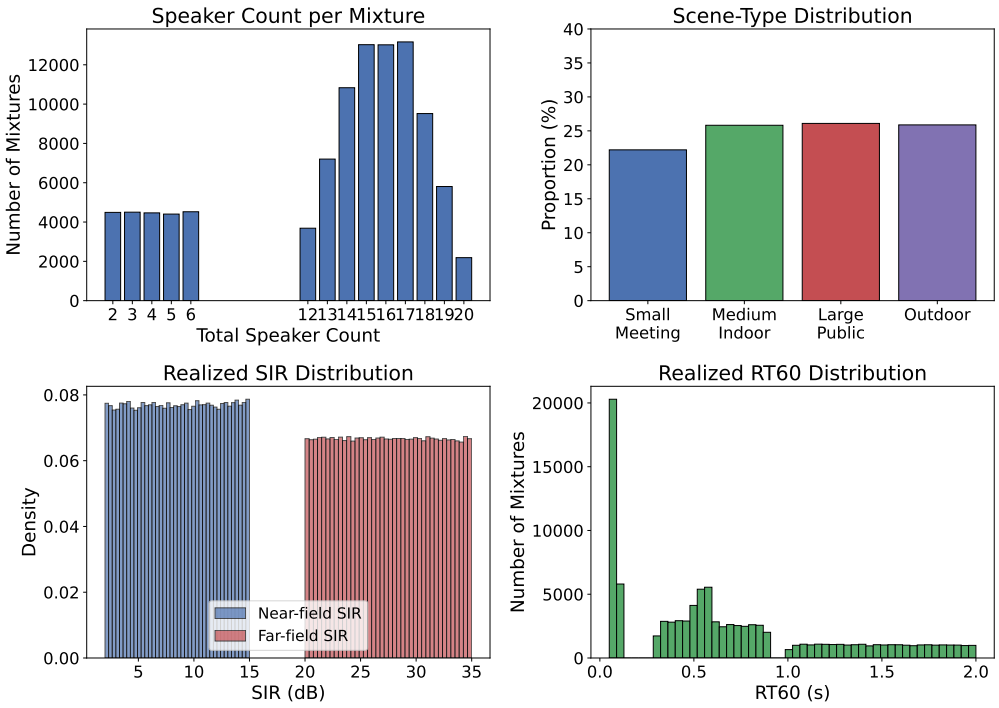


Figure 6: RealSSA-Sim distribution statistics. (a) Total number of speakers per mixture, including the target and all interferers. (b) Scene-type proportion. (c) Realized near-field and far-field SIR distributions. (d) Realized RT60 distribution. Histograms are computed from the released training-split metadata.

and outdoor environments contain 1 near-field talker. Sessions are organized so that the number of simultaneous speakers, the speaker positions, and the ambient-noise sources jointly match the scene categories of Table 11; this matching is used only to organize the test set and is not exposed to the model at inference time.

Table 14: Per-environment composition of RealSSA-Real. Total durations are reported as "minutes:seconds". The environments are grouped by the simulated scene category they correspond to in Table 11.

Simulated scene	Real-world environment	Sessions	Total duration (m:s)
Small meeting	Study room	12	10:36
	Classroom	8	10:16
Medium indoor	Internet cafe	5	10:22
	Billiard hall	5	10:39
Large public	Shopping mall	5	9:48
	Supermarket	5	10:46
Outdoor	Street (daytime)	7	9:57
	Street (night)	7	11:31
Total	–	54	83:55

References and evaluation metrics. We evaluated RealSSA-Real along three complementary dimensions: perceptual quality using DNSMOS [35] and NISQA [36], speech recognition using WER with Mega-ASR [37], and target-speaker similarity using WeSpeaker [38]. RealSSA-Real has no isolated target waveforms. WER and speaker similarity use target transcripts and independent



(a) Study room (b) Classroom (c) Internet cafe (d) Billiard hall

(e) Shopping mall (f) Supermarket (g) Street (daytime) (h) Street (night)

Figure 7: Eight environments covered by RealSSA-Real. All recordings were captured with a single Apple iPhone 17 Pro; detected faces have been pixelated for privacy before release.

enrollment utterances, respectively, with the protocols detailed in Appendix E. RealSSA-Real was held out for evaluation and was never used for training or validation.

Ethics statement. All recordings were collected under written informed consent. Participants were informed of the academic purpose of the data, were free to withdraw their contribution at any time, and were offered the option to redact specific segments before release. Participants received appropriate compensation for their time, at a rate at or above the local minimum hourly wage. The released package contains only the segments for which release consent was confirmed, and all personally identifiable visual information is pixelated prior to release; no speaker names, contact information, or other identifying metadata are included.

B Dataset Setup Details

This appendix complements Section 5.1 of the main text and groups the preprocessing details that are shared across the five evaluation sets.

Public AVTSE benchmarks. We used LRS2 [9], LRS3 [10], and VoxCeleb2 [11] under their official train, validation, and test partitions. To form two-speaker mixtures, we followed the protocol of prior causal AVTSE work [6, 17]. For every utterance we extracted a fixed-length 2-second clip; clips shorter than 2 seconds were zero-padded and clips longer than 2 seconds were randomly cropped during training. Audio was resampled to 16 kHz, and the paired video was decoded to 25 fps so that each clip yielded exactly $T_{\mathrm{v}} = 50$ video frames. We cropped a 96×96 mouth region of interest around the lip centroid and converted it to grayscale, matching the visual front-end used by [6, 17].

RealSSA-Sim. The RealSSA-Sim simulated suite was constructed as described in Section 3 and Appendix A. The training, validation, and test splits contain 100,826, 9,133, and 9,169 samples

respectively (Appendix A.3). Audio sampling rate, video frame rate, and mouth ROI followed the same configuration as the public benchmarks, so that a single Falcon checkpoint can be evaluated across all five sets without front-end re-tuning.

RealSSA-Real. Recording protocol, scene composition, and annotation procedure for RealSSA-Real are documented in Appendix A.4. The set was held out for evaluation only and was never used for training or validation. The same 96×96 mouth ROI extraction and $16\,\mathrm{kHz}$ audio resampling were applied so that the input format matches that of the simulated splits.

C Falcon Architecture Widths

This appendix complements Section 5.2 of the main text and maps every channel-width symbol introduced in Sections 4.1 and 4.2 to its concrete value (Table 15).

Table 15: Channel-width settings of Falcon. Symbols follow the definitions in Sections 4.1 and 4.2.

Symbol	Module	Value
C_{e}	STFT encoder output channels	112
C	Audio bottleneck output channels	112
C_{h}	Separator hidden channels	64
C_{v}	Visual hidden channels	32
$\dim(\mathbf{q})$	Speaker embedding dimension	64
$h_{\mathcal{F}}$	TFRB frequency-LSTM hidden size	24
$h_{\mathcal{T}}$	TFRB time-LSTM hidden size	72
n_{head}	TFRB attention heads	4
L	Separator levels	4

D Falcon Training Protocol

Table 16 lists the full training configuration. We used the same FRCNN visual backbone as CTC-Net [6], initialized from a publicly released lipreading-pretrained checkpoint and frozen throughout AVTSE training. The contrastive temperature τ was linearly annealed from 0 to 1 over the first 1000 training epochs and held at 1 thereafter, in line with the schedule of Section 5.2 and the analysis of Table 9. The reconstruction loss $\mathcal{L}_{\text{recon}}$ uses the negative SI-SNR formulation defined in Section 5.2, and the contrastive weight was fixed at $\lambda_{\text{spk}} = 1.0$ as reported in Section 5.2.

E Evaluation Metrics and Inference Latency Protocol

Target speech extraction quality. On the simulated benchmarks (LRS2-2Mix, LRS3-2Mix, VoxCeleb2-2Mix, and RealSSA-Sim), clean references are available and we reported scale-invariant signal-to-noise ratio improvement (SI-SNRi) and signal-to-distortion ratio improvement (SDRi) [34], both in dB, relative to the unprocessed mixture.

Perceptual quality on real recordings. We assessed the perceptual quality of extracted speech on RealSSA-Real with DNSMOS [35] and NISQA [36]. These non-intrusive predictors estimate perceived speech quality directly from the extracted waveform, complementing the speech recognizability and target-speaker consistency measured by WER and speaker similarity, respectively.

Speech recognition. We used Mega-ASR [37] to transcribe the extracted speech and compute word error rate (WER) against the target transcript. WER is the number of substituted, deleted, and inserted words divided by the number of reference words, reported as a percentage. Evaluation covered all 9,169 RealSSA-Sim test samples and all 54 RealSSA-Real recordings.

Target-speaker similarity. We extracted speaker embeddings using the ResNet34-LM checkpoint released by WeSpeaker [38][4]. This model was trained on VoxCeleb2 and further fine-tuned with a

[4]Official checkpoint: `https://huggingface.co/Wespeaker/wespeaker-voxceleb-resnet34-LM`.

Table 16: Training protocol of Falcon.

Item	Value
Optimizer	AdamW, learning rate 1×10^{-3}, weight decay 0.01
LR scheduler	ReduceLROnPlateau, factor 0.5, patience 15
Early stopping	Patience 30 on validation loss
Max epochs	1000
Batch size per GPU	4
Hardware	$8\times$ NVIDIA RTX 3090
Audio segment length	$2\,\mathrm{s}$ at $16\,\mathrm{kHz}$
Video frame rate	25 fps, $T_{\mathrm{v}} = 50$ frames
Mouth ROI	96×96 grayscale
STFT	Window 256, hop 128 ($8\,\mathrm{ms}$ at $16\,\mathrm{kHz}$), Hann
Encoder input channels	Real, imaginary, magnitude (3 channels)
Visual backbone	FRCNN (as in CTCNet), frozen, lipreading-pretrained
Reconstruction loss	$\mathcal{L}_{\mathrm{recon}} = -\mathrm{SI\text{-}SNR}(\hat{\mathbf{s}}, \mathbf{s})$
Contrastive loss weight λ_{spk}	1.0
Contrastive margin m	0.5
τ schedule	Linear $0 \to 1$ over first 1000 epochs, held at 1 thereafter
Random seed	42

large-margin objective. For each test sample, we computed the cosine similarity between the extracted speech and an independent enrollment utterance from the same target speaker. The enrollment utterance differed from the test utterance and was used only for evaluation. We reported the mean cosine similarity over each test set, multiplied by 100. This evaluation encoder is separate from Falcon's audio encoder.

Ground-truth reference scores. Evaluating the ground-truth target speech on RealSSA-Sim yielded 11.83% WER and 64.69 speaker similarity under the same protocol. Since similarity was measured against an independent utterance from the same speaker, the ground-truth score provided an empirical reference for interpreting the absolute values. Falcon achieved 43.05, leaving a gap of 21.64 points to this reference. RealSSA-Real has no isolated target waveforms, so corresponding ground-truth scores are unavailable.

Computational efficiency. We reported trainable parameters, multiply-accumulate operations (MACs), CPU and GPU forward-pass times, and memory usage. Each measurement used a $2\,\mathrm{s}$, $16\,\mathrm{kHz}$ audio clip paired with 50 mouth-crop frames at 25 fps and a batch size of one. The hardware was an NVIDIA RTX 3090 GPU and an Intel Xeon Gold 6226R CPU. CPU measurements used one execution thread. We discarded five warmup runs and reported the median of the following 50 forward passes. GPU time was measured with CUDA events, excluding host-side overhead. Memory measurements reported system RAM during CPU inference and peak GPU memory during GPU inference under the same protocol. Falcon required $51.3\,\mathrm{ms}$ on the GPU and $1361.2\,\mathrm{ms}$ on the CPU for the $2\,\mathrm{s}$ input.

Lookahead and streaming schedule. At $25\,\mathrm{fps}$, the visual stream buffers three future frames, corresponding to $120\,\mathrm{ms}$ of lookahead. The audio STFT uses a 256-sample window and a 128-sample hop, corresponding to $16\,\mathrm{ms}$ and $8\,\mathrm{ms}$ at $16\,\mathrm{kHz}$. The audio and visual timelines are aligned, the output advances at the $8\,\mathrm{ms}$ hop, and the configuration uses no additional future-chunk buffer.

Dividing the measured forward-pass times by approximately 250 hops per $2\,\mathrm{s}$ input gives an amortized cost of $0.21\,\mathrm{ms}$ per hop on the GPU and $5.44\,\mathrm{ms}$ per hop on the CPU. These values describe average computation amortized over a complete input clip. Per-hop execution overhead and end-to-end latency require measurement with the streaming implementation, including input availability, window alignment, and output scheduling.

F Causalization of Dolphin and AV-Mossformer2 Baselines

Dolphin [20] and AV-Mossformer2 [15] were originally released as non-causal AVTSE models. To compare them against Falcon under a common causal protocol, we re-implemented both baselines with the minimal modifications required to remove future-context access at every layer, while preserving their parameter counts, block topologies, and training objectives. Both baselines used the common optimizer and learning-rate schedule described in Section 5.2. This appendix lists, for each model, the operations that needed rewriting; every module not listed is already causal in the released code and was kept unchanged.

Across both models we used four rewriting primitives [17]. *Causal self-attention* adds a lower-triangular mask before the softmax. *Causal convolution* replaces symmetric padding by left padding only, with the same kernel size and dilation. *Causal pooling and interpolation* replaces full-axis adaptive average pooling and nearest-neighbor interpolation along time by their one-sided (past-only) variants. *Streaming-causal layer normalization* replaces full-axis LayerNorm or GroupNorm along time by the frame-local $\mathrm{LN_c}$ of Section 4.2. Frame-wise operators, including pointwise convolutions, element-wise gates, and LayerNorm on the feature axis, were kept as released.

F.1 Dolphin Causalization

Visual encoder and refinement. The DP-LipCoder visual encoder [20] was frozen during AVTSE training, as in the original paper. Inside it we added causal masks to the temporal axis of every dual-path attention block, while the spatial axes remain bidirectional, and we rewrote every temporal downsample and upsample as a causal convolution. The post-encoder visual refinement block uses dilated depthwise convolutions with symmetric padding, which we rewrote as causal convolutions.

Audio-Visual Fusion. The fusion module has three paths: a resize path that upsamples the visual feature to the audio frame rate by nearest-neighbor interpolation, a key-value path that multiplies a pointwise-projected audio feature by the upsampled visual feature, and an attention path that softmaxes a projected visual feature over its time axis and upsamples the result to the audio time axis. The two non-causal operations are the nearest-neighbor upsamplings, which we rewrote as one-sided, and the softmax over the visual time axis, which we rewrote as a causal softmax that masks visual frames later than the current audio frame. The element-wise sum of the three paths is unchanged.

U-shape Separator. The Separator alternates Global Attention and Local Attention blocks across encoder and decoder stages, joined by injection-based skip fusion and a global Bottleneck. Global Attention compresses the input by adaptive average pooling along time, applies relative-position multi-head self-attention, and restores the resolution by nearest-neighbor upsampling; we rewrote the pooling and the upsampling as one-sided, added a causal mask to the inner attention, and tightened the relative-position clip to non-positive offsets. Local Attention contains Heat Diffusion Attention, which as released applies a discrete cosine transform (DCT) along the full time axis, a per-frequency exponential decay, and an inverse DCT. We replaced the full-sequence DCT pair with a causal windowed DCT on a past-only window of length W, so that the heat-diffusion interpretation is restricted to a finite past and the transform pair becomes a causal convolution of effective kernel W. The surrounding depthwise convolutions and GroupNorm layers in this block were rewritten as causal convolutions and as $\mathrm{LN_c}$, respectively. The Bottleneck aggregates encoder-stage outputs by adaptive average pooling along time before applying multi-head self-attention and a feed-forward network; we rewrote the aggregation pool as one-sided, added a causal mask to the attention, and rewrote the feed-forward depthwise convolution as causal. The injection-based skip fusion modules align the global feature to each skip's resolution by nearest-neighbor interpolation along time, which we rewrote as one-sided; the projection sub-modules inside the injection module reuse Heat Diffusion Attention and inherit its causal windowed DCT rewrite.

Audio encoder and decoder. The strided 1D audio encoder and its transposed-convolution decoder were rewritten as causal convolutions along the time axis.

F.2 AV-Mossformer2 Causalization

Pre-separator audio-visual fusion. The visual encoder was frozen as in the original paper [15]. The pre-separator fusion layer upsamples the visual feature to the audio frame rate by nearest-neighbor interpolation, which we rewrote as one-sided; the pointwise projection and the element-wise gated sum that follow are unchanged.

MossFormer block. Each MossFormer block runs a quadratic intra-group attention with squared-ReLU scores in parallel with a linearized cross-group attention. The released code already exposes a causal switch: when enabled, it adds a strictly upper-triangular mask to the squared-ReLU score tensor inside each group, and replaces the bidirectional cross-group statistics by one-sided cumulative sums followed by a one-step shift that excludes the current group from its own summary. We enabled this switch on every MossFormer block, and no further modification was required to the joint attention path.

Gated FSMN. The Gated FSMN module is already causal in the release: its memory taps span only past frames, the surrounding pointwise convolutions have time receptive field 1, and the cumulative LayerNorm computes statistics only over past frames. The block was therefore kept unchanged.

G Per-Scene and Per-Speaker-Count Breakdown on RealSSA-Sim

This appendix complements the aggregate RealSSA-Sim column of Table 3 by resolving separation quality along the two axes that RealSSA-Sim was designed to vary jointly, namely the acoustic scene type and the number of near-field interfering speakers. Table 17 reports SI-SNRi in decibels for every causal baseline and for Falcon on each $(\text{scene}, N_{\text{near}})$ cell of the RealSSA-Sim test split, where N_{near} denotes the number of near-field interfering speakers observed by the microphone, excluding the target. The cells of the grid exactly match the near-field interfering-speaker ranges listed per scene in Table 11, so that $N_{\text{near}} = 1$ corresponds to a mixture with one near-field interferer in addition to the target. Unlike the aggregate numbers of Table 3, this breakdown exposes the interaction between room acoustics and near-field crowding that a single RealSSA-Sim score averages over.

Table 17: Per-scene and per-speaker-count SI-SNRi (dB, ↑) on the RealSSA-Sim test split. N_{near} denotes the number of near-field interfering speakers in the mixture, excluding the target, so that $N_{\text{near}} = 1$ is the one-interferer case. Columns group the four RealSSA-Sim scene types of Table 11. For each column, **bold** marks the best result and underline marks the second-best.

Method	Small meeting					Medium indoor			Large public				Outdoor			
	1	2	3	4	5	1	2	3	1	2	3	4	1	2	3	4
AV-ConvTasNet [39]	−0.6	0.1	0.4	0.7	0.7	1.6	1.5	1.5	1.3	1.4	1.6	1.5	0.9	1.5	2.0	1.9
AV-DPRNN [26]	4.0	4.4	4.3	4.3	4.1	5.1	4.9	4.8	5.4	5.4	5.5	5.6	6.6	6.8	7.0	7.1
AV-Sepformer [27]	4.0	4.4	4.3	4.4	4.2	4.9	4.8	4.7	5.4	5.4	5.6	5.6	7.1	7.3	7.5	7.4
CTCNet [6]	4.0	4.4	4.4	4.5	4.3	5.0	4.9	4.8	5.3	5.3	5.5	5.7	6.6	6.9	7.1	7.3
RTFSNet-12 [16]	2.5	3.4	3.5	3.9	3.7	4.5	4.6	4.5	4.7	4.9	4.9	5.3	4.8	5.4	5.8	6.0
AV-Mossformer2 [15]	4.6	5.0	5.0	5.0	4.7	5.4	5.3	5.1	5.8	5.8	6.0	6.1	7.7	8.0	7.9	8.0
Dolphin [20]	5.1	5.6	5.5	5.5	5.3	6.3	6.2	6.0	6.8	7.0	7.1	7.3	8.7	9.1	9.2	9.4
AV-TF-GridNet [14]	6.4	6.8	7.0	7.0	**6.9**	7.1	7.0	6.7	7.5	**7.6**	**7.8**	**8.0**	8.8	9.1	**9.4**	9.4
Swift-Net-12 [17]	6.1	6.5	6.6	6.6	6.3	6.9	6.9	6.7	7.2	7.3	7.3	7.5	8.5	8.7	8.8	8.8
Falcon (Ours)	**6.7**	**7.0**	**7.1**	**7.1**	6.8	**7.2**	**7.2**	**7.1**	**7.6**	7.6	7.8	7.8	**9.0**	**9.1**	9.3	**9.4**

Three patterns emerge from the grid. The scene axis did not follow a simple monotone difficulty ordering with respect to reverberation time. Outdoor cells consistently yielded the highest SI-SNRi for every architecture because the near-anechoic propagation of Table 11 leaves the target direct path largely intact, so that the separator only needs to suppress wide-band ambient noise and a small number of geometrically well separated interferers. Large public cells were the next easiest on average, since the dominant interference is reverberant rather than spatially co-located with the target and the visual enrollment cue is sufficient to disambiguate far-field talkers. Medium indoor cells lay in the middle and combine moderate reverberation with the first far-field interferers. Small meeting cells were the hardest, because the near-field interferers lie inside the critical distance of the microphone and therefore inject energy whose spectral envelope, onset timing, and reverberation fingerprint closely match those of the target. The fact that a low-reverberation outdoor scene produces

higher SI-SNRi than a short-RT60 small meeting confirms that crowding of near-field talkers, rather than reverberation alone, is the dominant difficulty axis on RealSSA-Sim.

The speaker-count axis followed a non-monotone trajectory that is characteristic of audio-visual target speaker extraction rather than of blind speech separation. In the outdoor scene every competitive model saw its SI-SNRi rise as N_{near} grows from one to four, for instance Falcon improved from 9.0 dB at $N_{\text{near}} = 1$ to 9.4 dB at $N_{\text{near}} = 4$, which is consistent with the input mixture becoming progressively more distorted and therefore admitting a larger improvement margin before the metric saturates. In the small meeting scene the improvement peaked around three or four near-field speakers and then declined, with Falcon rising from 6.7 dB at $N_{\text{near}} = 1$ to 7.1 dB at $N_{\text{near}} = 3$ and $N_{\text{near}} = 4$ before dropping to 6.8 dB at $N_{\text{near}} = 5$. This non-monotonicity arises because once a small room contains four or more nearby talkers with closely matched reverberation fingerprints, the visual enrollment cue remains able to resolve the target identity, but the total near-field interfering energy begins to exceed the budget that a single-channel mask can suppress while still preserving the target spectrum. The aggregate RealSSA-Sim column of Table 3 averages over this behavior and therefore underreports the cell-level gap that separates strong causal architectures from lightweight ones in the crowded-room regime.

The grid further shows that Falcon extended the quality of heavyweight causal baselines to conditions in which lightweight ones collapsed. Falcon obtained the best SI-SNRi on eleven of the sixteen cells and the second-best on five more. In the small-meeting and medium-indoor cells, its SI-SNRi margin over AV-TF-GridNet ranged from -0.1 to $+0.4$ dB despite using 91.9% fewer parameters (Table 5); its margin over Swift-Net-12 was 0.3 to 0.6 dB at the same parameter count. The lightweight iterative baselines lost ground most sharply on this same column, with RTFSNet-12 falling below 4 dB on every small meeting cell and AV-ConvTasNet dropping below 1 dB, which indicates that repeated application of a small separator cannot by itself compensate for the spectrally similar near-field interference that small rooms produce. The single-pass multi-scale separator and the encoder-space speaker contrastive loss of Falcon therefore act where they were designed to, namely in the crowded near-field cells that the aggregate RealSSA-Sim score tends to wash out, while remaining within 0.2 dB of the strongest heavyweight baseline in the reverberation-dominated large public and outdoor cells.

H Qualitative Target Speech Extraction Results

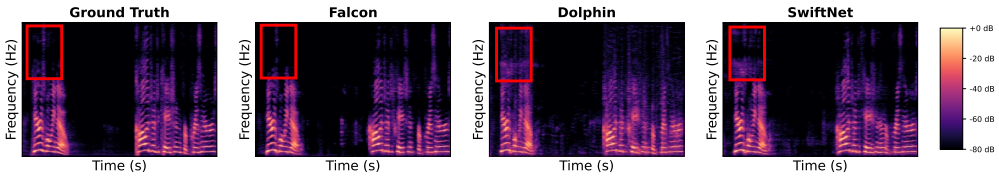
We conducted a qualitative spectrogram analysis of target speech extraction results. The models were evaluated on representative test mixtures, and the results are shown in Figure 8. For each example, we compared the ground-truth target speech with the outputs of Falcon, Dolphin, and SwiftNet. The highlighted regions mark representative time-frequency areas where the methods differ in preserving target speech components and suppressing residual interference.

In Figure 8a, the highlighted region contains a target speech component that is clearly visible in the ground-truth spectrogram. Falcon preserved this component with a structure closer to the ground truth, while keeping the surrounding non-target regions relatively suppressed. Dolphin retained more diffuse residual energy, and SwiftNet showed stronger residual interference around the highlighted region.

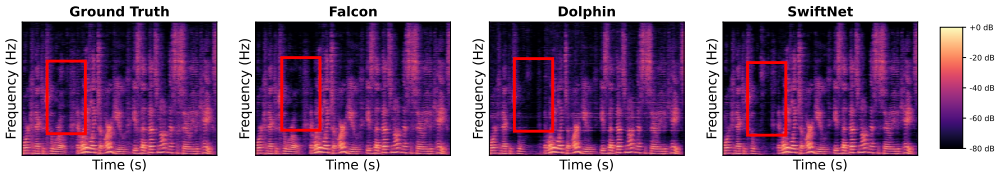
In Figure 8b, the mixture contains dense time-frequency patterns, making target extraction more challenging. Falcon better matched the ground-truth structure in the highlighted region and suppressed surrounding interference more effectively. By contrast, Dolphin and SwiftNet left more residual energy, especially around the low-frequency and neighboring time regions.

In Figure 8c, the target speech component is relatively narrow and localized. Falcon preserved this localized structure more faithfully and maintained cleaner background regions. Dolphin and SwiftNet recovered the main component, but their outputs contained more residual energy from nearby speech components, indicating weaker suppression of interfering speakers.

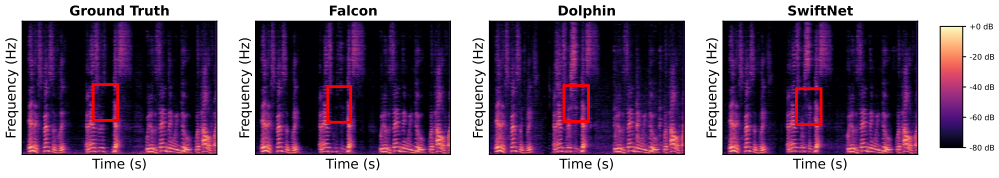
In Figure 8d, the target speech shows a clear harmonic pattern. Falcon reconstructed a more coherent harmonic structure and better suppressed non-target components. Dolphin produced a more diffuse spectrogram with noticeable residual interference, while SwiftNet tended to over-suppress parts of the speech structure and still left residual artifacts.



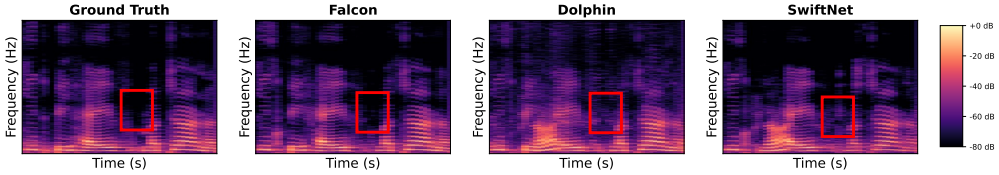
(a) Demo I



(b) Demo II



(c) Demo III



(d) Demo IV

Figure 8: Qualitative spectrogram comparison of target speech extraction results on four representative test mixtures. Each example compares the ground-truth target speech with the outputs of Falcon, Dolphin, and SwiftNet. Red boxes highlight regions where the methods differ in preserving target speech components or suppressing residual interference.

Across all four examples, Falcon produced spectrograms that were closer to the ground-truth target speech, with clearer target structures and less residual interference than the compared baselines.

I Recurrent Unit inside TFRB

Each TFRB operates both at the bottleneck and at the decoder levels (Section 4.2), so its recurrent unit must retain long-range context at low resolutions and preserve fine spectral structure after fusing multi-scale skip features. Table 18 compares four candidates. LSTM achieved the highest SI-SNRi (7.7 dB), exceeding GRU by 0.4 dB at the cost of 0.05 M parameters and 1.27 G MACs; its explicit cell state decouples long-term retention from short-term gating, which helps when the same module spans resolutions differing by factors of two to four. SRU matched GRU in quality with fewer parameters, but neither closed the gap to LSTM.

J Visualization of Encoder-Space Speaker Embeddings

To further analyze the effect of the proposed encoder-space speaker contrastive loss, we visualized speaker embeddings extracted from models trained with and without this loss. We selected the top-15 target speakers with the most utterances from the RealSSA-Sim training set, resulting in 5–8 utterances per speaker. For each utterance, we extracted an encoder-space speaker embedding using the speaker pooling head and projected all embeddings into two dimensions using principal component analysis (PCA).

Table 18: Recurrent unit comparison in TFRB. LSTM achieved the highest SI-SNRi with moderate parameter and computational overhead.

Recurrent Unit	SI-SNRi ↑	SDRi ↑	Params (M) ↓	MACs (G) ↓
RNN	6.2	7.1	0.08	1.90
SRU	7.4	7.9	0.15	2.71
GRU	7.3	7.9	0.20	4.42
LSTM (Ours)	**7.7**	**8.2**	0.25	5.69

Table 19: Comparison of output formulations on RealSSA-Sim. SI-SNRi and SDRi are reported in dB; higher is better. Bold indicates the best result.

Output formulation	SI-SNRi ↑	SDRi ↑
Complex-valued mask (Falcon)	**7.7**	**8.2**
Real-valued mask	7.3	7.9
Direct real/imaginary prediction	7.1	7.2

As shown in Figure 9, without the contrastive loss, embeddings from different speakers were scattered and overlapped with each other, indicating that the encoder-space representations were weakly structured with respect to speaker identity. In contrast, with the proposed contrastive loss, same-speaker embeddings became more compact and different speakers were more clearly separated. This visualization supports that the proposed loss introduces a speaker-discriminative structure in the encoder space, which helps the separator distinguish the target speaker from near-field interferers.

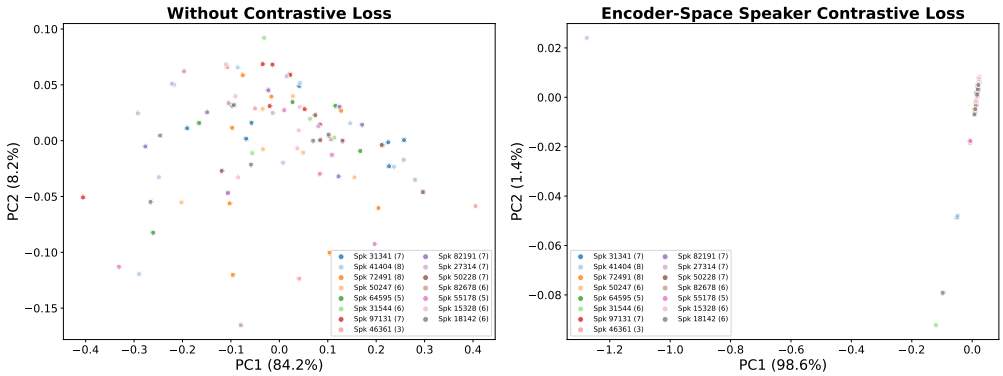


Figure 9: PCA visualization of encoder-space speaker embeddings for the top-15 target speakers in the RealSSA-Sim training set. Left: without the proposed contrastive loss, embeddings from different speakers are scattered and overlap. Right: with the contrastive loss, same-speaker embeddings become more compact and different speakers are more clearly separated.

K Output Formulation Comparison

We compared a complex-valued mask, a real-valued mask, and direct prediction of real and imaginary components on RealSSA-Sim. The backbone, training configuration, and evaluation protocol were kept the same across the three output formulations. Table 19 reports the results.

The complex-valued mask improved SI-SNRi and SDRi by 0.4 dB and 0.3 dB over the real-valued mask, and by 0.6 dB and 1.0 dB over direct real/imaginary prediction. These results support the complex-valued mask formulation used by Falcon under the evaluated model and training configuration.

Table 20: Falcon's performance across speech overlap ratios on RealSSA-Sim. SI-SNRi and SDRi are reported in dB.

Overlap range	Samples	Mean overlap	SI-SNRi ↑	SDRi ↑
0.11–0.72	1,834	0.635	7.95	8.39
0.72–0.79	1,834	0.756	7.71	8.30
0.79–0.84	1,834	0.811	7.80	8.33
0.84–0.89	1,834	0.863	7.73	8.28
0.89–1.00	1,833	0.946	7.73	8.22

Table 21: Performance under missing visual information on 500 RealSSA-Sim test samples. Each entry reports SI-SNRi / SDRi in dB. The $p = 0$ column is the uncorrupted control for this subset. At $p = 1$, the two missing patterns are equivalent. Bold indicates the best result for each metric under the same pattern and missing ratio.

Method	Missing pattern	$p = 0$	$p = 0.25$	$p = 0.50$	$p = 0.75$	$p = 1.00$
AV-Sepformer [27]	Random Drop	5.26 / 5.88	4.98 / 5.68	4.57 / 5.38	2.35 / 4.02	−44.09 / −4.66
AV-Sepformer [27]	Block Drop	5.26 / 5.88	2.42 / 3.65	−4.17 / −0.58	−12.42 / −3.46	−44.09 / −4.66
RTFSNet [16]	Random Drop	4.01 / 4.61	3.87 / 4.50	3.42 / 4.15	1.69 / 3.02	−10.80 / −1.73
RTFSNet [16]	Block Drop	4.01 / 4.61	2.77 / 3.66	0.51 / 2.20	−3.38 / 0.31	−10.80 / −1.73
CTCNet [6]	Random Drop	5.07 / 5.72	4.83 / 5.53	4.30 / 5.09	2.90 / 4.03	−14.85 / −3.72
CTCNet [6]	Block Drop	5.07 / 5.72	3.04 / 4.10	−0.28 / 1.76	−4.01 / −0.39	−14.85 / −3.72
Falcon	Random Drop	**7.48 / 7.88**	**7.28 / 7.72**	**6.75 / 7.30**	**5.97 / 6.73**	−9.91 **/ 0.13**
Falcon	Block Drop	**7.48 / 7.88**	**6.23 / 6.84**	**3.46 / 4.78**	**0.01 / 2.74**	−9.91 **/ 0.13**

L Performance across Speech Overlap Ratios

We grouped all 9,169 RealSSA-Sim test samples by speech overlap ratio to examine extraction performance at different degrees of simultaneous speech. A higher overlap ratio indicates that multiple speakers are active simultaneously for a larger fraction of the utterance. Table 20 gives the sample count, mean overlap ratio, and Falcon's extraction performance for each group.

Across the five groups, Falcon's SI-SNRi ranged from 7.71 to 7.95 dB and SDRi from 8.22 to 8.39 dB. In the highest-overlap group, whose mean overlap ratio is 0.946, Falcon achieved 7.73 dB SI-SNRi and 8.22 dB SDRi. These scores were 0.22 dB and 0.17 dB below the lowest-overlap group, respectively, showing similar extraction improvements across the reported groups.

M Missing Visual Information

We evaluated sensitivity to missing visual information on a fixed random subset of 500 RealSSA-Sim test samples. The missing ratio p was set to 0, 0.25, 0.50, 0.75, or 1.00. Random Drop removes visual frames at random to simulate intermittent information loss, whereas Block Drop masks a continuous segment to simulate sustained information loss. All methods used the same samples and missing-information settings.

Falcon achieved the highest SI-SNRi and SDRi in all evaluated partial-missing conditions. Under Random Drop, its performance decreased gradually as the missing ratio increased, retaining 5.97 dB SI-SNRi when 75% of the visual frames were missing. At the same ratio, Block Drop reduced SI-SNRi to 0.01 dB. Sustained visual loss therefore had a larger effect on extraction performance than dispersed frame loss in this evaluation.

When visual information was entirely absent, all compared methods degraded, and Falcon's SI-SNRi fell to −9.91 dB. The current systems depend on visual information to identify the target speaker. These experiments characterize intermittent and sustained loss of visual information; sensitivity to mouth-crop displacement and tracking a different speaker remains to be evaluated separately.

N LLM Usage Statement

We used large language models as general-purpose writing assistants for polishing sentence-level English phrasing in the manuscript. All scientific contributions, including the definition of the RealSSA benchmark, the design of the Separator and the encoder-space speaker contrastive loss, the experimental protocol, and the analysis of results, were conceived, implemented, verified, and written by the authors. No LLM was used to generate experimental results, to draft novel technical claims, or to position the work against related literature. The authors take full responsibility for the content of the paper, including any residual errors.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the two contributions claimed in the paper: the RealSSA benchmark with a real-recorded evaluation split, and Falcon's single-pass multi-scale separator with the encoder-space speaker contrastive loss. The reported efficiency and quality gains match the quantitative results in Tables 3 and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated Limitations section is provided in Conclusions, covering single-channel operation, the outdoor simulation approximation, the scale of the real-world evaluation set, and the supervised requirement of the contrastive loss.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include formal theoretical results; the contributions are a benchmark dataset, an architecture, and a training loss, all evaluated empirically.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The RealSSA construction pipeline is specified in Section 3 and Appendix A with a fixed master seed, disjoint per-split offsets, and released metadata. Falcon's architecture (Section 4.1–4.2), channel widths (Appendix C), training protocol (Appendix D), and evaluation protocol (Appendix E) are reported with all hyperparameters necessary to reproduce the main results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: An anonymized demo page with code and sample data is linked in the abstract (`https://submission-falcon.github.io/demo/`), and Appendix A.2 states that the RealSSA generation scripts and metadata will be released alongside the dataset. All source corpora used by RealSSA are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Data splits, optimizer, learning-rate schedule, early-stopping policy, batch size, segment length, and loss weights are reported in Section 5.2 and Appendix D; per-split RealSSA statistics appear in Appendix A.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: We did not report error bars because training each causal AVTSE model on RealSSA and on the three public corpora already requires multiple days on 8×RTX 3090, and the consistent cross-architecture and cross-dataset ordering (Tables 3, 6, 8) together with the wide SI-SNRi gaps provides evidence that the observed trends are not attributable to run-to-run variance.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Training hardware (8×NVIDIA RTX 3090) is reported in Appendix D, dataset-generation compute (approximately 48 GPU-hours on a single RTX 3090) in Appendix A.2, and the inference-latency measurement protocol (single RTX 3090 plus Intel Xeon CPU at batch size one) in Appendix E and Table 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. For the real-recorded split of RealSSA (Appendix A.4), every participant provided written informed consent prior to recording, was free to withdraw at any time, could redact segments before release, and received compensation at or above the local minimum hourly wage. No personally identifiable metadata is released. The simulated split is built from publicly available corpora (LRS3, LibriSpeech, WHAM!, Hive) used under their original academic terms.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Low-latency on-device AVTSE benefits accessibility applications such as hearing-assistive devices and video conferencing in noisy environments. On the negative side, audio-visual target-speaker extraction could be misused for selective eavesdropping on specific individuals in multi-talker recordings; the ethics statement in Appendix A.4 restricts our real-world release to consented participants to mitigate misuse on our data side.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The real-recorded split of RealSSA contains data of real people and therefore receives the strictest handling (Appendix A.4): release is limited to the segments for which written release consent was confirmed, identifiable faces are pixelated prior to release, and no names, contact information, or other identifying metadata are included. The set is distributed as an evaluation-only resource without train/test partitioning to discourage derivative speaker-identification uses. The Falcon model is a task-specific speech-extraction system rather than a generative model and carries a correspondingly lower misuse profile than generative pre-trained models.

    Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external corpora and toolkits (LRS2, LRS3, VoxCeleb2, LibriSpeech, WHAM!, Hive, gpuRIR, the FRCNN visual backbone, and the causal baselines) are cited at their point of use and are used under the academic terms granted by their original licenses. We use each corpus only for the role summarized in Appendix A.1 and do not redistribute any source waveforms outside the scope of those licenses. No source corpus used in this paper is on the NeurIPS list of deprecated datasets; the four RealSSA source corpora were actively maintained at the time of submission.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets are the RealSSA simulated benchmark and its real-recorded evaluation split. The source corpora, scene design, generation pipeline, per-split statistics, recording protocol, scene composition, and compensation and consent policy are documented in Appendix A. Participants of the real-recorded split provided written consent specifically covering academic release; the released metadata records the sampled room geometry, RT60, per-source positions, RIR hashes, SIR values, and Hive label identifiers, and contains no personally identifiable information. The released demographic composition (ten participants, five male and five female, across eight environments) and its scope are disclosed in Appendix A.4 so that users can judge representativeness; we make no claim of universal coverage.

    Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The real-recorded split of RealSSA is research involving human participants rather than crowdsourcing. Participants were briefed on the academic purpose of the recordings, on how the audio and video would be used, on the pixelation of faces prior to release, and on their right to withdraw their contribution or redact specific segments at any time before release. Compensation was paid at a rate at or above the local minimum hourly wage. Participant-facing materials and compensation details are summarized in Appendix A.4.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: The recording procedure carries only minimal foreseeable risk: participants held a short, scripted conversation in publicly accessible or shared spaces with a hand-held smartphone, with no physical intervention and no sensitive topic. The residual privacy risk, that released audio-visual segments could identify a participant, was disclosed in the consent form and mitigated by per-segment release consent, face pixelation prior to release, and exclusion of all identifying metadata (Appendix A.4). Because no formal IRB process was required for this type of minimal-risk data collection at our institution, the protocol was reviewed via an equivalent informal process, consistent with the NeurIPS Code of Ethics guidance for settings without a formal process. Details of the reviewing body are withheld at submission time to preserve anonymity.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for editing English phrasing and did not contribute to the core methodology, experimental design, or analysis, as declared in Appendix N.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.