System Description

Participant name: yijiahe; Team name: YiJiaHe

1. Overview

Our system makes two primary contributions: (1) we construct a ~71,771-sample training set from scratch by processing the raw recordings and diarization annotations of four public corpora (AISHELL-4, AliMeeting, AMI, CHiME6) into the REAL-T challenge format, including mixture audio windows, enrollment clips, ground-truth transcripts, and per-speaker VAD activity labels; (2) we fine-tune a BSRNN-based TSE model on this data using a differentiable multi-objective loss. The four complementary proxy objectives — ASR recognition quality, speaker identity similarity, temporal voice activity, and perceptual audio quality — jointly drive the TSE model update through backpropagation.

2. Model Architecture

2.1 TSE Model: BSRNN + ECAPA-TDNN

We download and start from the `bsrnn_ecapa_vox1` pre-trained checkpoint, which is a BSRNN-based TSE model trained on VoxCeleb1 as described in the REAL-T paper. This model combines Band-Split RNN (BSRNN) with an ECAPA-TDNN speaker encoder.

BSRNN separator:

- Feature type: consistent (complex STFT with 512-point window, 128-sample stride)
- Feature dimension: 128 (complex sub-band features)
- Number of BSRNN repeat blocks: 6
- Speaker embedding fusion: element-wise multiplication (`spk_fuse_type: multiply`)

ECAPA-TDNN speaker encoder:

- Architecture: `ECAPA_TDNN_GLOB_c512`
- Speaker embedding dimension: 192
- Acoustic features: 80-dimensional filterbank (ASTP pooling)
- Pre-trained on VoxCeleb1 (251 speakers)
- **Frozen during fine-tuning** (only the BSRNN separator is updated)

Model parameters: The total number of BSRNN separator parameters is approximately 20M. The ECAPA-TDNN speaker encoder (~6.5M parameters) is kept frozen throughout training.

3. Training Data

We construct the training set from scratch using the raw recordings and annotations of four public corpora (AISHELL-4, AliMeeting, AMI, CHiME6), reformatted to match the REAL-T challenge data schema. It should be noted that we only use the **training splits** of each corpus.

3.1 Data Construction Pipeline

For each corpus, we implement a two-pass construction procedure:

Pass 1 — Enrollment extraction: We scan all recording sessions for time intervals where exactly one speaker is active (solo segments). Solo segments are identified by parsing the original diarization annotations (RTTM for AISHELL-4, Praat TextGrid for AliMeeting, AMI segments XML for AMI, transcription JSON for CHiME6). Each qualifying solo segment (minimum 5 seconds, non-silent) is saved as a candidate enrollment audio clip. Up to 5 enrollment clips are retained per speaker per session.

Pass 2 — Mixture extraction: We detect regions containing speech overlap (≥2 speakers active simultaneously). Overlapping regions within 0.5 seconds of each other are merged. For each merged overlap region (minimum 5 seconds duration, maximum 100 seconds), we extract the corresponding audio window from the multi-channel recording mix, peak-normalize it to 0.95, and save it as the mixture audio. The ground-truth transcript for each target speaker is extracted from the original annotations (TextGrid text tiers for Chinese datasets; JSON transcription for English datasets) and cleaned of noise markers.

Target VAD label construction: We additionally extract per-session diarization annotations to build `target_activity_segments.jsonl` — a per-(mixture, speaker) file recording the time intervals when the target speaker is active within each mixture window. These labels are derived by intersecting the speaker's global activity segments with the mixture time window and converting to local relative timestamps.

3.2 Filtering Criteria

- **Speaker ratio > 20%:** Only samples where the target speaker accounts for more than 20% of the mixture duration are retained.
- **Transcript length ≥ 2 characters** (after noise marker removal): Samples with near-empty transcripts are discarded.
- **Mixture duration ≤ 30 seconds** at training time: Whisper processes at most 30 seconds of audio (480,000 samples @ 16kHz). Longer samples are filtered out.
- **Enrollment duration ≤ 10 seconds:** Enrollment audio is truncated to this limit to bound memory usage.

3.3 Dataset Statistics

Dataset	Language	Source Annotation	Valid Samples	Condition
AISHELL-4	Chinese	RTTM	~11,465	Meeting room, far-field microphone array
AliMeeting	Chinese	Praat TextGrid	~33,148	Meeting room, far-field microphone array
AMI	English	Segments XML	~14,933	Meeting room, lapel + array microphones
CHiME6	English	Transcription JSON	~12,225	Dinner party, far-field microphone array
Total			**~71,771**	

4. Training Objectives

We design a **multi-objective combined loss** that uses the evaluation metrics themselves as training signals:

$$\mathcal{L} = \lambda_{\text{ce}} \cdot \mathcal{L}_{\text{CE}} + \lambda_{\text{sim}} \cdot \mathcal{L}_{\text{SIM}} + \lambda_{\text{vad}} \cdot \mathcal{L}_{\text{VAD}} + \lambda_{\text{dnsmos}} \cdot \mathcal{L}_{\text{DNSMOS}}$$

with initial weights: **λ_ce = 1.0, λ_sim = 5.0, λ_vad = 0.5, λ_dnsmos = 0.2**. During the competition, we manually adjusted the weights several times based on the real-time rankings.

4.1 ASR Cross-Entropy Loss (L_CE)

We use a **frozen Whisper large-v3** model in teacher-forcing mode to compute cross-entropy loss between the recognized tokens and the ground-truth transcripts:

$$\mathcal{L}_{\text{CE}} = \text{CrossEntropy}(\text{logits}(f_{\text{Whisper}}(\hat{s})), y_{\text{text}})$$

where $\hat{s}$ is the TSE output waveform and y_{text} is the ground-truth transcript from the dataset metadata.

Differentiable Log-Mel Extraction: The standard HuggingFace `WhisperFeatureExtractor` uses NumPy operations that break the gradient path. We re-implement the complete Whisper log-mel pipeline in pure PyTorch:

- STFT with Hann window (N_FFT=400, hop=160, 16kHz → 128 mel bins)
- Mel filterbank matrix multiplication (`torchaudio.functional.melscale_fbanks`)
- Log compression: `log10(max(mel, 1e-10))`, normalized by `(log10_mel + 4) / 4`

This ensures gradients flow from the CE loss all the way back through the mel spectrogram to the TSE output waveform.

4.2 Speaker Similarity Ranking Loss (L_SIM)

To preserve speaker identity in the extracted speech, we apply a **hinge ranking loss** that enforces:

$$\text{sim}(\hat{s}, e) > \text{sim}(m, e) + \text{margin}$$

where $\hat{s}$ is the TSE output, e is the enrollment audio, m is the mixture audio, and initial margin = 0.3. Formally:

$$\mathcal{L}_{\text{SIM}} = \mathbb{E}\left[\max(0,\ \text{margin} - (\cos(\hat{s}, e) - \cos(m, e)))\right]$$

We use **dual frozen ResNet34 speaker encoders** aligned with the challenge evaluation:

- English (CHiME6, AMI): `voxceleb_resnet34_LM` (WeSpeaker)
- Chinese (AliMeeting, AISHELL-4): `cnceleb_resnet34_LM` (WeSpeaker)

Gradients flow through the TSE output → kaldi fbank → frozen encoder → cosine similarity, updating only the TSE model parameters.

4.3 Target VAD Loss (L_VAD)

We leverage the per-dataset `target_activity_segments.jsonl` annotations to apply **frame-level binary cross-entropy supervision** on the TSE output energy:

$$\mathcal{L}_{\text{VAD}} = \text{BCE}(\sigma(\text{log\_energy}(\hat{s})),\ \mathbf{y}_{\text{VAD}})$$

Frame energy is computed by unfolding the waveform into 25ms frames with 10ms shift, computing RMS energy, and mapping through a calibrated sigmoid: $\sigma((\log_e(\mathrm{rms}) + 10)/5)$. This encourages near-silence during non-speech frames and high energy during speech frames.

4.4 Differentiable DNSMOS Loss (L_DNSMOS)

To directly optimize perceptual quality, we convert the Microsoft DNSMOS ONNX models (`sig_bak_ovr.onnx`) to PyTorch and incorporate them as a differentiable loss:

$$\mathcal{L}_{\mathrm{DNSMOS}} = -\mathrm{DNSMOS\_OVRL}(\hat{s})$$

The DNSMOS inference uses 9.01-second audio segments with 120-mel spectrogram features (n_fft=321, hop=160), matching the original ONNX model's preprocessing exactly.

We employ the AdamW optimizer with an initial learning rate of 1e-5, decayed exponentially to 1e-6, and a 1-epoch warmup. The leaderboard shows the result of the model with epoch=37.

No post-processing is applied. It should be noted that our TSE model output is directly submitted as the extracted target speaker audio. No post-processing is performed.