PrefixCue-TSE: NCNU System Description for the REAL-TSE Challenge Track 1

NCNU
Participant name: NCNU
Team name: NCNU
REAL-TSE Challenge Track 1: Online Target Speaker Extraction

***Abstract*—This report describes the NCNU submission to the REAL-TSE Challenge Track 1, the online target speaker extraction task. Our submitted system, named PrefixCue-TSE, is a compact causal target-speaker extraction model built on a lightweight recurrent convolutional separator. The system uses enrollment-derived speaker-prefix tokens, prefix-aware causal attention, speaker-conditioned decoder skip fusion, and prompt-conditioned multi-mask estimation. The submitted waveform package was generated by a fixed averaged checkpoint, without per-utterance metric-driven optimization or evaluation-set candidate selection. The reported end-to-end latency budget is 56 ms, and the official latency-verification style script measured a maximum future dependency of 2.6 ms for the submitted system.**

***Index Terms*—target speaker extraction, online speech enhancement, recurrent convolutional networks, speaker conditioning, REAL-TSE Challenge**

I. System Summary

The submitted method is PrefixCue-TSE, an online target speaker extraction system for the REAL-TSE Challenge [1], [2]. This report describes the final NCNU system version submitted to the leaderboard. The submitted system uses a fixed averaged PrefixCue-TSE checkpoint selected on the official REAL-TSE development set. All EVAL1 and EVAL2 utterances were processed by this fixed system version.

II. Model Architecture

A. End-to-End Processing Flow

The complete extraction flow is as follows. First, the enrollment waveform is encoded and summarized into a short sequence of speaker-prefix tokens. Second, the mixture waveform is transformed into STFT-domain features and passed through a GTCRN-style recurrent convolutional encoder [3]. Third, the separator concatenates enrollment-prefix tokens, a separator token, and mixture-frame features along the time axis. Fourth, prefix-aware causal recurrent-attention blocks process this sequence while preventing mixture frames from attending to future mixture frames. Fifth, the decoder reconstructs full-band features with speaker-conditioned skip fusion. Finally, a prompt-conditioned latent-mask head estimates target-speaker masks, applies noise-sink suppression, reconstructs the enhanced complex spectrum, and returns the waveform through inverse STFT.

In compact form, the submitted model computes:

$$\hat{x}_t = \mathrm{iSTFT}\left(M_\theta(X_{1:t}, P(e)) \odot X_{1:t}\right), \tag{1}$$

where $X_{1:t}$ denotes the causal mixture STFT history up to time frame t, e is the enrollment utterance, $P(e)$ is the enrollment-derived speaker-prefix representation, and M_θ is the speaker-conditioned multi-mask estimator.

B. Input Features

All audio is processed at 16 kHz. The model uses an STFT representation with

$$n_{\mathrm{fft}} = 512, \quad \mathrm{win} = 512, \quad \mathrm{hop} = 128. \tag{2}$$

The mixture waveform is converted to complex STFT features. Magnitude, real, and imaginary components are used, followed by ERB-style frequency compression and a lightweight sub-band feature extraction front-end.

C. Enrollment-Derived Speaker Prefix

The model does not use an external speaker embedding network during inference. Instead, the enrollment utterance is encoded by the same neural front-end family and summarized into speaker-prefix tokens. The submitted configuration uses:

$$N_{\mathrm{prefix}} = 8, \quad d_{\mathrm{att}} = 32, \quad L_{\mathrm{enroll}} = 48000. \tag{3}$$

The internal separator sequence places the speaker-prefix tokens before a separator token and the subsequent mixture-frame features. This allows the mixture-side separator to access target-speaker information without relying on a separately trained speaker verification model. Because the enrollment utterance is available before mixture processing, it is treated as static side information; the online causality constraint is applied to the mixture stream.

D. Prefix-Aware Causal Separator

The separator contains six DPGRNN/GTCRN-style recurrent convolutional blocks with 80 channels. The main speaker-conditioning mechanism is prefix-aware causal attention. Mixture frames can attend to all enrollment-prefix tokens and only past or current mixture frames. Speaker-prefix tokens are not allowed to consume future mixture frames. In the attention path, this constraint is implemented with a lower-triangular causal mask over the mixture-frame portion of the sequence, while the enrollment-prefix tokens are treated as fixed side information. In the DPGRNN path, bidirectional recurrence is used only along the frequency/sub-band axis within each frame, while temporal recurrence is unidirectional

TABLE I
MAIN ARCHITECTURE CONFIGURATION OF PREFIXCUE-TSE.

Component	Mechanism
Sample rate	16 kHz
STFT	$n_{\text{fft}} = 512$, win=512, hop=128
Separator depth	6 DPGRNN/GTCRN blocks
Hidden channels	80
Frequency compression	ERB-style compression
Enrollment tokens	8 prefix tokens
Enrollment attention dim.	32
Speaker conditioning	Prefix tokens + decoder skip fusion
Attention mode	Prefix-aware causal local attention
Latent masks	4
Noise sink	Explicit non-target branch
Mask prompt conditioning	Speaker-prompt modulation
DSI prompt conditioning	Dynamic speaker-information fusion
Prompt routing	Enrollment-conditioned group-bias routing
Recall activity conditioning	Target-activity recall cue

over mixture frames. This design preserves causal mixture-side processing while making target-speaker information available throughout the separation path. The submitted system uses local prefix-aware causal attention rather than full-sequence non-causal attention.

E. Decoder and Multi-Mask Head

The decoder uses speaker-conditioned skip fusion. The prompt-conditioned mask-estimation head predicts four latent target-mask candidates and includes a noise-sink branch. Mask-prompt conditioning, dynamic-speaker-information prompt conditioning, prompt routing, and recall-activity conditioning are enabled. The noise-sink branch is used to reduce non-target leakage and background interference. The final enhanced complex spectrum is reconstructed and transformed back to waveform by inverse STFT.

F. Architecture Configuration

Table I summarizes the main submitted architecture configuration.

G. Speaker-Conditioning Components

The system contains four speaker-conditioning paths. The first path is the prefix-token path, where enrollment-derived tokens are prepended to the separator sequence. The second path is prefix-aware attention, where mixture frames can use the speaker-prefix tokens at every time step while remaining causal over mixture frames. The third path is speaker-conditioned decoder skip fusion, which injects enrollment information into decoder-side reconstruction. The fourth path is prompt conditioning in the mask head, where enrollment-derived prompts modulate latent-mask aggregation and routing. Together, these paths form the "PrefixCue" mechanism used by the submitted system.

H. Mask Estimation and Noise Sink

Instead of estimating a single target mask directly, the output head estimates four latent mask candidates. The prompt-routing module combines these latent masks according to enrollment-conditioned routing cues. The noise-sink branch provides an explicit non-target sink, which helps reduce interference leakage when the target speaker is inactive or weak. The predicted mask is constrained and applied to the mixture complex spectrum before inverse STFT reconstruction.

III. TRAINING DATA AND OBJECTIVE

A. Training Data

The model was trained only on the open-source Libri2Mix 16 kHz min training split, specifically the train-100 subset [5]. The Libri2Mix 16 kHz min development split was used as the neural-network validation split during training. We used the clean-mixture condition, 3.0-second mixture segments, and 3.0-second enrollment segments. No external pretrained model was used in the submitted extraction model. No REAL-TSE EVAL1 or EVAL2 data was used for training, fine-tuning, hyper-parameter tuning, data augmentation, or any form of model optimization. We also did not use the development or test splits of AliMeeting, AISHELL-4, AMI, DipCo, or CHiME6 for training or augmentation, following the data restrictions on the challenge website [1].

B. Optimization

The optimizer used an initial learning rate of 10^{-3} with exponential decay to 2.5×10^{-5}. The weight decay was 10^{-4}. Models were trained for up to 150 epochs with checkpoint saving at every epoch and gradient clipping at 5.0.

C. Loss Function

The training objective is a ratio-proxy target-speaker extraction loss. It combines SI-SNR/SI-SDR-oriented reconstruction with multi-resolution STFT regularization and target-activity terms. For an estimated waveform $\hat{s}$, target waveform s, mixture y, enrollment e, and auxiliary activity/mask outputs, the loss can be summarized as:

$$\begin{aligned}\mathcal{L} = &- \text{SI-SNR}(\hat{s}, s) + \lambda_{\text{stft}}\mathcal{L}_{\text{MR-STFT}}(\hat{s}, s) \\ &+ \lambda_{\text{active}}\mathcal{L}_{\text{active-L1}} + \lambda_{\text{under}}\mathcal{L}_{\text{under-energy}} \\ &+ \lambda_{\text{leak}}\mathcal{L}_{\text{silence-leak}} + \lambda_{\text{act-bce}}\mathcal{L}_{\text{activity-BCE}} \\ &+ \lambda_{\text{enroll}}\mathcal{L}_{\text{enroll-sim}}. \end{aligned} \tag{4}$$

The submitted weights were $\lambda_{\text{stft}} = 0.001$, $\lambda_{\text{active}} = 0.003$, $\lambda_{\text{under}} = 0.015$, $\lambda_{\text{leak}} = 0.0002$, $\lambda_{\text{act-bce}} = 0.002$, and $\lambda_{\text{enroll}} = 0.003$. The auxiliary terms encourage target-speaker preservation, reduce target under-extraction, and suppress leakage in target-inactive regions.

IV. VALIDATION AND CHECKPOINT SELECTION

The official REAL-TSE development set was used only for validation, model comparison, and system-level checkpoint selection. It was not used for training or fine-tuning. Candidate single checkpoints and averaged checkpoints were evaluated as complete fixed systems on the development set. We compared candidates using the four metrics provided by the official development-set evaluation toolkit: TER, Timing F1, speaker similarity, and DNSMOS OVRL. Following the

common checkpoint-averaging practice used in the official WeSep-based baseline recipes [4], we averaged selected training checkpoints after training and treated each averaged model as a single fixed system. The final submitted system used the averaged checkpoint that provided the best development-set tradeoff for TER, speaker similarity, and Timing F1 among the final candidates.

Leaderboard feedback, when available, was used only to decide whether to submit a new fixed system version, as permitted by the challenge policy. We did not use evaluation-set utterance-level scores or metric feedback for checkpoint selection, waveform refinement, post-processing, or per-utterance candidate selection. The selected checkpoint was fixed before producing the submitted EVAL1 and EVAL2 waveforms.

V. Online Latency

Track 1 requires low-latency target speaker extraction with end-to-end algorithmic latency not exceeding 100 ms [1]. For the submitted online setting, we report a conservative end-to-end latency budget of:

$$56\,ms. \tag{5}$$

This number is the nominal latency budget of the streaming system, not the wall-clock runtime. It includes the signal-analysis delay caused by the STFT/iSTFT framing and the input buffering allowance used by the online interface.

The submitted model uses a 512-sample analysis window and a 128-sample hop at 16 kHz. Following the latency definition used in the challenge rules, the STFT-based algorithmic delay is the window length minus the hop length:

$$L_{\mathrm{STFT}} = (512 - 128)/16000 = 24\,ms. \tag{6}$$

This accounts for the fact that an output frame cannot be finalized until the corresponding analysis window has been observed.

In addition, the submitted online setting reserves a four-hop chunk-level buffering allowance. Since one hop is $128/16000 = 8\,ms$, the buffering budget is:

$$L_{\mathrm{buffer}} = 4 \times 128/16000 = 32\,ms. \tag{7}$$

Therefore, the reported end-to-end latency budget is:

$$L_{\mathrm{total}} = L_{\mathrm{STFT}} + L_{\mathrm{buffer}} = 24\,ms + 32\,ms = 56\,ms. \tag{8}$$

The neural separator itself processes the mixture stream causally and advances its streaming state hop by hop; the four-hop term is the conservative buffering budget reported for the submitted online system.

The official latency-verification style script provides a perturbation-response estimate of future dependency; it is a consistency check and is not meant to replace the nominal latency budget reported above. Using the submitted system's streaming wrapper, the measured future dependency was:

minimum: -0.3 ms, mean: 1.1 ms, maximum: 2.6 ms.

The small negative value is an artifact of hop/frame quantization in the perturbation-response measurement. The maximum measured future dependency is well below the reported 56 ms latency budget. This measured value is not added to the nominal latency; instead, it confirms that perturbations in the future do not affect the output earlier than allowed by the stated STFT framing and buffering budget.

VI. Inference Protocol and Rule Compliance

For the final submitted system, all EVAL1 and EVAL2 utterances were processed by a single fixed averaged checkpoint. No ASR model, speaker verification model, DNSMOS model, or official evaluation model was used to modify the generated waveforms. We did not apply metric-targeted waveform refinement, adversarial perturbation, post-hoc enhancement, or per-utterance candidate selection. The output package contained both EVAL1 and EVAL2 results under the run name NCNU, and the submitted audio files were exported as WAV, PCM_16, mono, 16 kHz.

We followed the challenge data and submission rules summarized on the official website and challenge report [1], [2]. In particular, REAL-TSE DEV was used only for validation and system-level checkpoint selection, while EVAL1 and EVAL2 were reserved for final waveform generation. We did not use evaluation-set utterance scores, gradients, embeddings, or metric feedback to optimize, tune, select among candidates, or post-process individual evaluation utterances. The reported online latency budget is 56 ms, below the 100 ms Track 1 constraint.

References

[1] REAL-TSE Challenge, "REAL-TSE Challenge: Real-world Target Speaker Extraction Challenge," 2026. [Online]. Available: https://real-tse.github.io/challenge/

[2] S. Wang et al., "REAL-TSE Challenge: Real-world Target Speaker Extraction from Conversational Recordings," 2026. [Online]. Available: https://real-tse.github.io/assets/pdf/REAL-TSE-Challenge-Report.pdf

[3] X. Rong, T. Sun, X. Zhang, Y. Hu, C. Zhu, and J. Lu, "GTCRN: A speech enhancement model requiring ultralow computational resources," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 971–975, doi: 10.1109/ICASSP48485.2024.10448310.

[4] REAL-TSE, "wesep-real-tse: WeSep baseline recipes and inference toolkit for REAL-TSE," 2026. [Online]. Available: https://github.com/REAL-TSE/wesep-real-tse

[5] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "LibriMix: An open-source dataset for generalizable speech separation," arXiv preprint arXiv:2005.11262, 2020.