Bring the Noise: Introducing Noise Robustness to Pretrained Automatic Speech Recognition

Patrick Eickhoff[1(✉)], Matthias Möller[2], Theresa Pekarek Rosin[1], Johannes Twiefel[1,3], and Stefan Wermter[1]

[1] Knowledge Technology, Department of Informatics, University of Hamburg, Vogt-Koelln-Str. 30, 22527 Hamburg, Germany
{patrick.eickhoff,theresa.pekarek-rosin, johannes.twiefel,stefan.wermter}@uni-hamburg.de

[2] Centre for Applied Autonomous Sensor Systems (AASS), Örebro University, Örebro, Sweden
matthias.moeller@oru.se

[3] exXxa GmbH, Vogt-Koelln-Str. 30, 22527 Hamburg, Germany
https://www.knowledge-technology.info, https://exXxa.ai/

Abstract. In recent research, in the domain of speech processing, large End-to-End (E2E) systems for Automatic Speech Recognition (ASR) have reported state-of-the-art performance on various benchmarks. These systems intrinsically learn how to handle and remove noise conditions from speech. Previous research has shown, that it is possible to extract the denoising capabilities of these models into a preprocessor network, which can be used as a frontend for downstream ASR models. However, the proposed methods were limited to specific fully convolutional architectures. In this work, we propose a novel method to extract the denoising capabilities, that can be applied to any encoder-decoder architecture. We propose the Cleancoder preprocessor architecture that extracts hidden activations from the Conformer ASR model and feeds them to a decoder to predict denoised spectrograms. We train our preprocessor on the Noisy Speech Database (NSD) to reconstruct denoised spectrograms from noisy inputs. Then, we evaluate our model as a frontend to a pretrained Conformer ASR model as well as a frontend to train smaller Conformer ASR models from scratch. We show that the Cleancoder is able to filter noise from speech and that it improves the total Word Error Rate (WER) of the downstream model in noisy conditions for both applications.

Keywords: Conformer · Noise Robustness · Speech Recognition

1 Introduction

End-to-End (E2E) systems have been the state-of-the-art approach to Automatic Speech Recognition (ASR) for a few years now [1,4,17]. An E2E system usually takes audio as input, processes it into an internal representation, and

L. Iliadis et al. (Eds.): ICANN 2023, LNCS 14260, pp. 376–388, 2023.
https://doi.org/10.1007/978-3-031-44195-0_31

produces a transcript of the speech. The big advantage of these systems is that all components are trained together, so they can learn a joint representation. However, the disadvantage is that they often require deep models with a large number of parameters to perform well. For example, the recent Whisper-large model [17] contains about 1550 M parameters. Training a model like this from scratch is computationally expensive and usually not possible for research institutions. However, most of these models can be successfully adapted to smaller domains through the use of transfer learning, which indicates the quality of speech representations learned [9,17]. Additionally, E2E systems usually do not have preprocessing applied to their input and the model itself has to learn how to separate speech from noise [17]. Usually, the earlier layers of recent ASR architectures are required to separate noise from speech implicitly. When a new ASR architecture is developed, the earlier noise-handling layers need to be trained again. This raises the question if it is possible to separate the processing capabilities of such a large and powerful pretrained ASR model and reuse them for another model.

Our work takes inspiration from the recent findings of Möller et al. [13]. They were able to utilize a pre-trained Jasper [12] ASR model to create a preprocessor, which increases the noise robustness of pretrained models and improves the performance of smaller ASR models trained from scratch. However, their approach has two disadvantages: 1) their approach is only applicable to a specific architecture of Jasper, and 2) Jasper is no longer state-of-the-art for English ASR. Instead, attention-based models derived from the Transformer [23] architecture have outperformed convolutional and recurrent ASR approaches [3,7,9,17]. Therefore, we propose a new extraction method (Parallel Weighted Sum), which is potentially applicable to any encoder-decoder ASR architecture. We apply this method to a Conformer [7] model, a state-of-the-art attention-based architecture, to create our preprocessor called Cleancoder. Our model can function either as an independent frontend for pre-trained ASR models or can be used in combination with architectures trained from scratch to improve their noise robustness. In our experiments, we measure the performance (Word Error Rate; WER) on different noise levels on the Noisy Speech Dataset (NSD) [20] to show that our methods improve the performance under noisy conditions and the performance does not decrease under clean conditions (LibriSpeech [16]) when using our preprocessor.

2 Related Work

To improve the noise robustness of a speech recognition model, training processes usually include adding both artificial and realistic noise to the training data. This leads to large-scale ASR models showcasing certain noise robustness without any further preprocessing steps. However, since smaller models might not be able to perform the same internal denoising steps, it is important to examine how the capabilities of larger models can be exploited by smaller models. When mentioning ‘small’, ‘medium’, and ‘large’ ASR models we refer to the number

of parameters defined by Gulati et al. [7] for their Conformer configurations (∼10M, ∼30M, ∼100M).

There are different approaches to creating external preprocessors, that denoise speech for further speech recognition. Many of these focus on filtering noise from speech using statistical methods in combination with deep learning methods [2,6,8]. A recent example of such a method is the Cleanformer model. Cleanformer [2] is a multichannel frontend architecture for speech enhancement based on the Conformer [7] architecture. The model combines raw noisy speech and enhanced input features, produced by a SpeechCleaner [10] noise cancellation algorithm, to create an Ideal Ratio Mask (IRM) [14]. This mask in the spectral space, estimates the ratio of speech in the noisy signal. These ratios are then applied to the input signal to filter out noise. The model works independently of the combined ASR model and can reduce WERs across multiple SNR values by approximately 50%.

Instead of applying a filtering method to the noisy signal, our approach reconstructs clean spectrograms completely from latent representations. Our work is based on the findings of Möller et al. [13], who created a frontend architecture for noise filtering based on the Jasper [12] architecture. Based on the findings of Li et al. [11] and their probing methods to extract and predict spectrograms from hidden representations, Möller et al. applied this method to gauge the denoising capabilities of a pre-trained Jasper model. The underlying assumption is that areas of speech that the model perceives as noise are filtered out very early by the system and are not represented in the model's latent space. Thereby, those filtering capabilities could be leveraged for other ASR models and increase their noise robustness.

Möller et al. demonstrate how the reconstructed representations of speech support other already pretrained ASR systems in noisy conditions. Additionally, they observe that those features support other ASR systems as input while training and that models with those representations generally perform better on noisy and clean data. However, their approach relies strongly on the architecture of Jasper [12] and its residual connections. They retrain the batch normalization layers in the model and are therefore limited to one specific architecture, which is not state-of-the-art anymore. Our work introduces a way that could potentially reconstruct speech from any ASR system while still retaining denoising capabilities.

3 Methodology

Our method of constructing a denoising preprocessor from a pretrained ASR model is inspired by the work of Möller et al. [13]. However, we propose an architecture that can extract latent representation from potentially any encoder-decoder ASR architectures and is not limited to the Jasper architecture. Our Cleancoder model extracts the latent representations of an ASR model's encoder and reconstructs denoised spectrograms.

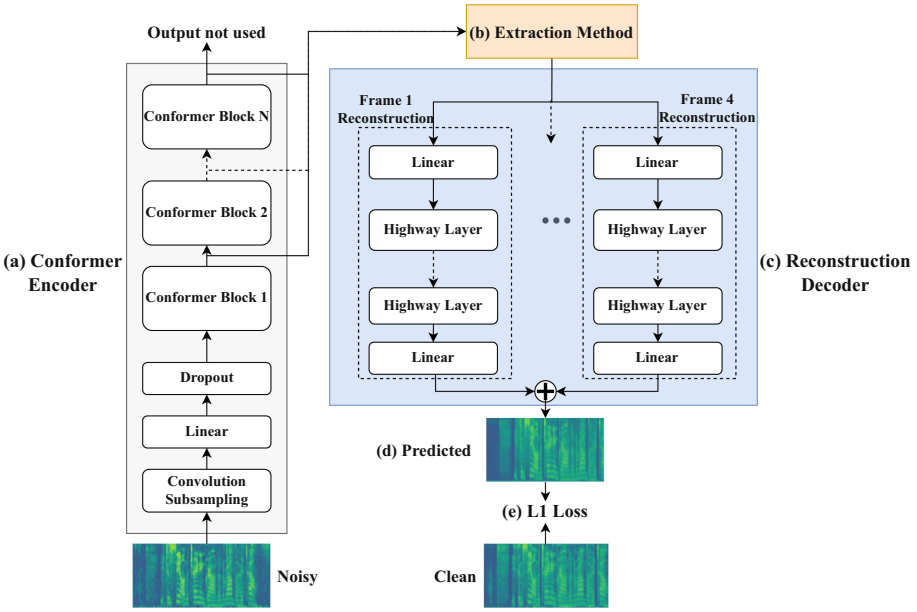


Fig. 1. This figure shows the architecture of our Cleancoder. On the left, we display the original Conformer encoder architecture (a). Then, the output of every Conformer block is fed into our extraction method (b). This method computes a weighted sum of the latent representation and feeds this vector to our reconstruction decoder (c). This decoder contains four different Highway Networks tasked with reconstructing one-fourth of the final output frame. The subsampling layer of the Conformer reduces the temporal dimension of the input by a factor of four. Thus we generate four outputs, which are appended along the temporal axis to reconstruct a complete spectrogram of a frame (d). Then we compute the L1 loss (e) between the reconstructed spectrogram and the clean ground truth.

The architecture follows an encoder-decoder structure, shown in Fig. 1. We choose pretrained Conformer models as a baseline to extract our preprocessor from it. These models are larger than the Jasper [12] used by Möller et al. [13] but still can be trained from scratch in a reasonable time. There are multiple pretrained Conformer models with Connectionist Temporal Classification (CTC) available from NVIDIA NeMo[1]. Jasper has been outperformed by many more recent ASR models, which are using attention-based architectures [4,5,7]. The Jasper model used by Möller et al. only reported a WER of 2.84%(test-clean)/7.84%(test-other) [12] on the LibriSpeech [16] test set, using an external language model. The large Conformer we base our model on reported 1.9%(test-clean)/3.9%(test-other) [7]. Thus, we assume that our Cleancoder will be able to yield better WER improvements for downstream models.

[1] https://catalog.ngc.nvidia.com/models.

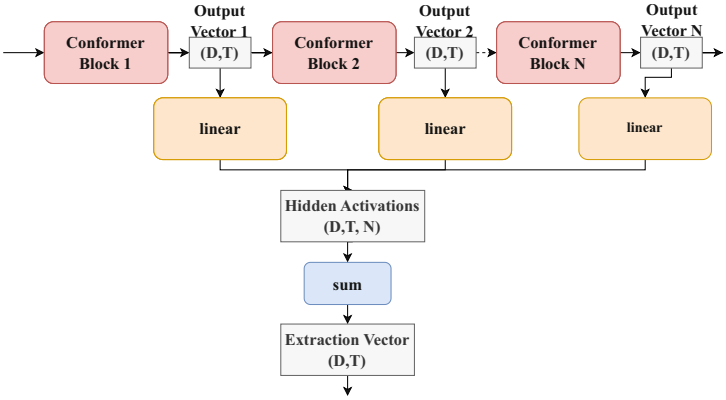


Fig. 2. This figure shows our Parallel Weighted Sum method. Each of the N latent vectors of the Conformer encoder has size (D, T) and is fed through a separate fully connected layer after which we sum all the projected vectors into one output vector of size (D, T).

We reuse the encoder of the ASR Conformer model (see Fig. 1, a). We disregard the decoder layers as we're only interested in the latent representations. To extract these hidden activations, we feed the output of each Conformer block into our extraction method. We create an approach that applies to potentially any encoder-decoder ASR architecture while remaining simple. We propose a Parallel Weighted Sum extraction method (see Fig. 1, b and Fig. 2), which extends the regular weighted sum. However, instead of choosing one weight vector to reduce all the hidden activations into one, our method feeds each layer through separate parallel projection layers and computes the sum across these layers. This way, we not only weigh the contribution of the different blocks to the denoised output but also weigh the information contained in each feature vector. We took inspiration from the work of Yang et al. [25], who compared different Self-Supervised Learned (SSL) representations.

For our decoder network, we choose to follow the example of Möller et al. [13] and use four-layer Highway Networks [18]. They have shown, that these networks can reconstruct spectrograms sufficiently for ASR from hidden representations. Since the Conformer preprocessing block reduces the temporal dimension by a factor of four, we train four different Highway Networks. The four outputs are appended along the temporal axis. Given the input x consists of t frames, the Conformer will reduce the temporal dimension by four yielding $t/4$ frames. We denote the latent representation constructed by our Parallel Weighted Sum as s_i for frame i and our four Highway Networks as $N_1, N_2, \ldots, N_4$. Since our decoder is almost identical to the one of Möller et al. [13], we obtain a similar equation for the output y of our model:

$$y = (N_1(s_0), N_2(s_0), N_3(s_0), N_4(s_0), \ldots, N_1(s_{t/4}), N_2(s_{t/4}), N_3(s_{t/4}), N_4(s_{t/4}))$$

4 Experimental Results

4.1 Datasets

Noisy Speech Database. The Noisy Speech Database (NSD) [20] was designed to test and train speech enhancement algorithms. It contains pairs of noisy and clean speech, sampled at 48 kHz, and is divided into a training and test set. For our experiments, we downsampled our input to 16 kHz. Each sample in the datasets provides noisy and clean audio, a transcript, information about the speaker, signal-to-noise ratio (SNR), and noise type. There are two sets of the NSD with 28 [22] and 56 speakers [21] taken from the Voice Bank Corpus [24]. The noisy samples were created by adding recorded noise from the DEMAND database [19] as well as generated babble and speech-shaped noise. These noise types were applied at different SNRs. We combine the 28 and 56-speaker sets to expose our model to a larger variety of speakers and noise conditions. Thus, we will ensure better generalization. We use the NSD to train our denoising preprocessor and to evaluate the performance of downstream ASR models on noisy data.

LibriSpeech. LibriSpeech [16] is a corpus of approximately 1000 h of clean English speech, sampled at 16 kHz. LibriSpeech is an established dataset for evaluating ASR models [7,12]. We use LibriSpeech in our experiments to train small ASR models from scratch.

4.2 Training the Cleancoder

To evaluate if the Cleancoder architecture filters the noise from speech we train two preprocessor models (medium, and large) on the NSD train set. This way, we can estimate the required size of the best preprocessor. Our preprocessors are trained to reconstruct spectrograms of the same form as the encoder's input. These are log-Mel spectrograms with 80 features, a window size of 0.025, and a windows stride of 0.01. We convert each clean and noisy audio signal of each sample in the NSD trainset into log-Mel spectrograms. While training, our models are fed the noisy spectrograms and predict denoised spectrograms.

Then, we can compute the L1 loss between the clean and denoised spectrograms. We train two different models with the medium and large-sized Conformer CTC models. The medium Conformer consists of ∼30.7M parameters, while the large one consists of ∼118.8M parameters [7]. For each encoder model, we train our preprocessor for 100 epochs on a batch size of 64 with L1 Loss. The learning rate is set to a magnitude of $1e^{-3}$, where the precise values are taken from a hyperparameter search, which we conduct before the actual training. The search was conducted on the NSD trainset. The Adam optimizer is configured with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and a weight decay of $1e^{-4}$. The learning rates are set to the optimal values from our hyperparameter search. We choose to omit the learning rate scheduler since the initial learning rate is already very small. Our decoder is configured as four four-layer Highway Networks.

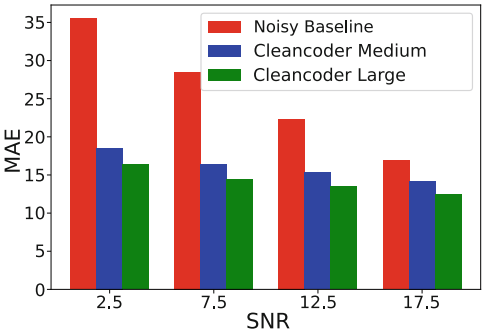


Fig. 3. This figure presents the mean absolute error (MAE) computed between the noisy and clean and respective denoised and clean spectrograms of the NSD test set grouped by the signal-to-noise ratio (SNR). We observe that the denoised spectrograms of both preprocessors show a lower MAE than the noisy baseline across all noise conditions. The large Cleancoder shows the lowest MAE.

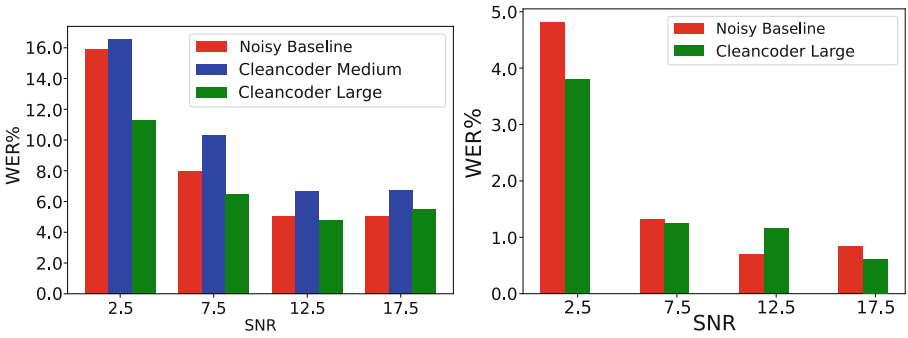
After training the two models, we inspect the differences between the noisy, clean, and denoised spectrograms. We measure their deviation by computing the mean absolute error (MAE) between the clean and noisy as well as clean and denoised spectrograms. Our results on the MAE are shown in Fig. 3. For both preprocessors, we observe that they reduce the MAE compared to just the noisy input. The lower the SNR the larger the improvement, indicating that the Cleancoder models filter noise from speech. However, the MAE of the Cleancoders remains at similar values for all SNRs, which could suggest that the MAE reduction is already saturated at a low SNR.

4.3 Frontend for Pretrained Models

Next, we test how our Cleancoder affects the performance of existing pretrained ASR models. Therefore, we use our preprocessors as frontends to first denoise the input signal and generate spectrograms. These are fed into a pretrained downstream ASR model which predicts transcriptions. Finally, we measure the WER between the ground truth texts, the transcripts recognized from the unprocessed noisy spectrograms (noisy baseline), and the transcripts recognized from the preprocessed noisy spectrograms (our preprocessor).

For our experiments, we choose a medium-sized Conformer with CTC and a large Conformer Transducer as downstream ASR models, which are both publicly available through NVIDIA's model collection. We chose two different ASR models to ensure a degree of invariance to the downstream architecture. This experiment verifies if it is possible to combine our front end with other downstream architectures without the risk of degrading the performance.

Our results are shown in Fig. 4. We can see, that overall, while the WER increases with the medium preprocessor compared to the baseline, it decreases for almost all SNR configurations with the large Cleancoder. The large Cleancoder



(a) Conformer CTC ASR model (b) Conformer Transducer ASR model

Fig. 4. Figure (a) presents the WER on the NSD test set for evaluating our frontends using a Conformer CTC downstream model. Figure (b) presents the WER using a Conformer Transducer downstream model. The Transducer was only evaluated with the large Cleancoder, as the medium Cleancoder had already proven to be unable to improve ASR performance. Both plots show the WER grouped by SNR. Each bar denotes either the noisy spectrograms or respective denoised spectrograms. The WER improves the most on low SNR samples and slightly degrades WER on high SNR samples.

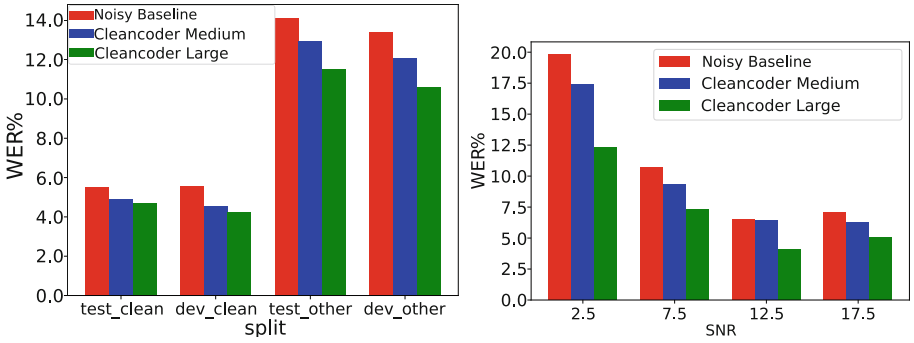
performs better on samples with low SNR, only for samples with the highest SNR of 17.5 the performance is slightly worse than the noisy baseline. We observe that the performance of the baseline Conformer Transducer got worse from SNR 12.5 to 17.5. When analyzing the errors we found minor anomalies in the predictions, however, since the WER is already very low we accredit this observation to general variance.

We further discuss the correlation between our MAE and WER results. Möller et al. [13] suggested, that the MAE and WER do not necessarily correlate. We found little research on the impact of the MAE between noisy and clean speech on the resulting WER. While we observe significant improvements of the MAE using our medium and large preprocessors, the WER shows significantly lower performance on the medium preprocessor. Only the large version yields positive results. There seems to be no strong correlation between the MAE and WER. We assume that a different loss function to train the preprocessor would be more appropriate, and we will examine this in our future work.

4.4 Training an ASR Model from Scratch

We evaluate how the Cleancoder impacts the training of a smaller downstream ASR model from scratch. The architecture of choice for the ASR model was a small Conformer using CTC without a language model. We train three different small Conformer models. All three are trained on LibriSpeech's training splits. The baseline model uses no front end, while the others are trained on the outputs of our medium and large preprocessor, respectively. All three are trained for 100

epochs with CTC loss using a batch size of 128 and an Adam optimizer with β_1 as 0.9 and β_2 as 0.98. We apply a NoamAnnealing learning rate scheduler with 10,000 warmup steps, an initial learning rate of 2.0, and a minimal learning rate of $1.0e^{-06}$.



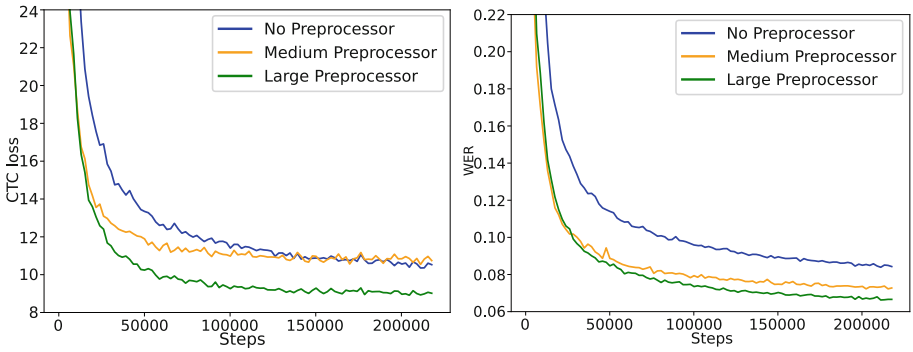
(a) LibriSpeech (b) NSD

Fig. 5. Figure (a) shows the WER (%) of our three ASR models trained from scratch for the dev- and test-splits of LibriSpeech. Figure (b) shows the WER (%) for the NSD test set grouped by SNR. Each model used the same small Conformer architecture and was trained on either the raw input or the output of our medium and large preprocessors. We observe that the ASR model using the large preprocessor shows the lowest WER in both figures.

In the first plot of Fig. 5, we report the WER computed on the test-clean, test-other, dev-clean, and dev-other splits of LibriSpeech. We observe that both models using our preprocessors outperform the baseline ASR model. The large Cleancoder shows the lowest mean WER on all splits.

Furthermore, we show the WER of our ASR models on the NSD test set grouped by SNR in the second plot of Fig. 5. We observe that both models using our preprocessors outperform the baseline model. The large preprocessor shows the best performance with an almost 4% improvement overall. We also observe that using the Cleancoders yields the biggest improvements on samples with a low SNR. This shows the models are more robust to noise.

Finally, we investigate the impact of using our Cleancoders over the course of training. The evaluation CTC loss and evaluation WER are shown in Fig. 6. We observe that both models using our front end converge faster and reach a lower loss and WER than the baseline model. The loss curves and WER curves follow an almost identical course. We observe that the validation loss and WER are both lowest for the large Cleancoder. As previously discussed, this supports the assumption that the large Cleancoder generalizes better to different noise conditions.



(a) Validation CTC Loss (b) Validation WER

Fig. 6. Plot (a) shows the CTC loss over the number of training steps for the validation dataset. Plot (b) shows the WER, computed on the validation dataset, over the training steps. Each plot shows three curves for the baseline ASR model (blue), the medium preprocessor (orange), and the large preprocessor (green). Both preprocessors converge lower than the baseline ASR model, except for the CTC validation loss of the medium preprocessor. (Color figure online)

5 Conclusion

We created preprocessors from pretrained Conformer [7] ASR models by extracting the hidden activations and training a decoder to predict denoised spectrograms. In our experiments, we showed that our Cleancoder improves the performance (WER) under noisy conditions (SNR: 2.5 and 7.5) for two different downstream ASR models. Under clean audio conditions (SNR: 12.5 and 17.5) the performance stayed mostly stable (with one outlier). The results indicate that our preprocessor is capable of improving the performance of downstream ASR models under noisy conditions without the necessity of performing any training for the downstream ASR models. In the second experiment, we trained the downstream ASR model from scratch by first feeding the audio training data through the Cleancoder and then using the generated spectrograms as training data for our downstream ASR model. The performance substantially improved under both noisy and clean audio conditions. Comparing the results of the first and second experiments suggests, that reconstruction errors of our Cleancoder might disturb a pretrained ASR model, but can be compensated by training on these errors. Furthermore, we measured the training and validation loss while performing the training. These results show that the training time of an ASR model can be reduced due to an improved convergence, and the performance can be increased when using our preprocessor as input to a downstream ASR model.

In future work, we plan to research different loss functions aside from MAE to train the Cleancoder for denoising. One example could be the Ideal Ratio Mask (IRM) [15], which has recently been successfully utilized for a denoising frontend [2]. A loss function with better correlation to the downstream WER

could further improve our Cleancoder's performance as an ASR frontend. Furthermore, we will evaluate our preprocessor using additional downstream ASR architectures. Especially the combination of our preprocessor and the recent Whisper [17] model would be worth investigating. Finally, applying our approach to create denoising preprocessors from other architectures will confirm if our method works with any encoder-decoder ASR architecture.

Acknowledgements. The authors gratefully acknowledge support from the German BMWK (SIDIMO), the DFG (CML, LeCAREbot), and the European Commission (TRAIL, TERAIS).

References

1. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: Wav2vec 2.0: a framework for self-supervised learning of speech representations. Adv. Neural Inf. Process. Syst. **33**, 12449–12460 (2020)
2. Caroselli, J., Naranayan, A., O'Malley, T.: Cleanformer: a microphone array configuration-invariant, streaming, multichannel neural enhancement frontend for ASR. ArXiv abs/2204.11933 (2022)
3. Chen, S., et al.: WavLM: large-scale self-supervised pre-training for full stack speech processing. IEEE J. Sel. Top. Signal Process. **16**(6), 1505–1518 (2022)
4. Chung, Y.A., et al.: w2v-BERT: combining contrastive learning and masked language modeling for self-supervised speech pre-training. In: IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 244–250 (2021)
5. Chung, Y.A., et al.: W2v-BERT: combining contrastive learning and masked language modeling for self-supervised speech pre-training. In: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 244–250. IEEE (2021)
6. Fang, H., Wittmer, N., Twiefel, J., Wermter, S., Gerkmann, T.: Partially adaptive multichannel joint reduction of ego-noise and environmental noise. arXiv preprint arXiv:2303.15042 (2023)
7. Gulati, A., et al.: Conformer: convolution-augmented transformer for speech recognition. In: Proceedings of the Interspeech, pp. 5036–5040, October 2020
8. Heymann, J., Drude, L., Haeb-Umbach, R.: Neural network based spectral mask estimation for acoustic beamforming. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),. pp. 196–200 (2016)
9. Hsu, W.N., Bolte, B., Tsai, Y.H.H., Lakhotia, K., Salakhutdinov, R., Mohamed, A.: Hubert: self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 3451–3460 (2021)
10. Huang, Y.A., Shabestary, T.Z., Gruenstein, A.: Hotword cleaner: dual-microphone adaptive noise cancellation with deferred filter coefficients for robust keyword spotting. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6346–6350 (2019)
11. Li, C., Yuan, P., Lee, H.: What does a network layer hear? Analyzing hidden representations of end-to-end ASR through speech synthesis. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6434–6438. IEEE Press, May 2020

12. Li, J., et al.: Jasper: an end-to-end convolutional neural acoustic model. In: Proceedings of the Interspeech, pp. 71–75, September 2019
13. Möller, M., Twiefel, J., Weber, C., Wermter, S.: Controlling the noise robustness of end-to-end automatic speech recognition systems. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN), July 2021
14. Narayanan, A., Wang, D.: Ideal ratio mask estimation using deep neural networks for robust speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7092–7096 (2013)
15. Narayanan, A., Wang, D.: Ideal ratio mask estimation using deep neural networks for robust speech recognition. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7092–7096. IEEE Press (2013)
16. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: LibriSpeech: an ASR corpus based on public domain audio books. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5206–5210. IEEE Press, April 2015
17. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. arXiv preprint arXiv:2212.04356 (2022)
18. Srivastava, R.K., Greff, K., Schmidhuber, J.: Highway networks. arXiv preprint arXiv:1505.00387 (2015)
19. Thiemann, J., Ito, N., Vincent, E.: DEMAND: a collection of multi-channel recordings of acoustic noise in diverse environments (2013)
20. Valentini-Botinhao, C.: Noisy Speech Database for Training Speech Enhancement Algorithms and TTS Models (2017)
21. Valentini-Botinhao, C., Wang, X., Takaki, S., Yamagishi, J.: Investigating RNN-based speech enhancement methods for noise-robust Text-to-Speech. In: Speech Synthesis Workshop (SSW), pp. 146–152, September 2016
22. Valentini-Botinhao, C., Yamagishi, J.: Speech enhancement of noisy and reverberant speech for text-to-speech. IEEE/ACM Trans. Audio Speech Lang. Process. **26**(8), 1420–1433 (2018)
23. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc., December 2017
24. Veaux, C., Yamagishi, J., King, S.: The voice bank corpus: design, collection and data analysis of a large regional accent speech database. In: Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), International Conference. Institute of Electrical and Electronics Engineers (IEEE), United States, November 2013
25. Yang, S.W., et al.: SUPERB: speech processing universal performance benchmark. In: Proceedings of the Interspeech, pp. 1194–1198 (2021)