



# The USTC-NERC SLIP Systems for the CHiME-8 NOTSOFAR-1 Challenge

Shutong Niu<sup>1</sup>, Ruoyu Wang<sup>1</sup>, Jun Du<sup>1,\*</sup>, Gaobin Yang<sup>1</sup>, Yanhui Tu<sup>2</sup>, Siyuan Wu<sup>2</sup>, Shuangqing Qian<sup>2</sup>,  
Huaxin Wu<sup>2</sup>, Haitao Xu<sup>2</sup>, Xueyang Zhang<sup>2</sup>, Guolong Zhong<sup>2</sup>, Xindi Yu<sup>2</sup>, Jieru Chen<sup>2</sup>,  
Mengzhi Wang<sup>2</sup>, Di Cai<sup>2</sup>, Tian Gao<sup>2</sup>, Genshun Wan<sup>2</sup>, Feng Ma<sup>2</sup>, Jia Pan<sup>2</sup>, Jianqing Gao<sup>2</sup>

<sup>1</sup>University of Science and Technology of China, China  
<sup>2</sup>iFlytek Research, China

{niust,wangruoyu}@mail.ustc.edu.cn, jundu@ustc.edu.cn

## Abstract

This technical report outlines our submission system for the CHiME-8 NOTSOFAR-1 Challenge [1]. The primary difficulty of this challenge is the dataset recorded across various conference rooms, which captures real-world complexities such as high overlap rates, background noises, a variable number of speakers, and natural conversation styles. To address these issues, we optimized the system in several aspects: For front-end speech signal processing, we introduced a data-driven joint training method for diarization and separation (JDS) to enhance audio quality. Additionally, we also integrated traditional guided source separation (GSS) for multi-channel track to provide complementary information for the JDS. For back-end speech recognition, we enhanced Whisper with WavLM, ConvNeXt, and Transformer innovations, applying multi-task training and Noise KLD augmentation, to significantly advance ASR robustness and accuracy. Our system attained a Time-Constrained minimum Permutation Word Error Rate (tcpWER) of 14.265% and 22.989% on the CHiME-8 NOTSOFAR-1 Devset-2 multi-channel and single-channel tracks, respectively.

**Index Terms:** CHiME challenge, speaker diarization, speech separation, speech recognition, joint training

## 1. System Description

Our overall system follows the process illustrated in Fig. 1. First, the diarization system is used to predict the speaker’s time distribution, which is then utilized to perform speech separation. Then, the separated speech is sent to the speech recognition system. In the following sections, we will describe the single-channel and multi-channel systems in detail.

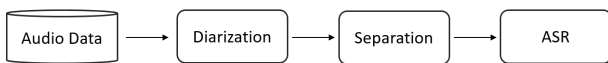


Figure 1: Overall framework of the system.

### 1.1. Multi-channel System

#### 1.1.1. Diarization

Fig. 2 illustrates the diarization component of the multi-channel system. For the original multi-channel data, we first perform weighted prediction error (WPE) algorithm, followed by overlap segment detection. We used the same architecture as the separation model of the official CSS baseline [2] for the overlapping segment detection model. However, we changed the

\* Corresponding author

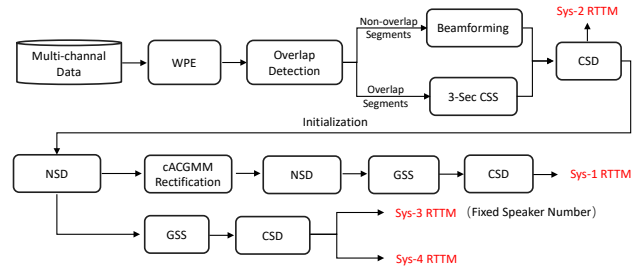


Figure 2: The diarization pipeline for multi-channel system.

sliding window length to 800 frames (12.8 seconds) and modified the final prediction output to the frame-level binary classification using a linear layer. For the detected overlapping segments, we employ the multi-channel 3-second continuous speech separation (CSS) method to effectively isolate each speaker’s speech. We modified the official baseline architecture [2] for the CSS on overlapping segments by adding a classification network for overlapping segment detection and conducting joint training for separation and overlap segment detection. We used the official model for initialization and conducted joint training to enhance the separation model’s ability to differentiate between overlapping and non-overlapping segments. We only used the predicted results from the separated parts. The sliding window length was kept at 3 seconds. We used the same training and inference procedures as the official baseline. The training data remained consistent with the baseline, utilizing only the official simulated data [3]. For non-overlapping segments, we enhance the multi-channel speech using the MVDR beamformer [4].

We conduct the clustering-based speaker diarization (CSD) method on these pre-processed speech, resulting in preliminary speaker diarization priors, referred to as ‘Sys-2 RTTM’ in Fig. 2. For CSD system, we use the spectral clustering algorithm. We leverage the ResNet-221 model for speaker embedding extraction, which is trained on the VoxCeleb [5] and LibriSpeech datasets. To obtain different diarization priors, we further apply various processing techniques to the speech used for clustering. Firstly, we use the results obtained from clustering as initial priors, and feeding them into the neural network-based speaker diarization (NSD) system to achieve more precise speaker boundary information. The NSD employed in our system is the memory-aware multi-speaker embedding with sequence-to-sequence architecture (NSD-MS2S) [6, 7], which combines the advantages of memory-aware multi-speaker embedding and sequence-to-sequence architecture. For the multi-channel track, we input different channels separately and then

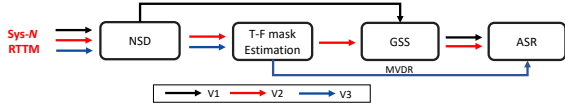


Figure 3: The separation pipeline for multi-channel system.

averaged the posterior probabilities of the different channels to obtain the final result for one session. The NSD uses the 800-frame window length with a frame length of 10ms, resulting in a total window length of 8 seconds. Then, following our previous methods in CHiME-7 DASR Challenge [8], we conduct cACGMM rectification on the original audios, adopting a window length of 120 seconds and a window shift of 60 seconds. This rectification utilizes the previous NSD decoding result as the initialization mask. By implementing a threshold on the spectrum mask of the cACGMM, we obtain a refined secondary initialization of diarization results for the NSD system. After the official GSS initialized with the second NSD decoding results, we perform the re-clustering to obtain better diarization priors (Sys-1 RTTM). Additionally, the decoding results from the first NSD can be directly utilized to initialize the GSS, thereby generating separated audios. For these separated audios, we conducted re-clustering with the fixed number of speakers (maintaining the global number of speakers within a session) and the non-fix number of speakers (the original version), resulting in two initial diarization priors, namely ‘Sys-3 RTTM’ and ‘Sys-4 RTTM’, respectively.

### 1.1.2. Separation

After obtaining the RTTMs from the diarization system, we acquire information about the speaker distribution. Utilizing this information, we proceed with various versions of speech separation as depicted in Fig. 3. For the first system (V1), we utilize the NSD to optimize the time boundaries. The optimized results are then used to initialize the GSS algorithm, resulting in the separated audios. For the second system (V2), we utilize the time masks estimated from the NSD as the inputs for JDS system. This guides the JDS system in estimating time-frequency (T-F) soft masks. These T-F masks are then employed to initialize the GSS in the T-F domain, thus providing the GSS with initialization information in both time and frequency dimensions. For the third system (V3), we directly utilize the T-F masks predicted by the JDS system to guide the MVDR beamforming, while still employing the time boundaries provided by the NSD to get the separated speech segments. Fig. 4 shows the overall framework of multi-channel joint training method for diarization and separation (JDS). The JDS system comprises two main components: the speaker diarization module and the speech separation module. Based on the original end-to-end speaker diarization systems, the JDS system serially integrates the separation module. This helps the speech separation system accurately identify the number of speakers and the corresponding identities. This information also facilitates the speech separation model in more effectively distinguishing between different speakers. Consequently, the separation module in JDS system primarily maps the time information of various speakers to time-frequency information, which significantly simplifies the speech separation process. In our system, the JDS uses a window length of 800 frames with a frame length of 16ms, resulting in a total window length of 12.8 seconds.

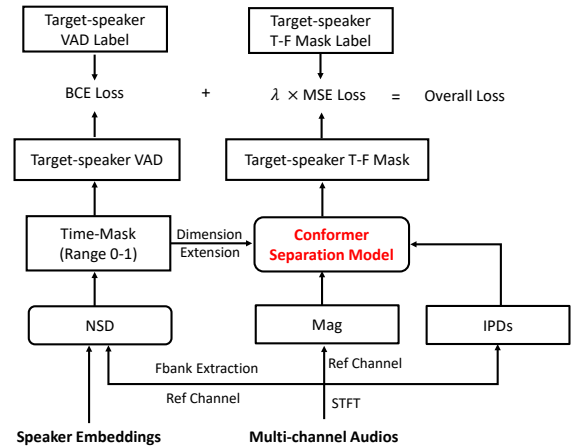


Figure 4: Overall framework of multi-channel JDS method.

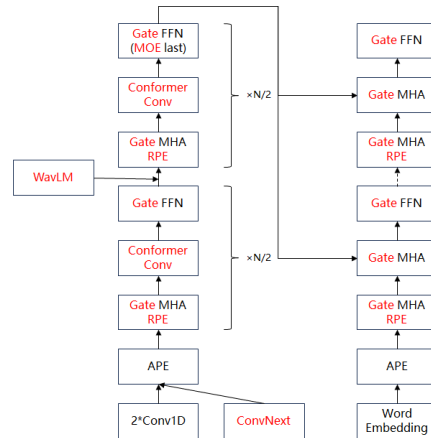


Figure 5: The architecture of Enhanced Whisper.

### 1.1.3. Speech Recognition

For automatic speech recognition tasks, we leverage Whisper [9], a state-of-the-art open-source model renowned for its high accuracy. Whisper follows an encoder-decoder architecture based on the Transformer framework. The input to the model is represented as a log Mel-spectrogram. Both the encoder and decoder components feature absolute positional encoding and are composed of several transformer layers. Notably, the encoder contains two layers of 1D convolution preceding the absolute positional encoding stage, which aids in extracting local features from the input audio data.

We introduce Enhanced Whisper, a variant that introduces a series of enhancements to the base Whisper model. An overview of the modified architecture is illustrated in Fig. 5. To refine input feature representation, we drew inspiration from the CHiME-7 DASR Challenge [8], leveraging features extracted from self-supervised pre-trained models, particularly WavLM [10]. Our experiments involved systematically integrating these WavLM-derived features at various stages within the Whisper encoder, including the initial, intermediate, and final layers. We observed that injecting these features at the intermediate layer of the encoder resulted in a slight yet noticeable improvement in performance. The outputs from WavLM and the intermedi-

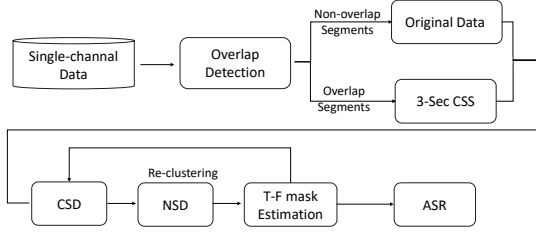


Figure 6: The framework of single-channel system.

ate layer of the Whisper encoder are integrated via a concatenation operation, followed by a linear transformation to ensure compatibility with the original feature dimensions of the model. Concerning downsampling convolutions, the baseline Whisper model utilizes two layers of 1D convolution. Inspired by recent advancements like NextFormer [11], we augmented the original Whisper model with a ConvNeXt structure, running in parallel to the standard 1D convolutions. The ConvNeXt output is added to the original Conv1D output after a linear transformation and then input into the transformer.

Regarding positional encoding, Whisper initially relies on absolute positional encoding. However, empirical evidence suggests that absolute positional encoding exhibits limitations in robustness compared to relative positional encoding [12]. Motivated by these findings, we adopted bias relative positional encoding [13] within our enhanced model, aiming to improve its resilience and performance consistency across varying input lengths.

In terms of the Transformer block, we took cues from relevant research [14, 15] to integrate a sigmoid gating mechanism. Specifically, the input is projected through a weight matrix ( $W$ ), followed by a sigmoid activation function. The output of this operation is then scaled by a factor of 2 before being element-wise multiplied with the original output, effectively controlling the flow of information within the Transformer block. Additionally, we explored the insertion of a depthwise convolution module, akin to those featured in Conformer [16] models, following the Multi-Head Attention (MHA) layers. This architecture enhances the model’s ability for localized modeling. Furthermore, we augmented the final layer of the encoder with a Mixture of Experts (MoE) [17] component, aimed at enhancing the model’s representational capacity.

## 1.2. Single-channel System

The framework of the single-channel system is illustrated in Fig. 6. Like the multi-channel system, the single-channel system also begins with speaker diarization followed by speech separation and ASR. However, unlike the multi-channel system, each module in the single-channel system (including the overlap detection, CSS, CSD, NSD, and T-F mask estimation) receives only single-channel audios or features as inputs. To get the separated audios for each speaker, the amplitude spectral features of the original mixed audio are multiplied by the T-F masks, and an inverse STFT transformation is performed. Furthermore, re-clustering the separated audios can enhance the precision of the speaker diarization priors, as illustrated at the bottom of the Fig. 6. For ASR, we use the same model for decoding as in the multi-channel system.

Table 1: The training sets of speech recognition.

Duration (h)	Corpus	Sample Scale
14	Train-set-1 MC GSS	1
16	Train-set-2 MC GSS	1
10	Dev-set-1 MC GSS	1
14	Train-set-1 MC GSS with timestamp	1
16	Train-set-2 MC GSS with timestamp	1
10	Dev-set-1 MC GSS with timestamp	1
14	Train-set-1 MC NN	1
16	Train-set-2 MC NN	1
10	Dev-set-1 MC NN	1
14	Train-set-1 MC ch0 NN	1
16	Train-set-2 MC ch0 NN	1
10	Dev-set-1 MC ch0 NN	1
960	LibriSpeech	1

## 1.3. Datasets

### 1.3.1. Diarization and Separation

For the speaker diarization system, the training data comprises the officially simulated training dataset [3], Train-set-1 [18], Train-set-2 [18] and Dev-set-1 [18]. We also employed LibriSpeech, MUSAN noise [19] and the noises in officially simulated training dataset [18] to simulate the diarization training data<sup>1</sup>. Additionally, we also use the near-field recordings from Train-set-1, Train-set-2 and Dev-set-1 as clean data to simulate multi-channel speaker diarization training data. For speech separation, we use the officially simulated training dataset and also use the near-field recordings from Train-set-1, Train-set-2 and Dev-set-1 to simulate the separation training data.

### 1.3.2. Speech Recognition

The ASR systems were trained using official NOTSOFAR-1 training data and the open-source LibriSpeech dataset with data augmentation methods. The data augmentation methods included speed perturbation and MUSAN noise [19] addition. The specific composition of the training data is shown in Table 1. We utilized multi-channel (MC) data processed by both Guided Source Separation (GSS) and Neural Networks (NN), and introduced a word-level timestamp prediction task into the GSS data. Specifically, for GSS, we used oracle RTTM labels on the multi-channel data to perform GSS, resulting in separated speech segments with corresponding speaker identities and timestamps that match the ASR annotations. Through this correspondence, we matched the recognition labels to the separated results. For NN-based separation, we directly apply the JDS method for separation and segment the separated audios according to the time steps of oracle RTTM. We found that this multitask training approach led to a slight improvement in recognition accuracy. We also adopted the practice from Whisper of providing the transcribed text from the preceding utterance as previous-text conditioning, which has noticeably improved the recognition rate. Contrary to using official single-channel (SC) data, we selected NN-processed MC channel 0 (ch0) data as our single-channel training input, observing superior performance with this choice. Drawing inspiration from the principles of RDrop [20], we developed a novel data augmentation technique called Noise KLD. This approach entails separately feeding both the original and augmented data samples into the model. Consistency between the model’s predictions for the original and augmented data is ensured by ap-

<sup>1</sup>[https://github.com/jsalt2020-asrdiar/jsalt2020\\_simulate](https://github.com/jsalt2020-asrdiar/jsalt2020_simulate)

plying Kullback-Leibler divergence (KLD) loss as a regularizer. Through extensive experimentation, we discovered that this method outperforms conventional data augmentation strategies in terms of boosting model performance and generalization.

## 2. Results

For diarization, the training requires approximately 88 hours, and testing all sentences in Dev-set-2 takes about 1 hour. For the JDS system, training takes approximately 4 days, while testing all sentences in Dev-set-2 requires about 1 hour. For ASR, training requires about 20 hours, and testing all sentences in Dev-set-2 consumes about 6 hours. Typically, we conduct our training on A100 GPUs and perform testing on V100 or A40 GPUs.

### 2.1. Overall Results

#### 2.1.1. Multi-channel System

Table 2 presents the tcpWER (%) of our multi-channel system on Dev-set-2, where ‘Sys- $N$  RTTM’ corresponds to the system depicted in Fig. 2, and ‘V\*’ corresponds to the system shown in Fig. 3. For each system, we fused the posterior probabilities from three different Whisper models (enhanced large-v2, enhanced large-v3, and enhanced large-v3 trained with more data simulated from Librispeech). The enhanced Whisper models were fine-tuned using the official Whisper large v2 (enhanced large-v2) and v3 (enhanced large-v3 and enhanced large-v3 trained with more data simulated from Librispeech) parameters for initialization. The last row ‘Fusion’ indicates the average of posterior probabilities across 9 ( $3 \times 3$ ) systems using the same speaker diarization priors. Finally, in the multi-channel track, we submit the fusion results of each ‘Sys- $N$  RTTM’ (last column).

Table 2: *TcpWER (%) comparisons on the multi-channel track on Dev-set-2.*

Sep \ Dia	RTTM			
	Sys-1	Sys-2	Sys-3	Sys-4
V1	14.953	14.649	15.116	14.571
V2	14.911	14.595	15.086	14.547
V3	15.577	15.160	15.703	15.018
Fusion	14.681	14.286	14.847	14.265

#### 2.1.2. Single-channel System

Table 3 presents the tcpWER (%) of our single-channel system on Dev-set-2. The diarization priors are derived from NSD and re-clustering, as illustrated in Fig. 6. These priors are then input into the JDS system, from which separated audio is obtained via multiplying T-F masks and amplitude spectrum. Similarly, for each subsystem, we have fused posterior probabilities from three different Whisper models (enhanced large-v2, enhanced large-v3, and enhanced large-v3 trained with more data simulated from Librispeech).

### 2.2. Ablation Results

To better illustrate our system, we present some ablation experiments conducted during the challenge, along with some discussions in this section. We will focus on showing the ablation

Table 3: *TcpWER (%) comparisons on the single-channel track on Dev-set-2.*

Sep \ Dia	NSD	Re-clustering
	JDS	24.611

Table 4: *Evaluation results [21] of the proposed diarization module on Dev-set-2 multi-channel track. The ASR model is based on Whisper-large-v3, fine-tuned on Train-set-1/2 datasets. Note that we removed the anomalous session ‘MTG-30522’.*

	Initialization				NSD Decoding				ASR
	FA	MISS	SpkErr	DER	FA	MISS	SpkErr	DER	tcpWER
Stage 1 (w/o CSS)	4.72	25.36	2.56	32.65	4.82	7.46	2.96	15.24	-
Stage 1	5.90	15.17	2.36	23.43	4.77	7.37	2.26	14.40	12.87
Stage 2	7.51	11.67	1.89	21.07	4.51	7.00	2.47	13.97	14.13
Stage 3 (w/o filter)	3.50	8.38	4.25	16.12	4.13	7.44	3.00	14.58	13.65
Stage 3	3.71	13.87	1.50	19.09	4.50	7.47	2.21	14.19	12.83

results of three main modules: diarization, speech separation, and speech recognition, respectively.

#### 2.2.1. Diarization

Table 4 presents the diarization results at different stages on Dev-set-2, where we define the stages based on the number of NSD decodings in Fig. 2. The first stage corresponds to the first decoding of NSD in Fig. 2 and the CSD results used for initialization. The second stage refers to the second decoding of NSD in Fig. 2 and the cACGMM rectification-based diarization results used for initialization. The third stage corresponds to the CSD results ‘Sys-1 RTTM’ in Fig. 2, along with the corresponding NSD results. For more detailed definitions, please refer to this paper [21]. The term ‘filter’ refers to the process of eliminating segments that contain fewer than one word using a speech recognition model to prevent interference from incomplete or very short segments. As we can see, the introduction of CSS significantly improves the performance of the diarization in stage 1, effectively reducing the MISS errors in the CSD results. At the same time, stage 2 shows a relatively effective improvement in DER compared to stage 1, but the recognition performance actually become worsens. Finally, the filtering operation in stage 3 can effectively reduce SpkErr errors. However, the final DERs still don’t show improvement compared to stage 2. To explore the performance improvements brought by real data to the diarization module, we provide a brief comparison of the performance of diarization models trained with different datasets in Table 5. As shown in the table, adding real training data in NOTSOFAR [18] leads to a substantial improvement (DER from 21.51% to 16.52%).

#### 2.2.2. Separation

Table 6 presents the results of different speech separation methods with a fixed back-end recognition model. From this table, we can observe that adding a classification network for overlapping segment detection brings some improvement to the speech separation results (from 26.68% to 25.14%). Additionally, JDS shows a noticeable improvement compared to the CSS method (from 25.14% to 20.62%), primarily due to its ability to utilize more accurate speaker time boundaries. Furthermore, we

Table 5: *DER (%) comparisons of different NSD training data sets on Dev-set-1 multi-channel track (without ‘rockfall\_1’).*

Training Data Sets	DER (%)
LibriSpeech Simulated Data + NOTSOFAR Simulated Data	21.51
+ Train-set-1/2 MC (split into single channel)	16.52

Table 6: *TcpWER (%) comparisons of different separation method on Train-set-1 multi-channel track (plaza\_0). The ASR model is based on original Whisper-large-v3. The separation training dataset is NOTSOFAR simulated data [3].*

Separation Methods	TcpWER (%)
CSS (3-Sec) + MVDR	26.68
CSS (3-Sec) + Overlap Detection (3-Sec) + MVDR	25.14
JDS (3-Sec) + MVDR	20.62
JDS (3-Sec) + Dia.Mask + MVDR	20.29
JDS (3-Sec) + Dia.Mask + CSD.RTTM + MVDR	19.95
JDS (8-Sec) + CSD.RTTM + MVDR	18.57
JDS (8-Sec) + Dia.Mask + CSD.RTTM + MVDR	17.47

can improve performance further (from 20.62% to 20.29%) by using speaker boundaries trained on more data (referred to as ‘Dia.Mask’ in the table) instead of only relying on the speaker time boundaries from JDS (the ‘Time-Mask’ in Fig 4). During the decoding process, we can also select matching CSD RTTM to segment the speech separation results. Since CSD results may have lower confusion errors, this can positively impact recognition results (from 20.29% to 19.95%). Finally, extending the window length of JDS from 3 seconds to 8 seconds leads to further improvements in speech separation performance.

### 2.2.3. Speech Recognition

Table 7 presents the performance improvements achieved through various modifications to the speech recognition model architecture. The term ‘Long Prompt’ refers to using the previous decoding history as a decoder prompt, which follows the methods used in Whisper. The results indicate that both the MOE and RPE methods effectively enhance speech recognition performance. Additionally, incorporating WavLM features further improves the performance of the speech recognition model. Table 8 shows the results of the backend ASR with different training datasets. ‘All-set MC GSS/NN’ means the sum of ‘Train-set-1/2 MC NN/GSS’ and ‘Dev-set-1 MC NN/GSS’ in Table 1. It demonstrates that real training data in NOTSOFAR [18], processed through oracle GSS, can significantly improve the performance of the backend ASR.

## 3. Conclusion

The NOTSOFAR-1 challenge explored a meaningful scenario, namely real-world far-field multi-speaker meeting environments. This includes many challenges that speech signal processing systems need to deal with in practical applications, including speaker movement, high speech overlap rates, rapid changes in speakers, various noise and reverberation, and a variable number of speakers. In order to deal with these challenges, we proposed some methods from the front-end signal processing and back-end speech recognition, mainly including the use of data-driven NSD models to predict speaker time boundaries, combining traditional spatial information-based GSS and data-

Table 7: *TcpWER (%) comparisons of different ASR models on Dev-set-1 (Oracle GSS using RTTM label). The training sets is ‘Train-set-1/2 MC GSS’.*

Models	TcpWER (%)
Whisper large v3	8.46
Whisper large v3 + RPE	8.42
Whisper large v3 + MOE	8.34
+ Timestamp (as showed in Table 1)	8.25
+ RPE + Long Prompt + Noise KLD	7.61
+ WavLM	7.50

Table 8: *TcpWER (%) comparisons of Whisper large v3 models with different training data sets on Dev-set-2 (Oracle GSS using RTTM label). Note that we removed the anomalous session ‘MTG-30522’.*

Training Data Set	TcpWER (%)
Original Datasets	16.57
Original Datasets + Train-set-1/2 MC GSS	12.07
All-set MC GSS/NN + LibriSpeech Simulated Data	9.87

driven JDS models for speech separation, as well as the construction of speech recognition training data and the modification of speech recognition model architecture. In this challenge, we found that the NSD method requires effectively matched training data to improve performance, including both real and simulated datasets. Additionally, multi-stage optimization of the diarization priors of NSD proved to be an important factor for diarization performance. Furthermore, incorporating time boundary information from diarization can help the speech separation model achieve better separation results with more accurate time boundaries, thereby effectively improving speech recognition performance. Finally, fine-tuning with matched datasets and improving model architecture are still crucial methods for enhancing speech recognition performance. In the NOTSOFAR-1 challenge, our system achieved the tcpWERs of 22.2% and 10.8% in the single-channel and multi-channel tracks of the evaluation set, respectively, winning first place in both tracks.

## 4. Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants No. 62171427.

## 5. References

- [1] A. Vinnikov, A. Ivry, A. Hurvitz, I. Abramovski, S. Koubi, I. Gurvich, S. Peer, X. Xiao, B. M. Elizalde, N. Kanda, X. Wang, S. Shaer, S. Yagev, Y. Asher, S. Sivasankaran, Y. Gong, M. Tang, H. Wang, and E. Krupka, “Notsofar-1 challenge: New datasets, baseline, and tasks for distant meeting transcription,” in *Inter-speech 2024*, 2024, pp. 5003–5007.
- [2] “CHiME-8 Baseline System,” <https://www.chimechallenge.org/current/task2/baseline>, 2024.
- [3] “CHiME-8 Simulated Training Dataset,” <https://www.chimechallenge.org/current/task2/datasimulated-training-dataset>, 2024.
- [4] T. Higuchi, N. Ito, S. Araki, T. Yoshioka, M. Delcroix, and T. Nakatani, “Online MVDR beamformer based on complex

- Gaussian mixture model with spatial prior for noise robust ASR,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 780–793, 2017.
- [5] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, “Voxceleb: Large-scale speaker verification in the wild,” *Computer Speech & Language*, vol. 60, p. 101027, 2020.
- [6] M.-K. He, J. Du, Q.-F. Liu, and C.-H. Lee, “ANSD-MA-MSE: Adaptive Neural Speaker Diarization Using Memory-Aware Multi-Speaker Embedding,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1561–1573, 2023.
- [7] G. Yang, M. He, S. Niu, R. Wang, Y. Yue, S. Qian, S. Wu, J. Du, and C.-H. Lee, “Neural speaker diarization using memory-aware multi-speaker embedding with sequence-to-sequence architecture,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 11 626–11 630.
- [8] R. Wang, M. He, J. Du, H. Zhou, S. Niu, H. Chen, Y. Yue, G. Yang, S. Wu, L. Sun, Y. Tu, H. Tang, S. Qian, T. Gao, M. Wang, G. Wan, J. Pan, J. Gao, and C.-H. Lee, “The ustc-nerclip systems for the chime-7 dasr challenge,” *ArXiv*, vol. abs/2308.14638, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:261244449>
- [9] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” *ArXiv*, vol. abs/2212.04356, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:252923993>
- [10] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, M. Zeng, and F. Wei, “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, pp. 1505–1518, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:239885872>
- [11] Y. Jiang, J. Yu, W. Yang, B. Zhang, and Y. Wang, “Nextformer: A convnext augmented conformer for end-to-end speech recognition,” 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:250113612>
- [12] P. Zhou, R. Fan, W. Chen, and J. Jia, “Improving generalization of transformer for speech recognition with parallel schedule sampling and relative positional embedding,” *ArXiv*, vol. abs/1911.00203, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:207870654>
- [13] Z. Dai, H. Liu, Q. V. Le, and M. Tan, “Coatnet: Marrying convolution and attention for all data sizes,” *ArXiv*, vol. abs/2106.04803, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:235376986>
- [14] Y. Sun, L. Dong, S. Huang, S. Ma, Y. Xia, J. Xue, J. Wang, and F. Wei, “Retentive network: A successor to transformer for large language models,” *ArXiv*, vol. abs/2307.08621, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:259937453>
- [15] B. Peng, E. Alcaide, Q. G. Anthony, A. Albalak, S. Arcadinho, S. Biderman, H. Cao, X. Cheng, M. Chung, M. Grella, G. Kranthikiran, X. He, H. Hou, P. Kazienko, J. Kocoń, J. Kong, B. Koptyra, H. Lau, K. S. I. Mantri, F. Mom, A. Saito, X. Tang, B. Wang, J. S. Wind, S. Wozniak, R. Zhang, Z. Zhang, Q. Zhao, P. Zhou, J. Zhu, and R. Zhu, “Rwkv: Reinventing rns for the transformer era,” in *Conference on Empirical Methods in Natural Language Processing*, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:258832459>
- [16] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented transformer for speech recognition,” *ArXiv*, vol. abs/2005.08100, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:218674528>
- [17] Z. You, S. Feng, D. Su, and D. Yu, “Speechmoe: Scaling to large acoustic models with dynamic routing mixture of experts,” in *Interspeech*, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:234094030>
- [18] “CHiME-8 Meetings Recordings Dataset,” <https://www.chimechallenge.org/current/task2/datasimulated-training-dataset>, 2024.
- [19] D. Snyder, G. Chen, and D. Povey, “MUSAN: A Music, Speech, and Noise Corpus,” 2015, arXiv:1510.08484v1.
- [20] W. Ji, S. Zan, G. Zhou, and X. Wang, “Research on an improved conformer end-to-end speech recognition model with r-drop structure,” *ArXiv*, vol. abs/2306.08329, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:259165369>
- [21] R. Wang, S. Niu, G. Yang, J. Du, S. Qian, T. Gao, and J. Pan, “Incorporating spatial cues in modular speaker diarization for multi-channel multi-party meetings,” 2024. [Online]. Available: <https://arxiv.org/abs/2409.16803>