



STCON System for the CHiME-8 Challenge

Anton Mitrofanov^{1,2}, Tatiana Prisyach¹, Tatiana Timofeeva^{1,2}, Sergei Novoselov^{1,2}, Maxim Korenevsky¹, Yuri Khokhlov¹, Artem Akulov¹, Alexander Anikin¹, Roman Khalili¹, Iurii Lezhenin^{1,3}, Aleksandr Melnikov¹, Dmitriy Miroshnichenko¹, Nikita Mamaev¹, Ilya Odegov¹, Olga Rudnitskaya¹, Aleksei Romanenko^{1,2}

¹STCON LLC., Kingdom of Saudi Arabia

²National center for cognitive research of ITMO University, Russia

³Peter the Great St.Petersburg Polytechnic University, Russia

{ mitrofanov-aa, prisyach, timofeeva, novoselov, korenevsky, khokhlov, akulov, anikin, khalili, lezhenin, melnikov-a, miroshnichenko, mamaev-n, odegov, rudnitskaya, romanenko }@speechpro.com

Abstract

This paper describes the STCON system for the CHiME-8 Challenge Task 1 (DASR) aimed at distant automatic speech transcription and diarization with multiple recording devices. Our main attention was paid to carefully trained and tuned diarization pipeline and speaker counting. This allowed to significantly reduce diarization error rate (DER) and obtain more reliable segments for speech separation and recognition. To improve source separation, we designed a Guided Target speaker Extraction (G-TSE) model and used it in conjunction with the traditional Guided Source Separation (GSS) method. To train various parts of our pipeline, we investigated several data augmentation and generation techniques, which helped us to improve the overall system quality.

Index Terms: speech recognition, speaker diarization, WPE, GSS, G-TSE, speaker counting, NSD-MS2S, RIR generator, WavLM, ZipFormer, CHiME-8.

1. System description

In general, our system follows a pattern that has worked well in previous CHiME challenges [1], [2], namely, speaker diarization, source separation and speech recognition. For the current challenge we have paid much attention to the improvement of the diarization and speaker counting. We applied sophisticated multi-step procedure to obtain high-quality clustering-based diarization which is followed by advanced neural diarization system providing accurate speakers' activity bounds. Several variants of Guided Source Separation (GSS) [3] as well as Target Speaker Extraction (TSE) [4] are then used to extract each speaker's utterances and recognize them with a carefully tuned Automatic Speech Recognition (ASR) models. Finally, ASR results are re-scored with a large Language Model and fused to provide a final speaker-attributed transcription results. The main components of the system are shown on Fig. 1(a).

2. Speaker diarization

Since the boundaries of the sentences in the training data are not always accurate and there are pauses inside them, we made a forced alignment of all data based on Individual Headset Microphone (IHM) recordings. This provided a more accurate references for DER measurements, which is important for training good diarization models. All DER results below correspond to these aligned segments, not original ones.

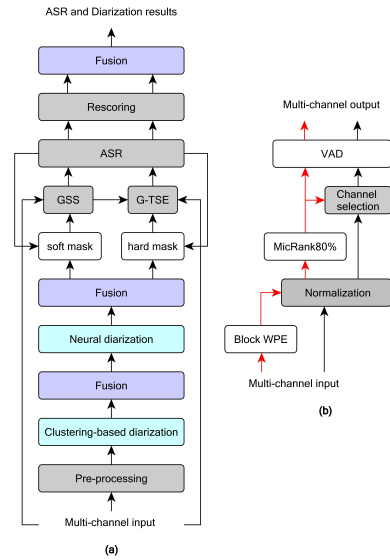


Figure 1: (a) STCON system. (b) Pre-processing diagram.

2.1. Clustering-based diarization

Basic clustering-based diarization begins with a short but important step of normalization. Each session recording is first clipped by a certain threshold value and then re-normalized to the maximum amplitude. The threshold value is chosen as a pre-defined percentile of absolute signal values. This allows to get rid of loud knocks and claps. Normalization is applied to both original recordings and ones processed with 2 minute-block Weighted Prediction Error (WPE) dereverberation [5]. Normalized recordings are then processed by an Envelope Variance [6] based channel selection algorithm (hereinafter referred to as MicRank) to select 80% best channels for each session. MicRank is applied to the WPE-processed recordings and corresponding channels are also selected from the normalized original recordings. All subsequent processing is processed channel-wise.

The pre-processing steps are depicted at Fig.1(b).

Voice Activity Detector (VAD) is applied to select segments where at least one speaker is active. These segments are passed into the speaker counting module (SCM). SCM includes speaker embeddings extraction, clustering and clusters post-processing. Two types of embeddings are used. First type is multi-speaker embeddings extracted with an in-house model

Table 1: Clustering-based diarization results.

System	max_spk	DER / speaker count accuracy							AVG
		chime6		dipco		mixer6		notsofar1	
		dev	eval	dev	eval	dev	eval	dev	
baseline [14]	4	26.8	-	24.78	-	16.53	-	-	-
	8	36	-	26	-	24	-	-	-
single_orig*	8	25.3/0.87	31.9/0.87	23.7/1	20.0/0.85	16.3/0.91	7.6/1	20.0/0.86	20.6/0.88
single_wpe*	8	24.1/1	30.9/1	22.4/1	17.1/0.85	12.8/0.97	7.7/0.97	20.8/0.85	19.4/0.87
fusion	8	23.5/1	29.6/1	21.4/1	17.3/0.85	13.0/0.98	7.5/1	13.0/0.89	17.9/0.90

* The best of 6 systems with different parameters thr and VAD segments.

based on Wav2Vec2.0 XLS-R 53¹. The model is similar to one described in [7] and was finetuned on VoxCeleb1,2 data. This model has Attention-based Encoder-Decoder architecture and is trained to extract several embeddings for segments with mixed speech, i.e. to act as a mixed speech detector. Several embeddings extracted from each segment are compared to each other using cosine similarity. If it exceeds a certain threshold, corresponding speakers are merged together. This processing makes it possible to select a subset of single-speaker frames only. Embeddings of the second type are extracted using the SpeechBrain Ecapa-TDNN model [8]. Both types of embeddings are concatenated and the resulting embeddings are normalized to unit length. UMAP [9] dimensionality reduction down to 12 is applied followed by the GMM clustering [10] of embeddings on the pre-selected single-speaker frames. The maximal number of clusters is set to 8 but some of them can be subsequently merged together based on the cosine similarity and rejected based on cluster size threshold thr ². Besides, a Wav2Vec2.0-based extractor can work as non-speech classifier and remove a cluster if it corresponds to non-speech. After the number of clusters is determined and fixed, all mixed speech frames are processed. Each such frame is greedily attracted to the cluster whose centroid is the nearest to the corresponding embedding. The scheme of the clustering-based diarization is depicted on Fig. 2.

2.2. Neural diarization (ND)

This module is based on the open-sourced CHiME7 winner solution Neural Speaker Diarization using Memory-Aware and Multi-Speaker embeddings (NSD-MS2S)³. Several models for NSD-MS2S were trained. Each of them was pretrained on a large set of simulated CHiME8-like data and then fine-tuned on all training data from all Multiple Distant Microphone (MDM).

To prepare the simulated dataset, a set of artificial Room Impulse Responses (RIRs) was generated, which are similar to those of CHiME8 data [11]. Five different RIR-classifiers [12] were trained, each on different 100K artificially generated⁴ RIRs. Classifiers were trained on Librispeech [13] data augmented by noises extracted from CHiME8 data. All recordings from the CHiME8 training data were classified⁵ by each of classifier and 20% RIRs with the highest probability on each subset of CHiME8 data were selected, resulting in 20K selected RIRs per classifier or 100K RIRs in total. The room parameters of the selected RIRs were used to generate a set of multichannel RIRs

¹<https://huggingface.co/facebook/wav2vec2-large-xlsr-53>

²Clusters with a size N smaller than N_{max}/thr are rejected

³<https://github.com/liyunlongaaa/NSD-MS2S>

⁴<https://pypi.org/project/rir-generator>

⁵Each RIR-classifier takes a recording and returns a probability distribution of its RIRs in this recording. These distributions were averaged over all oracle segments and all MDM channels for each dataset. The more probable RIR is the more relevant it is for the dataset.

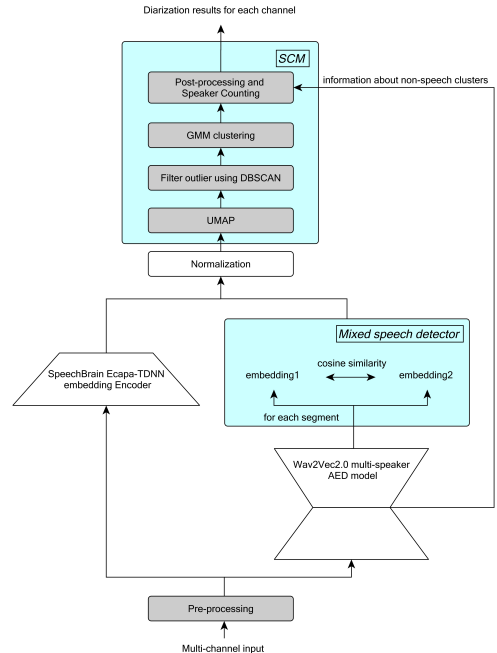


Figure 2: Clustering-based diarization.

with 20 sources and 10 receivers per each room. Using these multichannel RIRs, LibriSpeech data and noises extracted from CHiME8 data, the 10-channel diarization dataset of about 2500 hours was prepared with the help of mixture simulation script from BUT_EEND repository [14]. The statistics of speaker overlaps for the simulation was collected from NOTSO FAR-1 [15].

ND models were trained with both the original code and our re-implementation of it, keeping the original architecture but with minor changes of dataloader and optimizer. Models also differ from each other in the number of epochs for pre-training and finetuning as well as a set of channels selected for training. NSD-MS2S models uses the input segmentation to compute activity masks and internal speaker representations. To train such a model one needs a segmentation along with speaker labels. The best pretrained NSD-MS2S model was trained on a segmentation obtained as a result of applying clustering-based diarization from our CHiME7 pipeline [16] with **oracle** speakers count to the simulated dataset. For the finetuning, several segment boundaries augmentation approaches were tested. One of them uses segments from clustering-based diarization whose speaker labels are mapped to the reference ones according to the best permutation. Another one just truncates each segment by a predefined margin (0.5s) from the both sides (we call this “bounds erosion”), surprisingly, this provides one of the best

Table 2: *Neural diarization results.*

System	Data type	DER							AVG
		chime6		dipco		mixer6		notsofar1	
		dev	eval	dev	eval	dev	eval	dev	
nsd_ft_real_data	wpe	12.2	16.2	14.8	11.3	7.8	4.4	8.3	10.7
nsd_filter_train	wpe	12.4	15.8	15.7	10.9	7.9	5.3	8.9	10.9
nsd_ft_more_real_data	wpe	11.7	15.2	13.3	10.2	7.4	4.4	8.1	10.0
nsd_ft_more_real_data_er	wpe	11.7	15.0	14.2	10.9	7.2	4.4	8.1	10.2
fusion	orig&wpe	10.8	14.8	13.8	10.0	7.1	4.3	7.9	9.8

finetuning results.

2.3. Fusion and postprocessing of the diarization results

Diarization results are fused using the DOVER-Lap [17] tool in several stages. First, different clustering-based diarization pipelines are applied to each selected audio channel separately. They use 2 different Voice Activity Detectors, 2 versions of recordings (original and WPE-dereverberated) and 3 different values for the small cluster rejection parameter, 12 pipelines in total. Clustering-based diarization results are fused over all 12 pipelines in each channel separately. The diarization results obtained with the best single clustering-based diarization pipeline and fusion of all pipelines are presented in Table 1.

Obtained clustering-based diarization results for each channel are fed into the several different neural diarization models along with either original or dereverberated audio and all results are fused again for each channel separately. Finally the diarization results from different channels are fused together.

Fusion described above provides hard labels of speaker activity. But speaker separation may benefit from using soft activities predicted by ND models. Therefore, a soft version of the fusion was implemented as well. For each channel we keep only those ND models whose speaker count predictions match the clustering-based diarization results. Then, the best speaker permutation between each ND model and DOVER-Lap results is found based on activities correlation. Finally all speaker-permuted activities from each selected model are averaged.

Seven ND models were trained and applied for both original and WPE-processed recordings, resulting in 14 different sets of neural diarization results. Seven results corresponding to four different NSD systems were selected for fusion based on DER minimization on the dev sets. DER values for these wpe-based variants and their fusion are presented in Table 2. The first model was trained in the original pipeline and finetuned for 6 epochs on incomplete CHiME8 data (before releasing of 2nd and 3rd parts of NOTSO FAR-1). Other 3 models were trained in our version of NSD-MS2S code. The second model was finetuned on the results of clustering-based diarization as a reference. Since the clustering-based diarization was imperfect, the finetuning was made on a subset of CHiME8 data filtered with 2 criteria: session-channel pairs with $DER > 50\%$ or those where the number of speakers in clustering-based diarization results was less than the oracle one, were rejected. The third and fourth models were finetuned on the complete CHiME8 training data. The only difference between them was using bounds erosion in the fourth model.

3. Speech Separation

3.1. Guided source separation

Our system basically uses Guided Source Separation (GSS) [3] for the extraction of each speaker’s speech based on hard or

soft speech activities obtained from Neural Diarization stage. The GSS module is based on gpu-GSS [18] and is used with mostly the same parameters as in CHiME7 [16]. It includes multichannel WPE dereverberation, training complex Angular-Central GMMs (cACGMMs) for each frequency bands and using obtained source masks for MVDR beamforming.

In the current challenge we used only the version of GSS based on the soft speaker activities. Each segment for GSS processing was taken from the results of the neural diarization and extended by a small margin (0.5s) to both sides. GSS on these extended segments was initialized by soft activities from the ND results. And the best results were obtained when GSS was run for only 5 iterations.

There are sessions where speakers move a lot. To handle these situations a special chunked version of GSS was also used where long segments are processed by chunks of length 300 frames (about 5s). Since each of GSS versions is better on some sessions and worse on others, both of them were used in the subsequent ASR to complement each other in fusion.

One more GSS version was used as well. Speech segments obtained from GSS described above, were recognized and ASR results were used to infer a (0/1) VAD masks. This mask was multiplied by soft activities to suppress noises to which ND had assigned a high activity scores. We use the same mask correction technique for the TSE model described in Section 3.3.

3.2. Continuous source separation

We also made an investigation of using multichannel Continuous Speech Separation (CSS) for speaker separation. We chose 2 different architectures of multichannel CSS models, namely Spatial Net [19] and TF-GridNet [20]. Since there is no clean single-speaker speech is available for CHiME8 data, we had to simulate similar training data. Clean speech recordings were selected from both LibriSpeech and VoxCeleb1,2 datasets. They were reverberated with multichannel RIRs described in subsection 2.2. All training examples had a length of 4s and included 0 to 3 speakers but with no more than 2 speakers overlap, to limit CSS model by 2 output channel. 10 channel mixtures were generated and 6 best channels were selected according to MicRank to use in training. Targets were the clean single speaker parts of mixture convolved with respective RIRs.

Although both models worked well on synthetic data and relatively simple mixed speech chunks, there were multiple channel confusion errors in more complex situations. This makes such an approach impractical for the application in CHiME8 challenge, especially in the scenario where CSS precedes diarization. Therefore we refused to make complete CSS. Nevertheless, application of CSS to the single-speaker segments of the CHiME8 training data resulted in a set of denoised data which was added to ASR model training and provided some improvements of the ASR results.

3.3. Target speaker extraction

We decided to replace CSS with Target Speaker Extraction (TSE). This task is simpler and is more relevant to the task, especially in conversations with highly overlapped speech. Besides, it doesn't require a permutation resolution and thus a model can be trained without PIT. We kept the architecture of models mostly unchanged, just extended it a bit to process a target speaker information along with mixture itself. We tried to use several representations of target speaker information, for example target speaker embeddings and target speaker activity time masks as proposed in [21]. Unfortunately, ASR results on TSE outputs were inferior to those obtained on GSS+beamforming. The best ASR results on TSE outputs were obtained when the time-frequency masks from GSS were used as a target speaker representations. But they were still behind GSS+beamforming pipeline despite the good quality of target speaker's speech in most of the TSE outputs and low SNRs.

We assumed that this gap is due to a mismatch between losses used in TSE (si-sdr) and ASR (CTC+attention). That's why we concatenated the pretrained TSE and ASR models and finetuned TSE one on ASR criterion. The TSE model was based on hard activity masks available from the ND. The ASR model was based on Wavlm [22], see section 4. Since we didn't need TSE targets in this approach but only reference transcripts, this combined TSE-ASR model may be trained on CHiME8 data directly. This is an important merit of such approach.

Table 3: ASR results on CHiME8 devsets

tcpWER, %				
chime6	dipco	mixer6	notsofar1	MacroAvg
Constrained LM track				
22.79	28.96	10.14	19.07	20.24
Unconstrained LM track				
22.47	28.41	9.85	18.72	19.86

To prevent TSE from learning to reproduce the ASR training dataset, a less overfitted ASR model was chosen. The ASR model was trained using frozen wavlm on the reverberated Librispeech and CHiME8 training data. We noticed that GSS is very effective at targeting the speaker, because it uses a large context. In contrast, TSE works with smaller chunks and often misses the speaker. For the fine-tuning, the chunk size of the TSE model was increased to a maximum of 8 seconds. Utterances longer than 8 seconds were split into chunks of 6 seconds without overlapping. However, it is still much less than the GSS context. To improve accuracy of the TSE model, a signal from the GSS was used as a reference channel in the TSE input. As a result, TSE is in fact trained to be a GSS signal enhancer for downstream the ASR task, so we dubbed this model the Guided Target Speaker Extractor (G-TSE).

This additional finetuning significantly improved the performance of the TSE model on real-world data. The G-TSE model is a part of the best pipeline and outperforms GSS in different scenarios. We used outputs of G-TSE part as just an alternative to GSS and pass them to different ASR backends, described below.

4. ASR and post-processing

4.1. Speech recognition

The speech recognition module is similar to that used in our CHiME7 system. It includes several models, both DNN-HMM hybrids trained using Kaldi [23], and a bunch of E2E models

trained with either ESPNet [24] or k2⁶. The ESPNet models have a Uconv-Conformer [25] and an E-Branchformer [26] architecture and use the pretrained WavLM model as a frontend. The frontend is initially frozen but then it is finetuned along with the base Uconv-Conformer model. Embeddings from the trained Uconv-Conformer models are used as features for training ZipFormer [27] model in k2.

According to our previous investigation, we trained our ASR models mainly on GSS-processed CHiME8 train datasets. But we have found that some consistent augmentation may help to slightly improve ASR accuracy. We used the clean part of LibriSpeech (460 hours) distorted with the selected RIRs (see above) and noises extracted from CHiME8 data. Another source of augmentation was the single-speaker segments of training data processed with multichannel CSS to remove environmental noise (see Section 3).

4.2. Adaptation

To improve ASR results we also tried the unsupervised adaptation approach described in [28]. The adaptation is applied on per-session level and the adapted model can provide better results compared to unadapted one. Unfortunately we didn't have enough time to try the whole adaptation pipeline but have implemented the described selection of segments for adaptation. The standard finetuning of ESPNet-based models on the selected segments brought some improvements of the ASR results on the adaptation session.

4.3. Rescoring

Each trained ASR models was applied to each variants of Speaker Separation outputs (GSS, G-TSE). For each such combination ASR results obtained as NBest lists were re-scored with the large language model. This model was based on the Llama-2-7B⁷ LLM that was finetuned on all transcripts of CHiME8 training data augmented with LibriSpeech texts. For the finetuning, all utterances from each sessions were supplemented with BOS/EOS tokens and concatenated together in the order of ascending beginning time. During inference a context of 512 tokens composed from 1-best results of previous utterances recognition was used. This is very similar to the approach proposed in [29]. The rescoring results are used in the Unconstrained LM track only.

4.4. Fusion

All ASR results obtained on devsets were processed to select the best subset to be fused together. All selected Nbest results were converted to Kaldi lattice format and lattice fusion along with MBR decoding were applied to obtain the final ASR results.

The tcp-WER values and macro-average over all devsets for each selected systems and the fusion results are shown in Table 3. The set of systems selected for fusion based on devsets was used to obtain the final results on eval sets.

5. Acknowledgements

This research is financially supported by the Foundation for National Technology Initiative's Projects Support as a part of the roadmap implementation for the development of the high-tech field of Artificial Intelligence for the period up to 2030 (agreement 70-2021-00187).

⁶<https://github.com/k2-fsa>

⁷<https://huggingface.co/meta-llama/Llama-2-7b-hf>

6. References

- [1] S. Watanabe, M. Mandel, J. Barker *et al.*, “Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings,” *arXiv:2004.09249*, 2020.
- [2] S. Cornell, M. Wiesner, S. Watanabe, D. Raj, X. Chang, P. Garcia, M. Maciejewski, Y. Masuyama, Z.-Q. Wang, S. Squartini, and S. Khudanpur, “The chime-7 dasr challenge: Distant meeting transcription with multiple devices in diverse scenarios,” *arXiv:2306.13734*, 2023.
- [3] C. Boeddeker, J. Heitkaemper, J. Schmalenstroer *et al.*, “Front-end processing for the CHiME-5 dinner party scenario,” in *CHiME Workshop*, 2018, pp. 35–40.
- [4] K. Zmolikova, M. Delcroix, T. Ochiai, K. Kinoshita, J. Černocký, and D. Yu, “Neural target speech extraction: An overview,” *arXiv:2301.13341*, 2023.
- [5] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, “NARA-WPE: A python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing,” in *ITG*, 2018, pp. 1–5.
- [6] M. Wolf and C. Nadeu, “Channel selection measures for multi-microphone speech recognition,” *Speech Communication*, vol. 57, p. 170–180, 02 2014.
- [7] S. Novoselov, G. Lavrentyeva, A. Avdeeva, V. Volokhov, N. Khmelev, A. Akulov, and P. Leonteva, “On the robustness of wav2vec 2.0 based speaker recognition systems,” in *Proc. INTERSPEECH 2023*, 2023, pp. 3177–3181.
- [8] N. Dawalatabad, M. Ravanelli, F. Grondin, J. Thienpondt, B. Desplanques, and H. Na, “Ecapa-tdnn embeddings for speaker diarization,” in *Interspeech*, 2021.
- [9] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” *arXiv:1802.03426*, 2018.
- [10] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, “Unsupervised methods for speaker diarization: An integrated and iterative approach,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2015–2028, 2013.
- [11] S. Cornell, T. Park, S. Huang, C. Boeddeker, X. Chang, M. Maciejewski, M. Wiesner, P. Garcia, and S. Watanabe, “The chime-8 dasr challenge for generalizable and array agnostic distant automatic speech recognition and diarization,” *arXiv preprint arXiv:2407.16447*, 2024.
- [12] Y. Khokhlov, T. Prisyach, A. Mitrofanov, D. Dutov, I. Agafonov, T. Timofeeva, A. Romanenko, and M. Korenevsky, “Classification of room impulse responses and its application for channel verification and diarization,” in *INTERSPEECH*, 2024, p. to appear.
- [13] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An asr corpus based on public domain audio books,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.
- [14] F. Landini, A. Lozano-Diez, M. Diez, and L. Burget, “From Simulated Mixtures to Simulated Conversations as Training Data for End-to-End Neural Diarization,” in *Proc. Interspeech 2022*, 2022, pp. 5095–5099.
- [15] 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 *Interspeech 2024*, 2024, pp. 5003–5007.
- [16] T. Prisyach, Y. Khokhlov, M. Korenevsky, A. Mitrofanov, T. Timofeeva, I. Odegov, R. Nasretidinov, I. Lezhenin, D. Miroschnichenko, A. Karelina, M. Mitrofanova, R. Svechnikov, S. Novoselov, and A. Romanenko, “STCON System for the CHiME-7 Challenge,” in *Proc. 7th International Workshop on Speech Processing in Everyday Environments (CHiME 2023)*, 2023, pp. 87–92.
- [17] D. Raj, L. P. Garcia-Perera, Z. Huang, S. Watanabe, D. Povey, A. Stolcke, and S. Khudanpur, “DOVER-Lap: A method for combining overlap-aware diarization outputs,” *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021.
- [18] D. Raj, D. Povey, and S. Khudanpur, “Gpu-accelerated guided source separation for meeting transcription,” *arXiv:2212.05271*, 2022.
- [19] C. Quan and X. Li, “Spatialnet: Extensively learning spatial information for multichannel joint speech separation, denoising and dereverberation,” *arXiv:2307.16516*, 2023.
- [20] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, “Tf-gridnet: Integrating full- and sub-band modeling for speech separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. PP, pp. 1–15, 01 2023.
- [21] M. Delcroix, K. Zmolikova, T. Ochiai, K. Kinoshita, and T. Nakatani, “Speaker activity driven neural speech extraction,” *arXiv:2101.05516*, 2021.
- [22] 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, J. Wu, M. Zeng, X. Yu, and F. Wei, “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *arXiv:2110.13900*, 2021.
- [23] D. Povey, A. Ghoshal, G. Boulianne *et al.*, “The Kaldi speech recognition toolkit,” in *IEEE ASRU Workshop*, 2011.
- [24] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique, Y. Soplín, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, “Espnet: End-to-end speech processing toolkit,” *arXiv:1804.00015*, 2018.
- [25] A. Andrusenko, R. Nasretidinov, and A. Romanenko, “Uconv-conformer: High reduction of input sequence length for end-to-end speech recognition,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [26] K. Kim, Fel, a. Wu, Y. Peng, J. Pan, P. Sridhar, K. J. Han, and S. Watanabe, “E-branchformer: Branchformer with enhanced merging for speech recognition,” *arXiv:2210.00077*, 2022.
- [27] Z. Yao, L. Guo, X. Yang, W. Kang, F. Kuang, Y. Yang, Z. Jin, L. Lin, and D. Povey, “Zipformer: A faster and better encoder for automatic speech recognition,” *arXiv:2310.11230*, 2023.
- [28] Y. Hu, C. Chen, C.-H. H. Yang, C. Qin, P.-Y. Chen, E. S. Chng, and C. Zhang, “Self-taught recognizer: Toward unsupervised adaptation for speech foundation models,” *arXiv:2405.14161*, 2024.
- [29] A. Ogawa, N. Kamo, K. Matsuura, T. Ashihara, T. Moriya, T. Kano, N. Tawara, and M. Delcroix, “Applying llms for rescoring n-best asr hypotheses of casual conversations: Effects of domain adaptation and context carry-over,” *arXiv:2406.18972*, 2024.