AN EXPERIMENTAL COMPARISON OF NOISE ROBUST TEXT-TO-SPEECH SYNTHESIS SYSTEMS BASED ON SELF-SUPERVISED REPRESENTATION

Xiaoying Zhao[1,2]*, Qiushi Zhu*[1]*, Yuchen Hu*[3]*, Jie Zhang*[2]

[1]University of Science and Technology of China (USTC), Hefei, China
[2]NERC-SLIP, University of Science and Technology of China (USTC), Hefei, China
[3]Nanyang Technological University, Singapore

ABSTRACT

With the advancements in deep learning, text-to-speech (TTS) techniques utilizing clean speech have witnessed significant performance improvements. The data collected from real scenes often contain noise and generally needs to be denoised by speech enhancement models. TTS models trained on enhanced speech suffer from speech distortion and background noise, which thus affect the quality of synthesized speech. On the other hand, self-supervised pre-trained models have shown excellent noise robustness in various speech tasks, indicating that the learned representation is more tolerant to noise perturbations. Our previous work has demonstrated the superior noise robustness of WavLM representations for speech synthesis. However, the impact of different self-supervised representations on speech synthesis performance remains unknown. In this paper, we systematically compare the performance of four self-supervised representations, WavLM, Wav2vec2.0, HuBERT, and data2vec, using a HiFi-GAN-based representation-to-waveform vocoder and a Fastspeech-based text-to-representation acoustic model. Second, on the basis of our discovery that the representations have better noise and speaker information suppression, we further integrate speaker embedding to realize voice conversion tasks. Finally, experimental results on the LJSpeech and LibriTTS datasets demonstrate the effectiveness of the method. Audio samples are available at: `https://zzftts.github.io/`.

Index Terms— Noise robust text-to-speech, speech synthesis, self-supervised representation, voice conversion.

1. INTRODUCTION

Text-to-speech (TTS) [1–3] aims to synthesize natural and intelligible speech from text. Thanks to advanced deep learning techniques, neural network-based TTS models have demonstrated the ability to synthesize high-quality speech when trained with clean speech data. However, collecting clean speech data requires quiet environments and high-quality recording equipment, e.g., professional audio studios, resulting in high data collection costs. Meanwhile, noisy speech data is very easy to collect and is available in large amounts. If these noisy data can be effectively used for building TTS models, the cost of data collection will be largely reduced and the trained TTS model becomes more applicable. How to train TTS models using noisy data is therefore the focus of this work.

There are numerous approaches dedicated to training TTS models using noisy speech, where the majority uses speech enhancement models for denoising and then train TTS models using the enhanced speech. For example, pre-trained speech enhancement models were used in [4–6] for denoising, followed by training TTS models with the enhanced speech. In [4, 5], recurrent neural networks (RNNs) were trained using parallel noisy and clean speech and Hidden Markov Model (HMM) based acoustic models were then trained with the enhanced features. Although this scheme performs well in simple noise situations, the enhanced speech is susceptible to speech distortion and unseen noise, which can harm the training of the TTS model. To avoid the use of speech enhancement models, it was suggested in [7–12] to train TTS models directly using noisy data. In [7] an end-to-end TTS model was proposed by using speaker embedding and noise representation as conditional inputs to model speaker and noise information separately. In [8, 9], the noise representation was taken as input and the background noise was then removed from the speech via representation decoupling. As in [8, 9] the noise embeddings are sentence-level vectors with coarse granularity, which might not be suitable for complex noise scenarios, DenoiSpeech was proposed in [10], which considers fine-grained frame-level noise modeling to handle real-world noisy speech. DRSpeech was proposed in [11], which jointly represents time-variant additive noises with a frame-level encoder and an utterance-level encoder. Although these methods demonstrate a good noise-robust performance, most exploit mel-spectrogram features.

Self-supervised pre-trained models have shown excellent performance and strong noise robustness on many speech tasks. In the field of automatic speech recognition (ASR), the self-supervised pre-trained models Wav2vec2.0 [13], HuBERT [14], Data2vec [15] and WavLM [16] were proposed recently, and in [17] using the pre-trained models to learn different levels of information at different layers was analyzed. ASR models such as the problem-agnostic speech encoder (PASE+) [18], Wav2vec-switch [19] and enhanced wav2vec2.0 [20] exhibit an excellent noise robustness in noisy environments. Based on these methods, in [21, 22] the combination with speech enhancement models was revealed to further improve the ASR accuracy in noisy scenes. The enhanced speech is fed into the pre-trained model to reduce the impact of speech distortion, which somehow supports the fact that the pre-trained representation has a strong ability to resist speech perturbation. Our previous preliminary work [23] has shown that the self-supervised WavLM model for TTS has better noise robustness. However, it is unknown whether it is appropriate for other representations to be used for TTS.

In this paper, we therefore systematically compare the performance of four self-supervised models, WavLM, Wav2vec2.0, HuBERT and data2vec, for noise robust TTS. By extracting representa-

This work is supported by the National Natural Science Foundation of China (62101523), Hefei Municipal Natural Science Foundation (2022012), USTC Research Funds of the Double First-Class Initiative (YD2100002008). (Corresponding author: Jie Zhang)

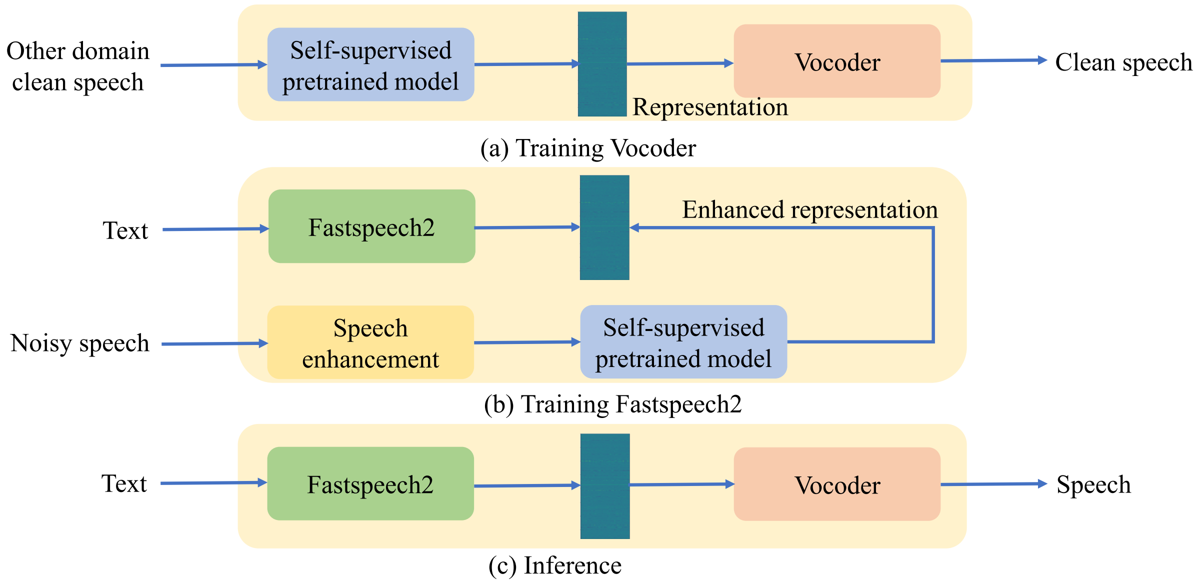

Fig. 1: The proposed TTS paradigm: (a) representation-to-waveform vocoder, (b) text-to-representation Fastspeech2, and (c) inference.

tions from different layers, we train the representation-to-waveform vocoder and the text-to-representation Fastspeech acoustic model, respectively. We measured objective evaluation metrics and speaker similarity metrics on the synthesized speech and found that 1) the higher the level of representation across models the stronger the suppression of noise and speaker information. 2) The weighted different layers of representations can balance both the noise robustness of the model and the speaker information. 3) The representation at the highest layer of the Data2vec model has the best noise and speaker suppression. Second, we found that high-level representations have the ability to decouple content information from speaker information, which naturally facilitates the voice conversion task. The vocoder is trained by concatenating speaker embeddings with the representation of the highest layer of Data2vec, which enables transitions across speakers while keeping the content of the speech unchanged. This initially validates that the representation at the highest layer has the potential to be used for voice conversion.

2. METHODOLOGY

In this section, we introduce the components of our noise-robust TTS model, including the representation-to-waveform vocoder and the text-to-representation Fastspeech2 model.

2.1. Representation-to-waveform: Vocoder

The representation-waveform vocoder follows the HiFi-GAN [24], which consists mainly of a generator and two discriminators, i.e., multi-scale and multi-period discriminators, and both the generator and the discriminator utilize multi-layer convolutional networks.The generator and the two discriminators are trained by an adversarial learning approach. The generator takes the representations of different layers of the pre-trained model as input and then upsamples them by multi-layer transpose convolutions until the length of the output sequence matches the temporal resolution of the original waveform. Each transposed convolution is followed by a multi-receptive field fusion module to achieve the purpose of modeling the features of the initial input under multiple receptive fields. The discriminator is employed to identify the signal patterns of different periodicities in the speech signal, which includes the Multi-Period Discriminator (MPD) and the Multi-Scale Discriminator (MSD).The MPD discriminator models the diverse periodic patterns inside the speech data.The MSD discriminator models the long-range information of the speech. For the detailed network structure of generators and discriminators, please refer to [24]. The procedure for training the vocoder is shown in Fig. 1(a). To ensure that the data for training the vocoder is universal, we select publicly available multi-speaker clean speech datasets from other domains. The clean speech x is fed into the pre-trained model to extract the output representation c from different layers, and then the representation c is fed into the vocoder to reconstruct the clean speech waveform. Given the generator G and discriminator D, the total generator loss function $\mathcal{L}_G$ and the discriminator loss function $\mathcal{L}_D$ for training the vocoder can be respectively formulated as

$$\mathcal{L}_G = \mathcal{L}_{adv}(G;D) + \alpha\mathcal{L}_{fm}(G;D) + \beta\mathcal{L}_{mel}(G), \tag{1}$$

$$\mathcal{L}_D = L_{adv}(D;G), \tag{2}$$

where the generative loss $\mathcal{L}_{adv}(G;D)$ and discriminative loss $\mathcal{L}_{adv}(D;G)$ are respectively given by

$$\mathcal{L}_{adv}(D;G) = \mathbb{E}_{(x,c)}\left[(D(x)-1)^2 + (D(G(c)))^2\right], \tag{3}$$

$$\mathcal{L}_{adv}(G;D) = \mathbb{E}_{(c)}\left[(D(G(c))-1)^2\right]. \tag{4}$$

The feature matching loss $\mathcal{L}_{fm}(G;D)$ and the mel-spectrogram loss $\mathcal{L}_{Mel}(G)$ in (1) keep the same as [24]. α and β are hyperparameters.

2.2. Text-to-representation: Fastspeech2

We utilize Fastspeech2 [25] to learn the mapping from text to representations. The Fastspeech2 model mainly consists of phone embedding, encoder, variance adaptor and decoder modules. The encoder consists of a multi-layer feed-forward transformer, which converts a

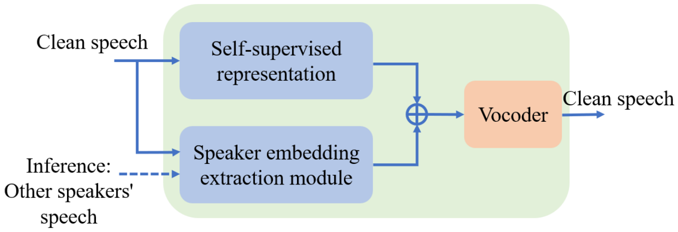
Clean speech
Self-supervised representation
Speaker embedding extraction module
Inference: Other speakers' speech
Vocoder
Clean speech

Fig. 2: The structure of the voice conversion model.

sequence of phonemes into a sequence of hidden states. The variance adaptor exploits a multi-layer convolutional network to predict duration, pitch, energy, etc., which introduces more information about variations in speech to tackle the one-to-many mapping problem in speech synthesis. The decoder consists of a linear projection layer, which is used to map the network output to the representation.During training, the representation is reconstructed by fusing the output of the variance adaptor with the output of the encoder used as input to the decoder. For more details about Fastspeech2, please refer to [25, 26]. The procedure of training text to representation is shown in Fig. 1(b). Firstly, we incorporate the original noisy speech into the speech enhancement model to obtain the enhanced speech, and then feed it into the pre-trained model to extract the corresponding representation. We train the Fastspeech2 model using paired text and enhanced representations. It is worth an additional observation that both the speech enhancement model and the pre-training model are publicly available models and their parameters do not get updated. After training, we inject the text and connect the Fastspeech2 model and the vocoder model to synthesize the speech waveform in the inference phase, as shown in Fig. 1(c).

2.3. The voice conversion model

In our experiments, we found that the representations of the higher layers of the pre-trained model have less speaker information. In order to verify that the representation at the higher layers can be used for the voice conversion task, we fuse the representation and speaker information, and the model structure is shown in Fig. 2. In the training phase, speech is fed into a pre-trained data2vec model to extract the representations of the twelfth layer, and then speech is similarly fed into a pre-trained speaker verification model to extract the speaker embeddings. The representations and speaker embeddings are added together and fed into the HiFi-GAN-based vocoder. The overall model is trained in the same way as in section 2.1. In the inference stage, we can input the speech of other speakers to extract the speaker embedding, and then realize the function of voice conversion. The speaker verification model uses the publicly accessible model Resemblyzer[1], and the entire experiment is done on the LibriTTS dataset.

3. EXPERIMENTAL SETUP

3.1. Model configurations

For the vocoder, we choose the publicly available multi-speaker clean speech dataset LibriTTS[2] [27] train-clean-100 subset in order to ensure that the data for training the vocoder is universal and does not overlap with the dataset for training FastSpeech2. In this paper, we use WavLM, Wav2vec2, HuBERT, and Data2vec representations for comparison experiments. For each type of representation, we trained seven models, extracting the representations of the 1st, 3rd, 5th, 7th, 9th, and 12th layers and averaging the representations of all layers as input. All audio sample rates are converted to 24kHz. We use mel-spectrogram features of clean speech to train a vocoder as the baseline. For the baseline model, The fast Fourier transform (FFT) size of the extracted mel-spectrogram is set to 1024, the hop size to 240, and the window size to 960. The number of frequency bins of the mel-spectrogram are set to 80, respectively. For our model, since the frame shift of the representation extracted by WavLM is 20 ms, and the frame shift of mel-spectrogram is 10ms, we set the FFT size to 1024, the hop size to 480, and the window size to 960. The dimension of representation is 768, respectively. The batch size is set to 16, and a total of 800k steps are trained. The hyper-parameters α and β in (1) are set to 2 and 45, respectively.

For training FastSpeech2, we utilize the LJSpeech dataset[3]. To simulate the noisy environment, we mix the LJSpeech speech data with noise at a signal-to-noise ratio (SNR) of 5 dB as a noisy dataset, where the noise data comes from the Freesound dataset [28]. To ensure that the speech enhancement model has not seen the LJSpeech dataset, the speech enhancement model[4] is publicly available and was trained on other datasets. The enhanced speech is fed to the pre-trained model to extract the representations of different layers. All audio sample rates are converted to 24kHz. We train two models using the mel-spectrogram of clean speech and the mel-spectrogram of enhanced speech as the baseline, respectively. For the baseline model, the FFT size of the extracted mel-spectrogram is set to 1024, the hop size to 240, and the window size to 960. The frequency bins of the mel spectrum are set to 80. For our model, we set the FFT size to 1024, the hop size to 480, and the window size to 960. The dimension of representation is 768. The batch size is set to 16, and a total of 900k steps are trained, respectively.

For the voice conversion model, we train the vocoder using data2vec representations and speaker embeddings. The dimension of the representation is 768 and the dimension of the speaker embedding is 256. The dimensions of speaker embedding are mapped to the representational dimensions by linear projection. The representations and speaker embeddings are then summed and fed to the vocoder. The entire voice conversion model is trained in the same way as the vocoder.

3.2. Evaluation metrics

For different models, we respectively generated 256 utterances from the test set. For objective evaluation metrics, we tested the Mean Opinion Score - Listening Quality Objective (MOS-LQO) using the VISQOL[5] [29] tool, and in audio mode, the MOS-LQO values ranged from 1 to 4.75, with higher values indicating better sound quality. To measure the speaker similarity of the synthesized speech, we compute the cosine distance of the speaker embedding as the speaker similarity metric. Specifically, speaker similarity metrics are computed using the public speaker verification model ECAPA-TDNN[6]. For the subjective evaluation, we use the mean opinion score (MOS) to evaluate the naturalness and speaker similarity of the speech, which ranges from 1 to 5 (the higher, the better).

[1] https://github.com/resemble-ai/Resemblyzer

[2] https://www.openslr.org/60/

[3] https://keithito.com/LJ-Speech-Dataset/

[4] https://huggingface.co/speechbrain/sepformer-wham16k-enhancement

[5] https://github.com/google/visqol

[6] https://modelscope.cn/models/damo/speech_ecapa-tdnn_sv_en_voxceleb_16k/summary

Table 1: The MOS-LQO metrics of synthesized audio using different layers of representations.

Feature	Type	MOS-LQO			
		WavLM	Wav2vec2	HuBERT	Data2vec
-	Ground Truth	-			
Mel-spectrogram	Clean	3.78			
	Enhanced	2.58			
Representation	Enhanced (Layer 1)	3.03	3.04	2.92	3.03
	Enhanced (Layer 3)	3.13	3.04	3.07	3.02
	Enhanced (Layer 5)	3.21	3.07	3.08	3.06
	Enhanced (Layer 7)	3.01	3.06	3.00	3.07
	Enhanced (Layer 9)	3.10	3.05	3.03	3.00
	Enhanced (Layer 12)	3.05	2.91	2.87	2.94
	Enhanced (Average of all layers)	3.32	3.17	3.06	2.97

Table 2: The speaker similarity metrics of synthesized audio using different layers of representations.

Feature	Type	Speaker similarity			
		WavLM	Wav2vec2	HuBERT	Data2vec
-	Ground Truth	-			
Mel-spectrogram	Clean	0.8037			
	Enhanced	0.6176			
Representation	Enhanced (Layer 1)	0.7044	0.6939	0.6942	0.7050
	Enhanced (Layer 3)	0.6661	0.6385	0.6871	0.5160
	Enhanced (Layer 5)	0.6447	0.5829	0.6395	0.2794
	Enhanced (Layer 7)	0.4522	0.5685	0.5453	0.2953
	Enhanced (Layer 9)	0.4103	0.5508	0.5063	0.2563
	Enhanced (Layer 12)	0.3747	0.2599	0.3699	0.0565
	Enhanced (Average of all layers)	0.6057	0.6759	0.6448	0.4092

4. EXPERIMENTAL RESULT

4.1. The comparison of synthesized speech with different representations

The subjective performance of the synthesized speech in terms of MOS-LQO is shown in Table 1. The baseline model trained with clean speech achieves a MOS-LQO of 3.78 and the baseline model trained with enhanced speech obtains a MOS-LQO of 2.58. This is due to the fact that the TTS models trained using the enhanced speech often contain noise (i.e., speech distortions), leading to low MOS-LQO values. Overall, the quality of synthesized speech using representations is generally better than the quality of that synthesized by the baseline model, and the noise component in the synthesized speech is significantly reduced. The quality of the synthesized speech using the WavLM representation was better than the other three representations, and the quality of the synthesized speech using the Data2vec representation was the worst. The quality of synthesized speech is similar using intermediate layers of representations, e.g., the 3rd, and 5th layers. In addition, models trained using representations averaged over all layers can synthesize better speech. For example, the model trained with the WavLM representation averaged over all layers achieved the best performance with a MOS-LQO of 3.32.

4.2. The speaker similarity of synthesized speech with different representations

In order to measure the speaker similarity of speech synthesized with different representations, we use the publicly available ECAPA-TDNN model to measure speaker similarity, and the experimental results are shown in Table 2. It is worth mentioning that the metrics in this paper are different from those in [23]. The results in [23] use an internal speaker verification model, which is not publicly available, so for convenience, the publicly available model ECAPA-TDNN is therefore used. Using the Mel-spectrum features of clean speech to train the model, the speaker similarity of the synthesized speech on the test set was able to reach 0.8037. When the Mel-spectrum features of enhanced speech are used to train the model, the speaker similarity of the synthesized speech reaches 0.6176. When the representation is used to train the model, the higher the layer of representation used the lower the speaker similarity of the synthesized speech. All four representations performed consistently. In addition, we were surprised to find that the representation at layer 12 of Data2vec has the least amount of speaker information. This indicates that the representation has a good decoupling effect.

4.3. The similarity of synthesized speech from voice conversion models.

To measure the naturalness and speaker similarity of the speech generated by the voice conversion model, we tested the MOS scores of the generated speech, and the experimental results are shown in Table 3. We found that voice conversion can be achieved well using data2vec representations, which suggests that data2vec representations have good potential for voice conversion tasks. However, the speaker similarity of the synthesized speech needs to be further improved, which we guess is due to the small dataset and insufficient training, and we will continue to validate it on a larger dataset subsequently.

Table 3: The MOS scores of the speech generated by the voice conversion model.

Model	MOS
Ground Truth	4.0
Voice conversion	3.2

5. CONCLUSION

In this paper, we systematically investigate representation-based noise robust TTS models. By constructing representation-to-waveform vocoders, and text-to-representation acoustic models, we find that representation-based TTS models have better noise robustness than mel-spectrogram-based TTS models. Averaging the representations across all layers provides a good balance of noise robustness and speaker information, and multiple self-supervised representations perform consistently. Furthermore, we found that the data2vec representation has the best noise suppression and speaker information suppression, and it has the potential to be applied to voice conversion tasks.

6. REFERENCES

[1] D. Klatt, "Review of text-to-speech conversion for english," *The J. of the Acoust. Soc. of America*, vol. 82, no. 3, pp. 737–793, 1987.

[2] Z. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted boltzmann machines and deep belief networks for statistical parametric speech synthesis," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 10, pp. 2129–2139, 2013.

[3] Y. Gu, X. Yin, Y. Rao, Y. Wan, et al., "Bytesing: A chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and wavernn vocoders," in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2021, pp. 1–5.

[4] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Speech enhancement for a noise-robust text-to-speech synthesis system using deep recurrent neural networks," in *Proc. Interspeech*, 2016, pp. 352–356.

[5] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Investigating rnn-based speech enhancement methods for noise-robust text-to-speech.," in *SSW*, 2016, pp. 146–152.

[6] C. Valentini-Botinhao and J. Yamagishi, "Speech enhancement of noisy and reverberant speech for text-to-speech," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 8, pp. 1420–1433, 2018.

[7] D. Dai, L. Chen, Y. Wang, M. Wang, et al., "Noise robust tts for low resource speakers using pre-trained model and speech enhancement," *arXiv preprint arXiv:2005.12531*, 2020.

[8] W. Hsu, Y. Zhang, R. J Weiss, H. Zen, et al., "Hierarchical generative modeling for controllable speech synthesis," in *Proc. ICLR*, 2018.

[9] W. Hsu, Y. Zhang, R. Weiss, Y. Chung, et al., "Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization," in *ICASSP*, 2019, pp. 5901–5905.

[10] C. Zhang, Y. Ren, X. Tan, J. Liu, et al., "Denoispeech: Denoising text to speech with frame-level noise modeling," in *ICASSP*, 2021, pp. 7063–7067.

[11] T. Saeki, K. Tachibana, and R. Yamamoto, "DRSpeech: Degradation-Robust Text-to-Speech Synthesis with Frame-Level and Utterance-Level Acoustic Representation Learning," in *Proc. Interspeech*, 2022, pp. 793–797.

[12] D. Yang, S. Liu, J. Yu, H. Wang, et al., "Norespeech: Knowledge distillation based conditional diffusion model for noise-robust expressive tts," *arXiv preprint arXiv:2211.02448*, 2022.

[13] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Proc. NeurIPS*, vol. 33, pp. 12449–12460, 2020.

[14] W. Hsu, B. Bolte, Y. Tsai, K. Lakhotia, et al., "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3451–3460, 2021.

[15] A. Baevski, W. Hsu, Q. Xu, A. Babu, et al., "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *Proc. ICML*, 2022, pp. 1298–1312.

[16] S. Chen, C. Wang, Z. Chen, Y. Wu, et al., "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE J. of Selected Topics in Signal Process.*, vol. 16, no. 6, pp. 1505–1518, 2022.

[17] A. Pasad, J. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *Proc. ASRU*, 2021, pp. 914–921.

[18] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, et al., "Multi-task self-supervised learning for robust speech recognition," in *ICASSP*, 2020, pp. 6989–6993.

[19] Y. Wang, J. Li, H. Wang, Y. Qian, et al., "Wav2vec-switch: Contrastive learning from original-noisy speech pairs for robust speech recognition," in *ICASSP*, 2022, pp. 7097–7101.

[20] Q. Zhu, J. Zhang, Z. Zhang, M. Wu, et al., "A noise-robust self-supervised pre-training model based speech representation learning for automatic speech recognition," in *ICASSP*, 2022, pp. 3174–3178.

[21] X. Chang, T. Maekaku, Y. Fujita, and S. Watanabe, "End-to-End Integration of Speech Recognition, Speech Enhancement, and Self-Supervised Learning Representation," in *Proc. Interspeech*, 2022, pp. 3819–3823.

[22] Q. Zhu, J. Zhang, Z. Zhang, and L. Dai, "A joint speech enhancement and self-supervised representation learning framework for noise-robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 1927–1939, 2023.

[23] Q. Zhu, Y. Gu, C. Weng, Y. Hu, L. Dai, and J. Zhang, "Rep2wav: Noise robust text-to-speech using self-supervised representations," *arXiv preprint arXiv:2308.14553*, 2023.

[24] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Proc. NeurIPS*, vol. 33, pp. 17022–17033, 2020.

[25] Y. Ren, C. Hu, X. Tan, T. Qin, et al., "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *Proc. ICLR*, 2020.

[26] Y. Ren, Y. Ruan, X. Tan, T. Qin, et al., "Fastspeech: Fast, robust and controllable text to speech," *Proc. NeurIPS*, vol. 32, 2019.

[27] H. Zen, V. Dang, R. Clark, Y. Zhang, et al., "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech," in *Proc. Interspeech*, 2019, pp. 1526–1530.

[28] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *IEEE Trans. Multimedia*, 2013, p. 411–412.

[29] M. Chinen, F. Lim, J. Skoglund, N. Gureev, et al., "Visqol v3: An open source production ready objective speech and audio metric," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, 2020, pp. 1–6.