CARTSE Submission to the REAL-TSE Challenge Track 1: Online Target Speaker Extraction

Li Li, Shogo Seki
CyberAgent, Inc., Japan

***Abstract*—This paper describes the CARTSE system submitted to the online track (Track 1) of the Real-World Target Speaker Extraction (REAL-TSE) Challenge. The system is a streaming band-split RNN (BSRNN) separator conditioned on a fine-tuned ECAPA-TDNN speaker encoder through a time-frequency enrolment map and frame-level context fusion. Training has two phases: pre-training on simulated mixtures, then multi-stage fine-tuning on real meeting recordings whose quality-filtered pseudo-label targets are generated by a separate non-causal (offline) teacher model. To respect the online constraint the separator uses a unidirectional temporal RNN, and we verify its effective latency to be approximately 22.9 ms, well within the challenge's 100 ms bound. On the official leaderboard our system reaches a token error rate (TER) of 0.699, a voice-activity timing F1 of 0.848, a speaker similarity (SIM) of 0.504, and a DNSMOS P.835 overall (OVRL) score of 3.436.**

***Index Terms*—target speaker extraction, band-split RNN, pseudo-labeling, data filtering, streaming**

I. Introduction

Target speaker extraction (TSE) isolates the speech of an enrolled speaker from a mixture, guided by an enrolment utterance of that speaker [1]. The Real-World Target Speaker Extraction (REAL-TSE) Challenge moves TSE from simulated mixtures to genuine far-field meeting and in-the-wild recordings, where the enrolment and the mixture are frequently captured by *different devices* (microphone array, headset, or personal phone), and where the target speaker may be silent for long intervals. These conditions break the clean-reference assumptions of classical TSE training. Track 1, the online track, additionally requires a streaming, low-latency separator that does not rely on future context.

Systems are ranked by the unweighted mean of the dense ranks of four metrics: token error rate (TER), enrolment-to-output speaker-embedding cosine similarity (SIM), DNSMOS P.835 overall quality (OVRL) [6], and a voice-activity timing ratio F1 (RATIO-F1).

Our system is a band-split RNN (BSRNN) separator conditioned on a fine-tuned ECAPA-TDNN speaker encoder. Our contributions are three-fold: (i) pre-training on simulated mixtures and speaker-verification data; (ii) multi-stage, multi-objective fine-tuning on real recorded meetings using quality-filtered pseudo-targets distilled from a non-causal teacher; and (iii) verification that the resulting causal system streams within roughly 22.9 ms of future dependency.

Participant name: **CARTSE**; team name: **CARTSE** (as registered on the leaderboard).

II. System Architecture

The system comprises two modules: a separator that extracts the target speech, and a speaker encoder that turns the enrolment e into a conditioning embedding. Both follow the official challenge baseline, a band-split RNN (BSRNN) [2] separator with target-speaker conditioning and an ECAPA-TDNN [3] speaker encoder, initialized from the released baseline weights and adapted by fine-tuning.

Separator: The mixture x is transformed by a short-time Fourier transform (STFT) with window length of 512 and hop of 128 into a real/imaginary spectrogram of shape $(2, F, T)$, where $F{=}257$ is the number of frequency bins and T the number of time frames. A time-frequency enrolment map [31] is concatenated to it, and the result is band-split into 32 sub-bands (100/200/500/2000-Hz widths) with sub-band normalization, giving a feature tensor $(32, 128, T)$. The frame-level speaker embedding then modulates these features. The separator stacks 6 repeats of a band-RNN followed by a temporal RNN, emits a complex band mask, and inverts it by an inverse STFT (iSTFT) to produce the target speech estimate $\hat{s}$. The separator is causal by using *unidirectional* for the temporal RNN and cumulative-layer normalization (`cLN`). The band-communication RNN stays bidirectional, as it operates across frequency within a frame, not across time. The network architecture is shown in Fig. 1. The recurrent core consumes no future (look-ahead) frames, so the algorithmic latency reduces to the STFT window − hop offset (24 ms), with a buffering latency of one hop (8 ms); an empirical perturbation test confirms an effective future dependency of mean 22.9 ms (Section V).

Speaker encoder: The conditioning embedding is produced by the baseline's ECAPA-TDNN [3] speaker encoder, which is further fine-tuned on English and Chinese datasets then adapted to the real recordings. Because the encoder runs on the enrolment, which is available in advance, it may remain non-causal without affecting streaming latency and causality is required only of the separator.

Conditioning: The enrolment conditions the separator through two paths. At the input, the time-frequency enrolment map (TF-Map) is an enrolment-magnitude $\times$ mixture-magnitude attention map concatenated to the spectrogram, making the input shape of $(3, F, T)$. Within the separator, the frame-level ECAPA embedding is linearly projected from 512 to the feature dimension 128 and applied by element-wise multiplication.

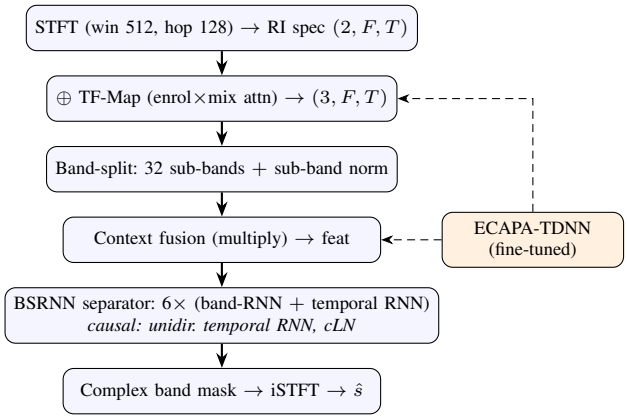

Fig. 1. CARTSE separator. Solid arrows are the signal path; dashed arrows are the two conditioning injections.

III. Data

Our data pipeline combines large-scale simulation with real meeting recordings. The separator is first pre-trained on *simulated* mixtures generated on-the-fly from clean speech, then fine-tuned on *real* far-field mixtures. Because these real mixtures have no clean per-speaker reference, their targets are *generated as pseudo-targets* by a teacher model and then quality-filtered. The speaker-conditioning encoder is trained separately on verification corpora and domain-adapted to the challenge recordings. This section describes each of these ingredients in turn: the simulated pre-training corpora, the speaker-encoder data, the real fine-tuning corpora, and the shared data augmentations including noise, reverberation, and channel-gap.

A. Simulated pre-training corpora

The base separator is pre-trained with on-the-fly mixing of clean English speech: LibriSpeech train-100 [11] with the urgent2025 speech set (EARS [14], VCTK [13], WSJ [15], LibriVox/DNS5 [16], LibriTTS [12]), containing containing 4,240 speakers / 498,665 utterances. Mixtures draw 1/2/3 speakers of $[0.1, 0.6, 0.3]$ at $[-5, 5]$ dB, with a reverberant target and added noise, where room impulse responses (RIRs) are simulated by setting RT_{60} at 0.05–0.3 s and signal-to-noise (SNR) is set at $[5, 15]$ dB. The signal lengths are set as 4 s for mixture and 3 s for enrolment with reverb/noise augmentation (0.5/0.5).

A fraction of mixtures are made *target-absent* so the separator learns to distinguish target speaker in mixtures and output silence if target speaker is absent, via two independently-drawn mechanisms: *distractor enrolment* (prob 0.35, a random absent speaker) and *noise-only* (prob 0.05, pure noise), making about 38.25% of target-absent in total. Validation uses a static held-out simulated set.

B. Speaker-encoder data

The conditioning encoder is a bilingual ECAPA-TDNN [3] trained for verification on VoxCeleb2-dev [18] (EN) and CN-Celeb1/2 train [19] (ZH), which consist of 8,987 speakers and 1,743,328 utterances in total, with RIR and 0.9/1.0/1.1 speed-perturbation augmentation ($\times 3 \rightarrow 26{,}961$ classes, prob 0.6). A domain fine-tune are then conducted on real recordings using enrolment clips and target-solo no-overlap segments from the far-field sessions, mixed 1:1 with the verification data to avoid forgetting. To generate speaker embedding robut against device properties, we introduce equalizer (EQ) filtering as another data augmentation.

C. Real fine-tuning corpora

Following REAL-T [10], we draw target-speaker training data from real conversational meeting corpora (AISHELL-4 [26], AliMeeting [27], AMI [28], CHiME-6 [29]). DipCo [30] is excluded since there is no official train split. These real mixtures have no clean per-speaker reference, so targets are *generated as pseudo-targets* by a teacher TSE model and quality-filtered. For every corpus we use only its official *train* split. The dev and eval splits of each corpus are never used for training. Each training example takes the mixture and the original enrolment as input and the generated target as the signal to reconstruct; roughly 3.6 enrolments per (mixture, speaker) are sampled at random. A target is kept only if it satisfies all four criteria:

- ΔOVRL > 0: a positive OVRL gain over the mixture;
- OVRL ≥ 2.2: the absolute OVRL clears a quality floor;
- prec ≥ 0.80: precision from FireRedVAD timing;
- TER < 0.6: token error rate from the official zipformer ASR.

Our system is fine-tuned on a real pseudo-target corpus (7,710 clips, 37.9 h) whose targets are generated by our non-causal Track 2 (offline) system acting as the teacher.

D. Noise, RIR, and channel-gap augmentation

Three augmentation families broaden the training conditions. Mixture noise is drawn from a 180,264-clip pool (urgent2025 DNS5+FSD50k [23]+FMA [24], WHAM! [21], MUSAN [20]); since the real mixtures are all meeting-room scenes, an 8-environment DEMAND [25] second source (vehicle/home/café) is added as weighted auxiliary noise (prob 0.3) for out-of-domain generalization. Reverberation uses a 40,248-wav RIR pool (SLR28 [22]+RWCP+AIR). Channel-gap augmentation models the enrolment–mixture device mismatch: a *light* form adds reverb/noise to the enrolment (each prob 0.2) in real-data fine-tuning, while a smooth, RMS-preserving EQ filter (Appendix A) is used only in the speaker-encoder fine-tune. On real data the mixture is genuine audio, so no synthetic RIR or noise-only mixtures are added and mixture noise is reduced to prob 0.4 (SNR $[5, 20]$).

IV. Training

Throughout, x is the mixture, $\hat{s}$ the output, s the target, e the enrolment, y the per-frame target-activity label, $s_t = (\langle \hat{s}, s \rangle / \|s\|^2)\, s$ the scaled target projection, and $\tilde{s}$ the target s with its non-speech frames (those with $y{=}0$) set to zero.

A. Simulated pre-training

The base separator is first pre-trained on simulated mixtures with the **zero-target masked SI-SDR loss**, split by target presence with floor τ=10^{-3}:

$$L_{\text{pres}} = -10\log_{10}\frac{\|s_t\|^2}{\|\hat{s}-s_t\|^2+\tau\|s\|^2}, \tag{1}$$

$$L_{\text{abs}} = \eta\cdot 10\log_{10}\big(\|\hat{s}\|^2+\tau\|x\|^2\big), \quad \eta=2.0. \tag{2}$$

Target-present chunks maximise SI-SDR (L_{pres}), while target-absent chunks are driven to silence (L_{abs}). The online separator is initialised from the official causal baseline and pre-trained on 4 s chunks with batch size 12 for 200 epochs at a learning rate of 1e-4.

B. Fine-tuning

The pre-trained separator is then fine-tuned on the real, pseudo-labeled data. All real-data fine-tuning adds mixture noise (probability 0.4, SNR $[5, 20]$), the DEMAND auxiliary source (0.3), and light channel-gap (0.2), uses uniform language batching, and is optimised with Adam (gradient clipping 5.0); the fine-tuned ECAPA encoder is grafted into the simulated-base checkpoint beforehand. The main objective is the **scenario-aware loss**, evaluated over target-active (TA) and target-silent (TS) intervals:

$$\begin{aligned} L_{\text{TA}} &= -\alpha_1\,\text{SI-SDR}_{\text{TA}} + w_{\text{m}}\,\text{MRSTFT}(\hat{s}\odot m_{\text{TA}}, s\odot m_{\text{TA}}),\\ L_{\text{TS}} &= \alpha_2\,10\log_{10}(\|\hat{s}\|^2_{\text{TS}}+\epsilon), \end{aligned} \tag{3}$$

where m_{TA} is the binary target-active mask (1 on TA frames, 0 elsewhere) that restricts the magnitude term to the active regions, α_1=0.5, α_2=0.01, w_{m}=1.0, and MRSTFT a multi-resolution STFT magnitude loss (fft/win $[512, 1024, 2048]$, hop $[128, 256, 512]$). L_{TA} is applied on target-active frames and L_{TS} on target-silent frames, the two sets being determined by the per-frame activity labels y. Weighted auxiliary losses are summed on top of this objective. All auxiliary networks are frozen (no gradient update) and, unless stated otherwise, operate on the waveform; w denotes the loss weight.

speaker consistency loss: Cosine distance between WeSpeaker ResNet34 embeddings of the output and the enrolment,

$$L = w\big(1-\cos(\text{WeSpk}(\hat{s}),\text{WeSpk}(e))\big), \tag{4}$$

where WeSpk is a frozen WeSpeaker ResNet34 model [4], the EN or ZH model selected by the utterance language. The enrolment embedding WeSpk(e) is detached, and the optional interferer-repulsion term is disabled.

mel-filterbank loss: L_1 distance between per-utterance CMVN-normalized 80-dimensional Kaldi log-mel features (25 ms window, 10 ms hop) of the output and of $\tilde{s}$,

$$L = w\,L_1\big(\text{CMVN}(\text{fbank}(\hat{s})),\ \text{CMVN}(\text{fbank}(\tilde{s}))\big), \tag{5}$$

where $\tilde{s}$ denotes the target s with its non-speech frames (those with y=0) set to zero; the same $\tilde{s}$ is used by the losses below that compare against the target.

DNSMOS loss: Direct maximization of the P.835 OVRL predicted by a frozen DNSMOS CNN [6]. With r the raw OVRL head and MOS $= ar^2+br+c$, $(a,b,c) = (-0.0677, 1.1155, 0.0460)$,

$$L = w\,(5-\text{MOS}(\hat{s})). \tag{6}$$

For our system, real-data fine-tuning runs in two stages on the pseudo-target corpus (Section III-C), both using 3 s chunks, batch size 12, and a learning rate decayed from from 1e-4 to 1e-5. Stage 1 (80 epochs, from the simulated base with the fine-tuned ECAPA grafted in) optimises the scenario-aware loss with weighted auxiliaries speaker-consistency 3.0, mel-filterbank 3.0, and DNSMOS 1.0. Stage 2 (40 epochs, initialised from the stage-1 epoch-60 checkpoint) keeps those terms and adds the **multi-layer zipformer loss** (overall weight 8.0), a cosine feature match at several frozen-zipformer encoder layers ℓ [7]:

$$L = w\sum_{\ell} w_\ell\big(1-\cos(f_\ell(\hat{s}), f_\ell(s))\big), \tag{7}$$

with per-layer weights {final:1.0, enc$_3$:0.4, enc$_4$:0.2} and the target detached. The submitted model is the stage-2 epoch-40 checkpoint.

TABLE I

ONLINE RESULTS. DEV IS HELD-OUT VALIDATION; EVAL IS THE OFFICIAL LEADERBOARD. `TFMAP_CAUSAL` AND `EMB_CAUSAL` ARE THE OFFICIAL CAUSAL BASELINES (THE FORMER IS OUR INITIALIZATION); SIM IS THE ENROLMENT-TO-OUTPUT SIMILARITY.

Set	System	TER↓	SIM↑	OVRL↑	F1↑
DEV	`TFMAP_CAUSAL`	0.652	0.535	1.560	0.844
	`EMB_CAUSAL`	0.705	0.492	1.630	0.829
	CARTSE	0.514	0.565	3.396	0.864
EVAL	`TFMAP_CAUSAL`	0.805	0.456	1.670	0.836
	`EMB_CAUSAL`	0.809	0.422	1.714	0.821
	CARTSE	0.699	0.504	3.436	0.848

V. RESULTS AND VALIDATION

Table I reports our DEV and official EVAL results against the two official causal baselines: `BSRNN_TFMAP_CAUSAL`, from which our separator is initialized, and `BSRNN_EMB_CAUSAL`. On both splits our system improves over both baselines on all four metrics, most markedly in perceptual quality (EVAL OVRL $1.67 \rightarrow 3.44$) and intelligibility (EVAL TER $0.81 \rightarrow 0.70$); the streaming latency budget is analysed below.

Streaming latency. We report latency following the challenge's ICASSP-2023-DNS conventions. Our STFT front-end uses a 32 ms window and an 128-sample hop (8 ms) at 16000 Hz, and the causal separator adds no future (look-ahead) frames, so the *algorithmic latency* is window − hop = 24 ms and the *buffering latency* is one hop, 8 ms; per-frame compute is constant, so the separator runs frame-synchronously. Mirroring the challenge's perturbation-based evaluation of effective system latency, we perturb the input at time indices *ahead* of a given output frame and take the effective future dependency as the largest look-ahead at which the relative change exceeds a 5×10^{-4} threshold. Sweeping

over frames and frequency bands yields $\approx$ 22.2–23.7 ms (mean 22.9 ms), just below the 24 ms algorithmic latency and well within the challenge's 100 ms effective-latency bound.

VI. Conclusion

The CARTSE system is a causal band-split RNN separator, verified to stream within a future dependency of ≈ 22.9 ms. It is pre-trained on simulated mixtures and fine-tuned on quality-filtered pseudo-targets from a non-causal teacher. All evaluation models used as training objectives were applied only to eligible training data; the evaluation set was processed by a single fixed system version, without per-utterance metric-driven optimization.

Appendix A
Channel-Gap Augmentation

In the real recordings the enrolment and mixture often come from different devices — different frequency responses, bandwidths, and recording chains. On DEV the learned mixture–enrolment similarity is the strongest WER predictor (Spearman $\rho -0.52$) and the out-of-domain evaluation condition shows the largest enrolment–mixture gap, yet synthetic training draws both from the same chain, so the conditioning path is never trained to be device-invariant.

The augmentation passes each training enrolment through a random simulated device response (enrolment channel $\neq$ mixture channel on almost every sample): a smooth, RMS-preserving dB gain curve applied by rFFT multiply, combining a spectral tilt (± 4 dB/oct), a cubic-spline random EQ (6 log-spaced nodes in $[80, 7600]$ Hz, ± 6 dB), and Butterworth band limits (low-pass $U(3.4$–7.5 kHz) p0.4, high-pass $U(50$–150 Hz) p0.5), in source–room–device order. TSE fine-tuning uses this light enrolment-side setting (reverb/noise-enroll 0.2); the speaker-encoder fine-tune uses an "extreme" preset (prob 0.7, tilt ± 10 dB/oct, 8-node ± 12 dB EQ, wider band limits).

Appendix B
Data Sources

Per challenge rules, no official training set is defined; participants may use any open-source data and pre-trained models with documented sources. Speech corpora: LibriSpeech [11], LibriTTS [12], VCTK [13], EARS [14], WSJ [15], DNS5/LibriVox [16], VoxCeleb1/2 [17], [18], CN-Celeb1/2 [19]. Noise/RIR: MUSAN [20], WHAM! [21], RIRS_NOISES/SLR28 [22], FSD50k [23], FMA [24], DEMAND [25]. Challenge corpora (training splits only): AISHELL-4 [26], AliMeeting [27], AMI [28], CHiME-6 [29]. Pre-trained models: ECAPA-TDNN and WeSpeaker ResNet34 [3], [4], FireRedVAD, DNSMOS P.835 [6], and zipformer-zh/en [7].

References

[1] K. Žmolíková *et al.*, "SpeakerBeam: Speaker aware neural network for target speaker extraction in speech mixtures," IEEE J. Sel. Topics Signal Process., vol. 13, no. 4, pp. 800–814, 2019.

[2] Y. Luo and J. Yu, "Music source separation with band-split RNN," IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 31, pp. 1893–1901, 2023.

[3] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in Proc. Interspeech, 2020, pp. 3830–3834.

[4] H. Wang *et al.*, "WeSpeaker: A research and production oriented speaker embedding learning toolkit," in Proc. ICASSP, 2023.

[5] A. Radford *et al.*, "Robust speech recognition via large-scale weak supervision," in Proc. ICML, 2023, pp. 28492–28518.

[6] C. K. A. Reddy, V. Gopal, and R. Cutler, "DNSMOS P.835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in Proc. ICASSP, 2022, pp. 886–890.

[7] Z. Yao *et al.*, "Zipformer: A faster and better encoder for automatic speech recognition," in Proc. ICLR, 2024.

[8] M. Wortsman *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in Proc. ICML, 2022, pp. 23965–23998.

[9] C. Boeddeker *et al.*, "Front-end processing for the CHiME-5 dinner party scenario," in Proc. CHiME Workshop, 2018.

[10] S. Li, S. Wang, J. Han, K. Zhang, W. Wang, and H. Li, "REAL-T: Real conversational mixtures for target speaker extraction," in Proc. Interspeech, 2025, pp. 1923–1927.

[11] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in Proc. ICASSP, 2015, pp. 5206–5210.

[12] H. Zen *et al.*, "LibriTTS: A corpus derived from LibriSpeech for text-to-speech," in Proc. Interspeech, 2019, pp. 1526–1530.

[13] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," Univ. Edinburgh, 2019.

[14] J. Richter *et al.*, "EARS: An anechoic fullband speech dataset benchmarked for speech enhancement and dereverberation," in Proc. ICASSP, 2024.

[15] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in Proc. Workshop on Speech and Natural Language, 1992, pp. 357–362.

[16] H. Dubey *et al.*, "ICASSP 2023 deep noise suppression challenge," in Proc. ICASSP, 2023.

[17] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in Proc. Interspeech, 2017, pp. 2616–2620.

[18] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in Proc. Interspeech, 2018, pp. 1086–1090.

[19] Y. Fan *et al.*, "CN-Celeb: A challenging Chinese speaker recognition dataset," in Proc. ICASSP, 2020, pp. 7604–7608.

[20] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," arXiv:1510.08484, 2015.

[21] G. Wichern *et al.*, "WHAM!: Extending speech separation to noisy environments," in Proc. Interspeech, 2019, pp. 1368–1372.

[22] T. Ko *et al.*, "A study on data augmentation of reverberant speech for robust speech recognition," in Proc. ICASSP, 2017, pp. 5220–5224.

[23] E. Fonseca *et al.*, "FSD50K: An open dataset of human-labeled sound events," IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 30, pp. 829–852, 2022.

[24] M. Defferrard *et al.*, "FMA: A dataset for music analysis," in Proc. ISMIR, 2017, pp. 316–323.

[25] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multi-channel acoustic noise database (DEMAND)," in Proc. Meetings on Acoustics, 2013.

[26] Y. Fu *et al.*, "AISHELL-4: An open source dataset for speech enhancement, separation, recognition and speaker diarization in conference scenario," in Proc. Interspeech, 2021, pp. 3665–3669.

[27] F. Yu *et al.*, "M2MeT: The ICASSP 2022 multi-channel multi-party meeting transcription challenge," in Proc. ICASSP, 2022, pp. 6167–6171.

[28] J. Carletta *et al.*, "The AMI meeting corpus: A pre-announcement," in Proc. MLMI, 2006, pp. 28–39.

[29] S. Watanabe *et al.*, "CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," in Proc. CHiME Workshop, 2020.

[30] M. Van Segbroeck *et al.*, "DiPCo — dinner party corpus," arXiv:1909.13447, 2019.

[31] K. Zhang *et al.*, " Multi-level speaker representation for target speaker extraction," in Proc. ICASSP, 2025, pp. 1–5.