Technical Report for MERL's Real-TSE Challenge Submission

Dominik Klement*[1,2], Yoshiki Masuyama[1], Christoph Boeddeker[1],
Kohei Saijo[3], Julius Richter[1], Gordon Wichern[1], Jonathan Le Roux[1]
[1]Mitsubishi Electric Research Laboratories, Cambridge, USA,
[2]Speech@FIT, Brno University of Technology, Czechia, [3]Mitsubishi Electric Corporation, Kanagawa, Japan

I. Introduction

Target speaker extraction (TSE) aims to extract a speaker of interest from a speech mixture, potentially corrupted by noise or reverberation. Many existing approaches are trained and evaluated on simulated datasets, such as Libri2Mix [1], where they achieve improvements exceeding 20 dB SI-SDRi. However, these models often fail to generalize to real-world recordings, such as CHiME-6 [2] and DipCo [3], which exhibit substantially more challenging acoustic conditions. This performance degradation is primarily caused by the domain gap between simulated training data and real-world evaluation data.

To address this challenge, the authors of the Real-T dataset [4] introduced a challenging development and evaluation benchmark comprising both English and Chinese recordings. The long recordings are post-processed into conversational speech mixtures of up to one minute in duration with multiple speakers.

In this report, we present our submission to the RealTSE Challenge, which builds on the baseline model architecture. Rather than developing a more powerful model, we focus on data preprocessing and training with real far-field conversational mixtures. Our results demonstrate that strong performance can be achieved with a relatively simple model when paired with an effective data processing and training pipeline.

II. Model

We decided to use one of the provided baseline models Band-split RNN (BSRNN) with multi-level speaker conditioning [5] that utilizes per-frame and pooled speaker embeddings for speaker conditioning.

Compared to the baseline configuration, we replaced EcapaTDNN-512 with EcapaTDNN-1024 [6], which is a larger and more powerful speaker embedding model. We also increased the depth of the BSRNN model from 6 to 10 blocks. This modification significantly improved the extraction performance.

III. Data Processing

In this section, we describe our multi-stage data processing pipeline, consisting of cleaning single-speaker data for simulated pre-training and processing far-field and close-talk mixture pairs for real noisy data training.

A. Pre-training Speech Data Processing

We combine data from multiple open datasets containing vast amounts of speakers and speaking style variability: LibriSpeech [7], VoxCeleb2 [8], Emilia-YODAS English and Chinese [9], VCTK [10], and EARS [11].

First, we applied ClearerVoice [12] speech enhancement models based on masking MOSSFormer [13] to preprocess the noisy corpora such as VoxCeleb2 and Emilia-YODAS. Then, we computed speaker cosine similarity between the original and the enhanced recordings using a pre-trained ResNet34 model from WeSpeaker [14] to further filter out data with speaker similarity lower than 0.85. We observed that the outliers with speaker similarity lower than 0.5 contained multiple speakers or severe degradation that the enhancement could not clean up. Furthermore, we applied another filtering to only consider audio with DNSMOS above 3.0. Even though a high DNSMOS score does not guarantee clean speech, this filtering step ensured that the enhancement likely produced estimates accurate enough for training.

Afterwards, we computed word-level alignments using Montreal Forced Aligner (MFA) [15] to avoid sampling non-speech segments as target speaker enrollments. Although such segments could help train the model not to always expect the target speaker to be present in the mixture, they would make it difficult to precisely control their proportion during training. Instead, with a probability of 5%, we explicitly generated non-target samples by pairing the enrollment with a speaker absent from the input mixture and trained the model to produce a zero output.

B. Noises and Room Impulse Responses

We applied room impulse responses (RIRs) on a clean input mixture and enrollment independently with a probability of 0.8. The RIRs were randomly sampled from DNS4 [16] and from synthetic impulse responses simulated using Pyroomacoustics [17] with weights 0.8 and 0.2, respectively. We purposely simulated larger rooms to cover the scenarios of extreme echoes.

Furthermore, to increase the robustness against multiple noises, we used the following noise databases containing more stationary and also impulsive noises: CHiME-3 [18],

*Work done while Dominik Klement was an intern at MERL.

DEMAND [19], DNS4 [16], FMA [20], FSD50K [21], MUSAN [22], WHAM! [23], and simulated Wind noises from the URGENT challenge [24].

We used weighted sampling to increase the weight towards the stationary-like noises, where we set the probability of sampling CHiME3, WHAM!, and Wind noises to 0.15, and the probability of all the other noises to 0.11. We randomly sampled SNR from a $[-5, 15]$ dB range and added the noise to the input mixture and enrollment with probability 0.8. Mixture and enrollment noises were selected independently, covering all combinations of clean and noisy mixture and enrollment pairs during training.

C. Real Data Processing

To include real-world data for training, we utilized CHiME-6 [2], AMI [25], AISHELL-4 [26], and AISHELL-5 [27]. As AISHELL-5 provides close-talk recordings that are already synchronized with far-field microphones and do not contain significant cross-talk, we only perform speech enhancement using ClearerVoice. For AISHELL-4, we perform GSS [28] on multi-channel audio using ground-truth speaker diarization and apply speech enhancement afterwards. During training, we use both AISHELL datasets for simulated far-field mixtures only, as both contain a low amount of overlapped speech.

Single-talker Far-field Mixtures: To further adapt the model to real noisy data, we select single-talker segments of speech based on the provided dataset annotations. Then, we cut out the same segment from the speaker's close-talk microphone and average all the close-talk channels if multiple are available, and proceed with applying ClearerVoice speech enhancement to denoise and dereverberate the close talk.

To enable training on such mixtures, we perform max-peak cross-correlation synchronization with maximum lag set to one second. Afterwards, inspired by [29], we apply a causal Wiener filter with 10 ms window (160 taps) to project the close-talk signal to the corresponding far-field. To ensure that such filter can be estimated, we shift the close-talk microphone by a quarter of the width of the Wiener filter such that the far-field recording is ahead of the corresponding close-talk.

After projecting the close-talk to the corresponding far-field, we observed that the projected signal contains noises, caused by the fact that the far-field recording contains noise. Hence, we apply speech enhancement once more at the very end to clean up the projected close-talk.

Before training, we use Nvidia Parakeet-V2 [30] to filter out all the utterances with WER higher than 30%. We do not perform WER-based filtering on AISHELL datasets, as these are relatively simple and clean data compared to CHiME-6 or AMI.

During mixing, to prevent potential device or channel classification, we simulate far-field mixtures on-the-fly using the speakers from the same session, device, and the same channel.

Real Far-field Mixture Training: First, we find multi-talker conversational segments with max duration 30 s with at least 2 active speakers. Then, we find an enrollment for each speaker that is at least 10 seconds long, the majority of which is speech. Afterwards, we use the best-performing TSE model trained on far-field mixtures to reduce cross-talk on CHiME-6 and AMI close-talk microphones. The Parakeet-V2 ASR is then used to filter out segments with WER above 50% on CHiME-6 and 30% on AMI, keeping the rest of the close-talk far-field pairs for training. At the end, we perform synchronization and close-to-distant projection as described above, and apply speech enhancement to get rid of the projection noise.

IV. Training Curriculum

It has been shown in [31] that multi-stage curriculum pre-training starting from easier data and gradually progressing towards more challenging data increases model performance, generalization, and convergence. Therefore, we performed training in 4-stages.

A. Fully-overlapped Pre-training

We start with pre-training the TSE model from scratch using fully-overlapped Libri2Mix-style data simulated on-the-fly with randomly sampled enrollments.

During this stage, the model should learn the mapping between the enrollment and the mixture, and extract the target speaker. This training phase is performed for 200k training steps.

B. Noisy Simulated Conversations

Next, we initialize the model from the previous stage and train the model using on-the-fly generated conversational mixtures simulated by a modified FastMSS [32] simulator. During this stage, we also introduce reverberation and noise to increase the model's robustness.

C. Far-field Mixtures

After the second pre-training stage, we continue training on on-the-fly simulated far-field mixtures generated from single-talker segments of CHiME-6, AMI, AISHELL-4, and AISHELL-5. Enrollment utterances are randomly sampled from the same segments. Since AMI does not contain background noise, we augment both the mixtures and the enrollment utterances with artificial noise with a probability of 0.2.

D. Real Far-field Mixtures

Lastly, the fourth stage contains real noisy mixture training against targets that were extracted by our previous TSE model. To increase the variability during this training stage, we mix in 20% of synthetic data, and around 30% of far-field mixtures from the third stage of training.

V. Training Setup

During the first two stages, we optimized the following time-domain loss function:

$$\mathcal{L}^{\text{time}} = \log\left(\frac{1}{B}\sum_{i=1}^{B}|\mathbf{y}_i - \hat{\mathbf{x}}_i|\right), \tag{1}$$

where $\hat{\mathbf{x}}_i$ is the prediction and $\mathbf{y}_i$ is the corresponding target, and B is the mini-batch size. Optimizing this loss instead

of SI-SNR or SNR prevents numerical errors for zero targets (i.e., total silence). The loss function was optimized using AdamW [33] optimizer with learning rate (LR) 0.0005 and weight decay 0.001. We used the ReduceLROnPlateau[1] learning rate scheduler according to the validation SNR. The training recordings were randomly cropped to eight seconds while target speaker speech presence was ensured according to the aligned word-level timestamps.

During the last two stages, we used linear LR warmup for 2k steps with peak LR set to $1e-4$ and then applied cosine learning rate scheduler [34]. Because the targets are enhanced projections and not the true clean signals, the phase between the far-field mixture and the projected close-talk might not be well aligned. Hence, we utilize magnitude-based multi-scale STFT (MS-STFT) loss during the last two training stages, defined as:

$$\mathcal{L}^{\text{stft}} = \sum_{i=1}^{B}\sum_{k=1}^{K} \left| |S(\mathbf{y}_i, w_k, h_k)| - |S(\hat{\mathbf{x}}_i, w_k, h_k)| \right|, \quad (2)$$

where $S(\mathbf{y}_i, w_k, h_k)$ represents STFT computed with the window length w_k and hop length h_k. We selected $h_k = \frac{1}{4}w_k$ samples. We used 5 window lengths: $100, 200, 400, 800, 1600$, which corresponds to 6.25 ms, 12.5 ms, 25 ms, 50 ms, 100 ms. As described before, during the fourth stage, we mixed in a small portion of simulated data, so that we could utilize the time-domain loss $\mathcal{L}^{\text{time}}$ on that data.

Besides the reconstruction losses, we also utilized metric-aware losses. First, we used a torch-based implementation of DNSMOS that allowed us to backpropagate through it and optimize the following loss function:

$$\mathcal{L}^{\text{MOS}} = \frac{1}{B}\sum_{i=1}^{B} \left(5.0 - \text{DNSMOS}_{\text{ovrl}}(\hat{\mathbf{x}}_i)\right)^2. \quad (3)$$

Similarly, to slightly push the model to better preserve the enrollment speaker identity, we optimized a cosine similarity between the enrollment and the output:

$$\mathcal{L}^{\text{spk}} = \frac{1}{B}\sum_{i=1}^{B} \left| 1.0 - \cos(\mathbf{e}_i^{\text{enrl}}, \mathbf{e}_i^{\text{pred}}) \right|, \quad (4)$$

where $\mathbf{e}_i^{\text{enrl}}, \mathbf{e}_i^{\text{pred}}$ represent the corresponding speaker embeddings extracted using ResNet34 from the enrollment and the prediction, respectively.

The final loss for the third and fourth stage training is a weighted combination of all the preceeding losses:

$$\mathcal{L} = 10\mathcal{L}^{\text{stft}} + \mathcal{L}^{\text{time}} + 0.01\mathcal{L}^{\text{MOS}} + \mathcal{L}^{\text{spk}}, \quad (5)$$

where $\mathcal{L}^{\text{time}}$ is only computed on the simulated part of the training batch.

VI. Results

Table I shows the progressive improvements in all of the metrics. First, TER significantly improves between the first and the second stage, proving that conversational simulated data with heavy augmentations improve the TSE performance on challenging far-field data. Furthermore, training with 3rd and 4th stage shows that using real noisy and reverberated data for training improves the performance even further.

[1]PyTorch - ReduceLROnPlateau

TABLE I
Comparison between different training stages on the DEV Set. The last "Pure" model was trained only using reconstruction losses.

Model	TER	F1	SPK-SIM	DNSMOS
1st stage	0.53	0.86	0.48	2.42
+ 2nd stage	0.44	0.86	0.47	2.37
+ 3rd stage	0.39	**0.88**	0.54	2.81
+ 4th stage	**0.37**	**0.88**	**0.55**	**2.84**
4 stage, Pure	0.37	0.87	0.47	2.89

In terms of spk-sim and DNSMOS, the main improvement comes from slight metric tuning, which we used to balance out these metrics with TER and F1. However, we observed that training using these metrics did not improve the perceptual quality when listening to the extracted audio. We even observed that over-tuning to speaker similarity causes the TSE model to output target-speaker-like artifacts in the presence of noise even if no one speaks. This further proves how brittle the speaker similarity models are.

To show the effects metric optimization had on our submission, we performed a four-stage training again without using the metric-aware losses during the last two stages. In this case, the model was trained using $\mathcal{L}^{\text{stft}}$ on all data and $\mathcal{L}^{\text{time}}$ on simulated-only data. The last row ("Pure") in Table I shows that we could achieve the same TER and DNSMOS as with metric-aware fine-tuning. Also, $F1$ score only slightly decreased compared to the metric-aware training model due to the lower recall. This was likely caused by not optimizing for spk-sim, which usually forced the model to leak more noise that could potentially trigger the VAD system. The only metric that got significantly decreased is speaker similarity, whose value remains equal to that at the end of the second training stage. After listening to a few examples between the metric-optimized and the pure model, we could not hear any significant difference in terms of extracted speaker identity, proving how overoptimistic the metric value can get once we start optimizing it.

VII. Metric Attack

"When a measure becomes a target, it ceases to be a good measure."

— Goodhart's law

Many speech separation and target speech extraction systems optimize evaluation metrics either explicitly or implicitly during training, including metrics such as SI-SDR and PESQ. From the perspective of Goodhart's Law, this practice fundamentally compromises the validity of such metrics as evaluation tools: once a metric becomes an optimization target, it ceases to function as an independent measure of system quality.

Algorithm 1: Metric Attack.

```
def attack(inp, enrl):
  delta = nn.Parameter(torch.zeros_like(inp))
  optim = Adam([delta], lr=1e-3)

  for i in range(200):
    delta_t = tanh(delta) * 0.01
    audio_t = (inp + delta_t).clamp(-1, 1)
    loss = (stft(audio_t).abs() - stft(inp).abs())
    loss = loss.pow().mean()
    loss = loss + (5.0 - dnsmos(inp))^2
    loss = loss + (1.0 - spksim(inp, enrl))

    optim.zero_grads()
    loss.backward()
    optim.step()

  return audio_t
```

This issue is particularly evident for non-intrusive neural-network-based metrics. For example, clean speech recordings from datasets such as VCTK achieve DNSMOS scores of only around 3.2 [35]. Therefore, observing multiple leaderboard submissions with DNSMOS scores exceeding 3.5 raised concerns about the susceptibility of model-based metrics to over-optimization.

Neural networks are universal function approximators that can approximate arbitrary functions given sufficient capacity and width [36]. Consequently, when a metric is directly optimized during training, a sufficiently expressive model can effectively learn the metric itself and, in the extreme case, internalize at training time an adversarial attack required to maximize its score. The main technical challenge to do so is carefully balancing the optimization process to avoid degrading performance on other evaluation criteria such as TER and F1. Given the better-than-clean-speech DNSMOS scores we observed on the leaderboard, we suspect some teams may have unintentionally performed such an adversarial attack.

To demonstrate the upper bound of metric-aware training and illustrate how metric optimization during training can invalidate the optimized metrics as evaluation measures, we conducted a series of per-utterance adversarial attacks using different metric combinations. We were surprised by how easily a barely perceptible perturbation could be added to the target speech extraction output waveform, yielding state-of-the-art (but perceptually meaningless) DNSMOS and speaker-similarity scores while preserving performance on the remaining metrics, including TER and F1.

We based our attack algorithm on [37], which is straightforward and described in Algorithm 1. It performs 200 steps, in which we first calculate a small delta and add it to the waveform, then compute all the losses and sum them with unit weights. It is important to note that we heavily regularize the adversarial noise energy and also minimize the difference between the original and modified magnitude spectrograms to prevent large changes. Finally, we backpropagate the loss gradients through the metric models and update the delta using the Adam optimizer. We observed during the experiments that no special loss weights, learning rate, or number of steps tuning was required for successful convergence, proving the simplicity of the attack.

TABLE II
COMPARISON BETWEEN BEFORE AND AFTER METRIC ATTACK.

Attack	TER	F1	SPK-SIM	DNSMOS
None	**0.37**	**0.88**	0.52	2.83
DNSMOS	**0.37**	**0.88**	0.52	**4.18**
SPK-SIM (EN)	**0.37**	**0.88**	0.82	2.80
+ DNSMOS	**0.37**	**0.88**	0.82	4.11
+ SPK-SIM (CN)	**0.37**	**0.88**	**0.99**	4.07

Table II shows the model results before the adversarial attack in the first row. It can be seen that attacking DNSMOS only (2nd row) does not change the unattacked metrics. Also, we can see that attacking only the English spk-sim model does not achieve an overall score close to 1, as the Chinese speaker similarity stays at around 0.5 even after the attack. The last row shows the scenario where DNSMOS and speaker models for English and Chinese are attacked. Once again, we can see that TER and F1 stayed the same, while spk-sim and DNSMOS are achieving their extremes. It is noteworthy to observe that attacking only DNSMOS achieves the best DNSMOS score out of all the attacks. The score decreases as we increase the number of simultaneously-attacked metrics. However, the relative change in DNSMOS between attacking all the metrics and only DNSMOS is negligible relative to its value.

Given the fragility of non-intrusive metrics as demonstrated by our attack and also shown in [37], [38], we suggest that the Challenge Organizers either remove DNSMOS and spk-sim when calculating the official ranking or replace them with alternative speech quality and speaker similarity metrics that were not attacked either advertently or inadvertently by submitted systems.

VIII. CONCLUSION

In this work, we described our submission to the RealTSE challenge, consisting of the baseline TSE model architecture and multi-stage data processing and training curriculum. Our results confirmed that each training stage improved the metrics, proving that thorough data preparation with heavy mixture and enrollment augmentations and real noisy data training can push the performance even further. We also showed that metrics such as speaker similarity or DNSMOS are brittle and susceptible to adversarial attacks. To push it to the extreme, we applied a per-waveform adversarial attack, which serves as an upperbound of what can be potentially achieved with training-time optimization if enough capacity and training balance is achieved. This raises challenges for the organization of future challenges in an age where generative models are thriving, and suggests the need for further metric development, which will be focus of our future work.

REFERENCES

[1] 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.

[2] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj, D. Snyder, A. S. Subramanian, J. Trmal, B. B. Yair, C. Boeddeker, Z. Ni, Y. Fujita, S. Horiguchi, N. Kanda, T. Yoshioka, and N. Ryant, "CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," in *Proc. 6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, 2020, pp. 1–7.

[3] M. Van Segbroeck, A. Zaid, K. Kutsenko, C. Huerta, T. Nguyen, X. Luo, B. Hoffmeister, J. Trmal, M. Omologo, and R. Maas, "DiPCo – dinner party corpus," in *Proc. Interspeech*, 2020, pp. 434–436.

[4] 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.

[5] K. Zhang, J. Li, S. Wang, Y. Wei, Y. Wang, Y. Wang, and H. Li, "Multi-level speaker representation for target speaker extraction," in *Proc. ICASSP*, 2025.

[6] 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.

[7] 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.

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

[9] H. He, Z. Shang, C. Wang, X. Li, Y. Gu, H. Hua, L. Liu, C. Yang, J. Li, P. Shi, Y. Wang, K. Chen, P. Zhang, and Z. Wu, "Emilia: A large-scale, extensive, multilingual, and diverse dataset for speech generation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 33, pp. 4044–4054, 2025.

[10] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92)," 2019.

[11] J. Richter, Y.-C. Wu, S. Krenn, S. Welker, B. Lay, S. Watanabe, A. Richard, and T. Gerkmann, "EARS: An anechoic fullband speech dataset benchmarked for speech enhancement and dereverberation," in *Proc. Interspeech*, 2024, pp. 4873–4877.

[12] S. Zhao, Z. Pan, and B. Ma, "ClearerVoice-Studio: Bridging advanced speech processing research and practical deployment," in *Proc. Interspeech*, 2025, pp. 2980–2984.

[13] S. Zhao and B. Ma, "MossFormer: Pushing the performance limit of monaural speech separation using gated single-head transformer with convolution-augmented joint self-attentions," in *Proc. ICASSP*, 2023.

[14] H. Wang, C. Liang, S. Wang, Z. Chen, B. Zhang, X. Xiang, Y. Deng, and Y. Qian, "WeSpeaker: A research and production oriented speaker embedding learning toolkit," in *Proc. ICASSP*, 2023.

[15] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using Kaldi," in *Proc. Interspeech*, 2017, pp. 498–502.

[16] H. Dubey, V. Gopal, R. Cutler, A. Aazami, S. Matusevych, S. Braun, S. E. Eskimez, M. Thakker, T. Yoshioka, H. Gamper, and R. Aichner, "ICASSP 2022 deep noise suppression challenge," in *Proc. ICASSP*, 2022, pp. 9271–9275.

[17] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *Proc. ICASSP*, 2018, pp. 351–355.

[18] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME speech separation and recognition challenge: Dataset, task and baselines," in *Proc. ASRU*, Dec. 2015, pp. 504–511.

[19] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multi-channel acoustic noise database (DEMAND): A database of multichannel environmental noise recordings," in *Proceedings of Meetings on Acoustics*, vol. 19, no. 1. Acoustical Society of America, 2013, p. 035081.

[20] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *Proc. ISMIR*, 2017, pp. 316–323.

[21] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 829–852, 2022.

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

[23] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. Le Roux, "WHAM!: Extending speech separation to noisy environments," in *Interspeech 2019*, 2019.

[24] W. Zhang, R. Scheibler, K. Saijo, S. Cornell, C. Li, Z. Ni, J. Pirklbauer, M. Sach, S. Watanabe, T. Fingscheidt, and Y. Qian, "URGENT challenge: Universality, robustness, and generalizability for speech enhancement," in *Proc. Interspeech*, 2024, pp. 4868–4872.

[25] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The AMI meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*. Springer, 2005, pp. 28–39.

[26] Y. Fu, L. Cheng, S. Lv, Y. Jv, Y. Kong, Z. Chen, Y. Hu, L. Xie, J. Wu, H. Bu, X. Xu, J. Du, and J. Chen, "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] Y. Dai, H. Wang, X. Li, Z. Zhang, S. Wang, L. Xie, X. Xu, H. Guo, S. Zhang, H. Bu, and W. Chen, "AISHELL-5: The first open-source in-car multi-channel multi-speaker speech dataset for automatic speech diarization and recognition," in *Proc. Interspeech*, 2025, pp. 5493–5497.

[28] C. Boeddecker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach, "Front-end processing for the CHiME-5 dinner party scenario," in *Proc. 5th International Workshop on Speech Processing in Everyday Environments (CHiME 2018)*, 2018, pp. 35–40.

[29] T. Nakatani, R. Ikeshita, N. Kamo, M. Delcroix, and S. Araki, "Generating training targets for real-world speech enhancement via close-to-distant microphone projection," in *Proc. ICASSP*, 2026, pp. 18 912–18 916.

[30] M. Sekoyan, N. R. Koluguri, N. Tadevosyan, P. Zelasko, T. Bartley, N. Karpov, J. Balam, and B. Ginsburg, "Canary-1B-V2 & Parakeet-TDT-0.6B-V3: Efficient and high-performance models for multilingual ASR and AST," *arXiv preprint arXiv:2509.14128*, 2025.

[31] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. ICML*, ser. ACM International Conference Proceeding Series, vol. 382. ACM, 2009, pp. 41–48.

[32] A. Polok, I. Medennikov, J. Černockỳ, S. Watanabe, L. Burget, and S. Cornell, "Mind the gap: Impact of synthetic conversational data on multi-talker ASR and speaker diarization," *arXiv preprint arXiv:2605.15442*, 2026.

[33] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.

[34] ——, "SGDR: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=Skq89Scxx

[35] J.-W. Jung, W. Zhang, S. Maiti, Y. Wu, X. Wang, J.-H. Kim, Y. Matsunaga, S. Um, J. Tian, H.-J. Shim, N. Evans, J. S. Chung, S. Takamichi, and S. Watanabe, "The text-to-speech in the wild (TITW) database," in *Interspeech 2025*. ISCA: ISCA, Aug. 2025, pp. 4798–4802.

[36] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608089900208

[37] Y. Liao, Z. Zhang, Z. Sun, Y. Sun, X. Zheng, and X. He, "Beyond waveform robustness: Robust feature-vocoder adversarial attacks on automatic speech recognition," *arXiv preprint arXiv:2606.05678*, 2026.

[38] W.-C. Huang and T. Toda, "Attacking UTMOS: Probing the robustness of a speech quality assessment model," 2026. [Online]. Available: https://arxiv.org/abs/2606.31105