Interspeech 2022
18-22 September 2022, Incheon, Korea

Ultra-Low-Bitrate Speech Coding with Pretrained Transformers

Ali Siahkoohi[1], Michael Chinen[2], Tom Denton[2], W. Bastiaan Kleijn[2,3], and Jan Skoglund[2]

[1]School of Computational Science and Engineering, Georgia Institute of Technology
[2]Chrome Media, Google
[3]School of Engineering and Computer Science, Victoria University of Wellington

alisk@gatech.edu, mchinen@google.com, tomdenton@google.com,
bastiaan.kleijn@ecs.vuw.ac.nz, jks@google.com

Abstract

Speech coding facilitates the transmission of speech over low-bandwidth networks with minimal distortion. Neural-network based speech codecs have recently demonstrated significant improvements in quality over traditional approaches. While this new generation of codecs is capable of synthesizing high-fidelity speech, their use of recurrent or convolutional layers often restricts their effective receptive fields, which prevents them from compressing speech efficiently. We propose to further reduce the bitrate of neural speech codecs through the use of pretrained Transformers, capable of exploiting long-range dependencies in the input signal due to their inductive bias. As such, we use a pretrained Transformer in tandem with a convolutional encoder, which is trained end-to-end with a quantizer and a generative adversarial net decoder. Our numerical experiments show that supplementing the convolutional encoder of a neural speech codec with Transformer speech embeddings yields a speech codec with a bitrate of 600 bps that outperforms the original neural speech codec in synthesized speech quality when trained at the same bitrate. Subjective human evaluations suggest that the quality of the resulting codec is comparable or better than that of conventional codecs operating at three to four times the rate.

Index Terms: speech coding, Transformers, self-supervised learning, generative adversarial nets.

1. Introduction

Speech compression aims to reduce the bitrate required to represent a speech signal. In classical coding methods [1–7], all processing was based on knowledge of human experts only. Recent advances in speech coding follow progress in speech synthesis [8–10] by replacing the decoder [11–13] as well as the quantizer [14] with a machine-learning (ML) based model that significantly improve the coding quality. More recently, end-to-end coding schemes have been developed [15, 16] that employ an autoencoder structure with quantization in the bottleneck (latent space). With SoundStream [16], this autoencoding structure, in the form of a VQ-VAE [17], was further combined with the learned distortion measures from generative adversarial networks (GANs) [18]. While [16] represents the current state-of-the-art above 3 kbps, its relative effectiveness deteriorates at lower rates. Indications exist [19] that lengthening the effective receptive field of the encoder may improve performance at very low rates. This motivates us to study the combination of the approach of [16] with an encoder that can exploit long-term dependencies in the input speech signal.

Our work focuses on reducing the bandwidth of neural speech codecs by incorporating speech embeddings that are obtained from a pretrained Transformer model [20–22]. The Transformer is pretrained in a self-supervised learning framework [20, 22] that involves performing a contrastive task [17] over large quantities of raw speech. Owing to their multi-head attention layers [23], Transformers have the inductive bias to exploit long-distance dependencies in the input speech, which enables them to learn speech embeddings that result in state-of-the-art performance on speech-related downstream tasks [20–22, 24, 25].

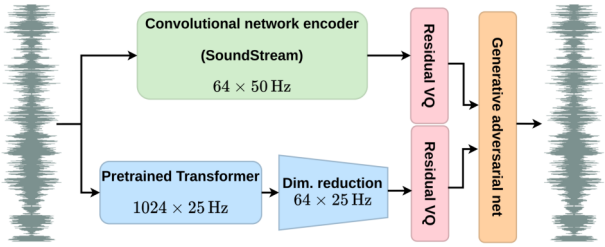

Figure 1: *The proposed speech codec. Speech is encoded via Transformer embeddings and SoundStream-based CNN encoder features. After quantization with residual VQs, a GAN-based decoder reconstructs the input speech.*

To design the encoder in our ultra-low-bitrate codec, we supplement features learned via a convolutional neural network (CNN) speech encoder based on SoundStream [14, 16] with speech embeddings obtained from a pretrained Transformer model. The Transformer embeddings and encoder speech features are concatenated after quantization—with residual vector quantizers (VQs) [26]—and fed to the decoder. We rely on GANs for the decoder design for their ability to synthesize speech with high perceptual quality [9, 10, 16]. Training both the CNN encoder, residual VQ, and the GAN-based decoder results in a noncausal speech codec that, at a rate of 600 bps, reproduces speech with a quality significantly higher than traditional or CNN-based codecs operating at similar rates. Figure 1 schematically presents our proposed codec. Some of the quality differences in synthesized speech can be attributed to the noncausality of our codec, and further research on causal Transformer models is required to precisely quantify the gains resulting from speech features that exploit long-distance dependencies.

Our work is closely related to [24], which synthesizes speech from noncausal discrete speech features, involving speech embeddings obtained from self-supervised trained Transformer models. In contrast to our approach, the encoder in [24] includes an additional pitch extractor [27] and speaker identity [28] models. While the authors reported high-quality speech synthesis at 365 bps, our choice of encoder design is

This work was done during a research internship at Google.

10.21437/Interspeech.2022-10988

more general in that it can be adapted to any neural speech codec by augmenting the encoder with Transformer-predicted speech embeddings. Additionally, our proposed codec eliminates the need to have a speaker identity model, the usage of which may lead to privacy concerns.

Our primary contribution is the reduction of the bitrate of neural speech codecs by using speech embeddings derived from pretrained Transformer models. In particular, we propose an end-to-end trained speech coding approach in which the input speech is encoded by combining Transformer embeddings with CNN encoded features. We demonstrate that the resulting codec is capable of outperforming codecs operating at three to four times the rate according to subjective human evaluation metrics.

In what follows, we first describe self-supervised learning, including the specific framework for the speech domain, and the Transformer architecture we use. Next, we introduce our Transformer-based codec, including details of the encoder, quantizer, and decoder modules. Finally, we provide experimental results, evaluating our codec with objective and subjective metrics.

2. Self-supervised learning

With the goal of acquiring knowledge from large quantities of raw speech, self-supervised learning involves performing a proxy task [17] via a neural network over the raw dataset. This task typically entails predicting the components of the input that have been masked [20]. In order to accomplish this task more effectively, the neural network model must extract high-level features from contextual data that inform the model about the masked components. Due to their inductive bias, Transformers, which have large effective receptive fields by design [23], are often employed as neural network models for the purpose of self-supervised learning. Following training, the learned embeddings are used to perform various downstream tasks [20, 22, 25]. Before presenting the self-supervised learning framework aimed at speech, we introduce the architecture for the Transformer model.

2.1. Transformer architecture

We use the Transformer model introduced by [22], which consists of two sub-networks: (1) a feature encoder network that maps input raw speech to latent speech representations; and (2) a context network that predicts context representations given the latent representations. The feature encoder converts the $16\,\mathrm{kHz}$ raw input speech into a log mel-spectrogram, which is subsequently passed to a series of subsampling convolutional layers. The overall subsampling rate of the feature encoder network is 640, which results in latent speech representations with a $25\,\mathrm{Hz}$ frame rate. This design aims to reduce the high dimensionality of speech signals, to facilitate the learning of long-distance dependencies between the input features more easily [20]. The latent speech representation is then passed to the context network, consisting of a linear layer, followed by a stack of 24 Conformer blocks [21], each of which is a series of multi-headed self attention [23], depth-wise convolution, and linear layers. With this architecture, the Transformer outputs speech embeddings of size 1024 with a $25\,\mathrm{Hz}$ frame rate.

2.2. wav2vec 2.0—a self-supervised learning framework

The training objective in wav2vec 2.0 is based on predicting certain masked components in the latent speech space. To achieve this, the masked latent speech representations are passed to the context network to yield predicted context representations. The objective of the Transformer is to correctly predict target context representations, which are the result of applying a linear layer to unmasked latent speech representations [22]. To train the Transformer, we apply a modified wav2vec 2.0 pretraining procedure that was introduced by [22], which minimizes a contrastive loss [17] between the predicted and target context representations in the masked positions. This ensures that the context representation associated with the masked portion is accurately predicted while being dissimilar to other target context representations. After pretraining, the speech embeddings extracted from the Transformer models can be used to solve downstream tasks [17, 20, 22, 24], including speech coding. In the next section, we explore speech coding as a downstream task and describe how to exploit the learned speech embeddings for designing an low-bitrate speech codec.

3. Transformer-based speech codec

Speech codecs are usually composed of three components: an encoder, a quantizer, and a decoder. The encoder takes raw speech as an input and extracts low-rate features that contain sufficient information to reconstruct the speech. For a given bitrate, the quantization module finds discrete representations of the inherently continuous encoded features. Lastly, the decoder reconstructs the input speech signal from the discrete encoded features. In the following sections, we describe the encoder, quantizer, and decoder modules of the proposed speech codec.

3.1. Encoder

We use speech embeddings derived from a pretrained Transformer model, $E_T : \mathcal{X} \to \mathcal{E}_T$ for speech coding, where $\mathcal{X}$ and $\mathcal{E}_T$ are the raw speech and Transformer embeddings spaces, respectively. The Transformer takes raw speech,$\mathbf{x} \in \mathcal{X}$, as input and extracts low-rate speech features, $E_T(\mathbf{x}) \in \mathcal{E}_T$. We use the $21\,\mathrm{st}$ Conformer block for embedding extraction as it results in higher quality synthesized speech. Similar observations have been made for other speech-related downstream tasks [29, 30]. To encode other sources of information that the Transformer embeddings potentially lack [24], we concatenate the embeddings with features learned from a CNN encoder based on SoundStream [16]. This encoder, denoted by $E_C : \mathcal{X} \to \mathcal{E}_C$, takes raw speech as input and generates 64 dimensional speech features with a frame rate of $50\,\mathrm{Hz}$.

3.2. Quantizer

Transmission of continuous speech features over low-bandwidth channels is achieved via VQs [26], where the features are turned into discrete representations while introducing minimal distortion. To prevent the requirement to store a very large codebook, we utilize residual VQs [16, 26] in which the allotted codebook size is distributed among a cascade of VQs. This approach has the advantage of increasing the bitrate for a fixed actual codebook size [16]. In this approach, each VQ uses the quantization error of the preceding VQ as input with the first VQ inputting the original feature vector. We use two independent residual VQs for quantizing the Transformer embeddings and CNN features. To improve the quantization efficiency of the Transformer embeddings, we reduce their dimensionality via a 1024×64 linear layer. In the rest of the paper, $Q(\cdot)$ denotes the quantizer module, involving two residual VQs and the dimensionality reduction operator. Quantization is inherently non-differentiable. Therefore, during minimization of

loss functions that involve quantization, we define the gradient of $Q(\mathbf{x})$ with respect to $\mathbf{x}$ as identity. While this introduces errors in gradients, empirical observations [16] suggest that the error does not prohibit the optimization procedure from providing reasonable results.

3.3. Decoder

Following quantization, the decoder synthesizes the original speech signal. In this work, we adapt the GAN-based decoder proposed in the SoundsStream coder [16], motivated by its ability to synthesize high perceptual quality speech. We replicate the 25 Hz quantized Transformer embeddings to match the 50 Hz frame rate of CNN encoder features. After concatenation, these features are passed to the generator network, $G : \mathcal{E}_T \times \mathcal{E}_C \rightarrow \mathcal{X}$, which aims to map quantized speech features back to the speech domain. Our adversarial training framework (cf. Section 4) relies on two types of discriminators, time-domain and short-time Fourier transform (STFT) discriminators. The STFT discriminator [16], D_0, uses the STFT of the speech as input with real and imaginary parts as separate channels. On the other hand, three wave domain discriminators [9, 16], denoted by D_k, $k = 1, 2, 3$, use the speech signal at different resolutions, e.g., original signal and subsampled by factors of two and four, as their input. This allows them to learn the characteristics of speech signal at different time scales, improving the perceptual quality of the synthesized speech [16].

4. Training

We train the parameters involved in our proposed codec in an end-to-end fashion, except for the pretrained Transformer weights, which are kept fixed. The optimization problem consists of minimizing a series of loss functions to determine the codec parameters $\boldsymbol{\theta}$, of the CNN encoder, the dimensionality reduction operator, the residual VQs, and the generator network, as well as the weights for the discriminators, denoted by $\boldsymbol{\phi}$. Note that the Transformer is fixed and we do not finetune it. The resulting loss function is the weighted sum of the loss functions described below, where the weights are hyperparameters.

4.1. Adversarial loss

GANs are known for their ability to generate speech with high perceptual quality [9, 16]. Their success is partially attributed to the adversarial loss, which involves training discriminators to guide the generator network to produce high quality speech. In this work, we use a hinge GAN loss function [31] given by,

$$\mathcal{L}_{\boldsymbol{\theta}}^{\text{adv}} = \mathbb{E}_{\mathbf{x}}\left[\frac{1}{4}\sum_{k=0}^{3}\frac{1}{T_k}\max\Big(0, 1 - D_k \circ G \circ Q \circ E(\mathbf{x})\Big)\right], \tag{1}$$

where E is the encoder, Q the quantizer module, T_k, $k = 0, \ldots 3$, the number of logits at the D_k output along the time dimension, and $\circ$ the composition operator. The loss function (1) is used to update the parameter $\boldsymbol{\theta}$. On the other hand, the discriminators loss function for updating $\boldsymbol{\phi}$ is

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\phi}}^{\text{adv}} = \mathbb{E}_{\mathbf{x}}&\left[\frac{1}{4}\sum_{k=0}^{3}\frac{1}{T_k}\max\Big(0, 1 - D_k(\mathbf{x})\Big)\right] \\ &+\mathbb{E}_{\mathbf{x}}\left[\frac{1}{4}\sum_{k=0}^{3}\frac{1}{T_k}\max\Big(0, 1 + D_k \circ G \circ Q \circ E(\mathbf{x})\Big)\right].\end{aligned} \tag{2}$$

4.2. Feature matching loss

In addition to the output of the discriminators, we use their feature maps, i.e., values in the intermediate layers, to construct a loss function. This loss, known as the feature matching loss [9], minimizes the difference in discriminator features maps for real and synthesized input speech. This difference, usually calculated as an ℓ_1 distance, acts as a learned metric function and has proven useful for generating high quality speech with GANs [9]. The loss function is given by

$$\mathcal{L}_{\boldsymbol{\theta}}^{\text{feat}} = \mathbb{E}_{\mathbf{x}}\left[\sum_{k=0}^{4}\sum_{l=1}^{L}\frac{1}{4LT_{k,l}}\Big\|D_k^{(l)}(\mathbf{x}) - D_k^{(l)} \circ G \circ Q \circ E(\mathbf{x})\Big\|_1\right], \tag{3}$$

where l denotes the layer index of the discriminators.

4.3. Reconstruction loss

To ensure fidelity of the synthesized speech with respect to the encoder features, we enforce the input and synthesized speech signals to be coherent in the mel-spectrogram domain by minimizing [16]

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\theta}}^{\text{recon}} = &\sum_{s\in\{2^i|i=6,\ldots,11\}}\sum_{t}\Big\|\mathcal{S}_t^s(x) - \mathcal{S}_t^s \circ G \circ Q \circ E(\mathbf{x})\Big\|_1 \\ &+\sqrt{\frac{s}{2}}\sum_{t}\Big\|\log\mathcal{S}_t^s(\mathbf{x}) - \log\mathcal{S}_t^s \circ G \circ Q \circ E(\mathbf{x})\Big\|_2,\end{aligned} \tag{4}$$

where S_t^s denotes the t^{th} frame of a 64-bin mel-spectrogram computed with window length s and hop length $s/4$.

4.4. Quantization loss

To train the residual VQ in the context of our speech codec, we initialize the codebooks via running k-means on a batch of feature vectors, i.e., $E(\mathbf{x})$ for a batch of $\mathbf{x}$, as advocated by [16]. Here E represents either E_T or E_C. The update of codebooks during training involves: (1) bringing the codebook closer to the output of the encoder $E(\mathbf{x})$ via the codebook loss and (2) encouraging the output of the encoder to create features close to the codebook via the commitment loss [32]. Note that, as for E_T, the Transformer weights are kept fixed and only the dimensionality reduction operator is optimized. By combining these two loss functions we arrive at,

$$\mathcal{L}_{\boldsymbol{\theta}}^{\text{quant}} = \|\text{sg}[E(\mathbf{x})] - \mathbf{e}\|_2^2 + \beta\|\text{sg}[\mathbf{e}] - E(\mathbf{x})\|_2^2, \tag{5}$$

where $\mathbf{e}$ is the codebook entry nearest to $E(\mathbf{x})$, sg is the stop gradient operation, which prevents the gradients from backpropagating during optimization, and β is a hyperparameter.

5. Experiments

In the experiments presented here, we compare the quality of synthesized speech with our proposed codec with traditional or neural speech codecs via objective and subjective metrics. We begin with describing the details of training our codec.

5.1. Training configurations

To create our codec modules, we adapted the architectures of the encoder, generator, and discriminators used in SoundStream [16]. In order to evaluate the performance of the proposed codec as a function of bitrate we trained nine codecs

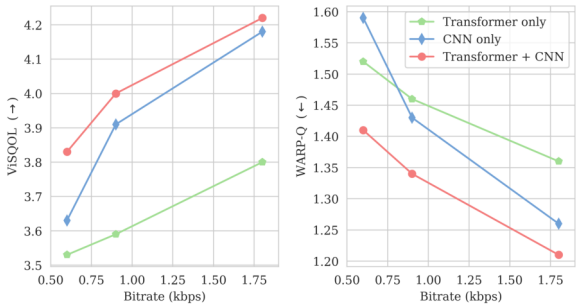

Figure 2: *ViSQOL [34] (left, higher is better) and WARP_Q [35] (right, lower is better) metrics for three encoder designs: (1) pretrained Transformer only (green); (2) CNN encoder only (blue); and (3) Transformer and CNN encoder together (red).*

identical in encoder and decoder architectures, but operating at 600 bps, 900 bps, and 1800 bps, where at each bitrate we trained three codecs with varying bitrate allocations: (1) using only Transformer embeddings; (2) using only CNN features; (3) using both sources of information. To adjust the total bitrate of the codec as well as to distribute it amongst the CNN and the Transformer, we fixed the codebook size of each VQ to 64 but varied the number of cascaded VQs. For the third set of codecs, we selected the bitrate allocation between Transformer embeddings and CNN features that results in the best synthesized speech quality. This amounts to an even split of bitrates for the 600 bps and 1800 bps codecs, and a 1/3–2/3 Transformer–CNN split for the codec operating at 900 bps. We trained these models on a training split subset of the Mozilla Common Voice and LibriVox datasets (approximately 700 hours each), by taking up to 10 random 1.28 s segments per file, which discards a significant portion of longer files. We computed Transformer embeddings over the entire utterance samples and created batches of 256 1.28 s utterance segments and Transformer embedding pairs for training. The training involved minimizing the combination of the loss functions described in Section 4 with equal weights of 1.0, except for the features matching and quantization losses, where they are weighted by 100.0 and 0.4, respectively. We used 200 k iterations of Adam optimizer with learning rate 10^{-4}. The set of weights for balancing different loss objectives as well as the optimizer learning rate were obtained via extensive hyperparameter tuning. For testing, we evaluated the quality of the synthesized speech using 4 s long clean English speech utterances from the VCTK dataset [33]. We did not test our method on noisy utterances as the data during training was not contaminated with noise.

5.2. Results

For a fixed bitrate, combining Transformer embeddings and CNN encoder features produces higher quality speech than the use of either Transformer embeddings or CNN encoder features alone We computed ViSQOL [34]—an objective metric for estimating perceptual speech quality—and WARP_Q [35]—an objective metric specifically designed for neural speech codec—shown in Figure 2. We observe that our proposed codec consistently achieves higher ViSQOL and lower WARP_Q scores, indicating better speech quality across the three selected bitrates. This observations suggests that the CNN encoder learns to encode certain speech information that compliments the Transformer embeddings in synthesizing speech. We also qualitatively noticed that using only Transformer embeddings for speech synthesis sometimes leads to speaker identity distortions

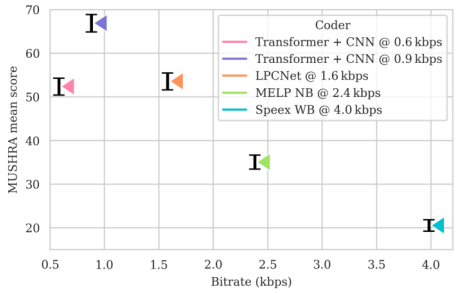

Figure 3: *MUSHRA subjective test. Codecs closer to the top-left corner have better perceptual quality and lower bitrates. The reference with MUSHRA score near* 100 *is not included in the figure. The indicted interval in black represents the* 95% *confidence interval for each score.*

which were absent in our proposed codec.

We evaluated the perceptual quality of the synthesized speech produced by our ultra-low bitrate codecs, operating at 600 bps and 900 bps, by comparing to Speex wideband [7], MELP narrowband [2], and LPCNet [12], operating at 4 kbps, 2.4 kbps, and 1.6 kbps, respectively. We performed MUSHRA [36] subjective evaluations of our proposed codec by human raters in order to evaluate its performance. The test used 24 raters per utterance (78 raters in total) over 32 clean speech male and female utterances. We further post-screened the raters by removing the ones that did not rate the hidden reference above 90 at least 80 percent of the time. We did not include the codecs utilizing only one of CNN features or Transformer embeddings in this subjective test as their perceptual quality was clearly worse than that of the proposed codec. Figure 3 plots the mean MUSHRA score and 95% confidence intervals as a function of codec bitrate, where higher values indicate better quality. We observe that our 900 bps codec clearly outperforms all the other codecs. The 600 bps codec performs as well as the LPCnet codec while outperforming MELP and Speex.

6. Conclusions

Reducing the bandwidth required to transmit speech while preserving the perceptual quality remains challenging. To extract high performance from an ultra-low-bitrate neural speech codec, we utilized the long-distance dependencies inherent in speech signal by incorporating speech embeddings from a pretrained Transformer in the encoding phase. Using these speech embeddings in conjunction with speech features encoded by a convolutional encoder yielded a noncausal speech codec capable of operating at 600 bps with high perceptual quality. Our experiments show that the proposed codec significantly outperforms the original neural speech codec with respect to the quality of synthesized speech when operating in the ultra-low bitrate regime. In addition, the subjective experiments indicate comparable to or better perceptual speech quality compared to conventional codecs operating at three to four times the rate. Further research on causal Transformer models is required to quantify the extent to which the gain in bitrate is related to the incorporating long-distance dependencies in the speech signal.

7. Acknowledgements

We thank Marco Tagliasacchi and Neil Zeghidour for providing the SoundStream architecture and advice, and Wei Han and the speech team for providing the pretrained Transformer model.

8. References

[1] W. B. Kleijn, P. Kroon, and D. Nahumi, "The rcelp speech-coding algorithm," *European Transactions on Telecommunications*, vol. 5, no. 5, pp. 573–582, 1994.

[2] M. R. Bielefeld and L. M. Supplee, "Developing a test program for the DoD 2400 bps vocoder selection process," in *International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 2. IEEE, 1996, pp. 1141–1144.

[3] A. McCree, K. Truong, E. B. George, T. P. Barnwell, and V. Viswanathan, "A 2.4 kbit/s MELP coder candidate for the new US federal standard," in *International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1. IEEE, 1996, pp. 200–203.

[4] T. Wang, K. Koishida, V. Cuperman, A. Gersho, and J. Collura, "A 1200/2400 bps coding suite based on MELP," in *Speech Coding, 2002, IEEE Workshop Proceedings.*, 2002, pp. 90–92.

[5] J.-M. Valin, K. Vos, and T. Terriberry, "Definition of the Opus audio codec," *IETF, September*, 2012.

[6] M. Dietz, M. Multrus, V. Eksler, V. Malenovsky, E. Norvell, H. Pobloth, L. Miao, Z. Wang, L. Laaksonen, A. Vasilache *et al.*, "Overview of the EVS codec architecture," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5698–5702.

[7] J.-M. Valin, "Speex: A free codec for free speech," *arXiv preprint arXiv:1602.08668*, 2016.

[8] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[9] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," *arXiv preprint arXiv:1910.06711*, 2019.

[10] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[11] W. B. Kleijn, F. S. Lim, A. Luebs, J. Skoglund, F. Stimberg, Q. Wang, and T. C. Walters, "Wavenet Based Low Rate Speech Coding," in *International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 676–680.

[12] J.-M. Valin and J. Skoglund, "LPCNet: Improving Neural Speech Synthesis Through Linear Prediction," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5891–5895.

[13] J. Klejsa, P. Hedelin, C. Zhou, R. Fejgin, and L. Villemoes, "High-quality speech coding with Sample RNN," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7155–7159.

[14] C. Gârbacea, A. van den Oord, Y. Li, F. S. Lim, A. Luebs, O. Vinyals, and T. C. Walters, "Low bit-rate speech coding with VQ-VAE and a WaveNet decoder," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 735–739.

[15] F. S. Lim, W. B. Kleijn, M. Chinen, and J. Skoglund, "Robust low rate speech coding based on cloned networks and wavenet," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6769–6773.

[16] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[17] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[19] S. Jafarlou, S. Khorram, V. Kothapally, and J. H. Hansen, "Analyzing large receptive field convolutional networks for distant speech recognition," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 252–259.

[20] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[21] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[22] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, "Pushing the limits of semi-supervised learning for automatic speech recognition," *arXiv preprint arXiv:2010.10504*, 2020.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[24] A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W.-N. Hsu, A. Mohamed, and E. Dupoux, "Speech Resynthesis from Discrete Disentangled Self-Supervised Representations," *arXiv preprint arXiv:2104.00355*, 2021.

[25] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "W2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," *arXiv preprint arXiv:2108.06209*, 2021.

[26] A. Vasuki and P. Vanathi, "A review of vector quantization techniques," *IEEE Potentials*, vol. 25, no. 4, pp. 39–47, 2006.

[27] K. Kasi and S. A. Zahorian, "Yet Another Algorithm for Pitch Tracking," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 2002, pp. I–361–I–364.

[28] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[29] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," *arXiv preprint arXiv:2107.04734*, 2021.

[30] J. Shor, A. Jansen, W. Han, D. Park, and Y. Zhang, "Universal Paralinguistic Speech Representations Using Self-Supervised Conformers," *arXiv preprint arXiv:2110.04621*, 2021.

[31] J. H. Lim and J. C. Ye, "Geometric GAN," *arXiv preprint arXiv:1705.02894*, 2017.

[32] A. van den Oord, O. Vinyals, and k. kavukcuoglu, "Neural Discrete Representation Learning," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[33] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92)," 2019.

[34] M. Chinen, F. S. Lim, J. Skoglund, N. Gureev, F. O'Gorman, and A. Hines, "ViSQOL v3: An open source production ready objective speech and audio metric," in *International conference on quality of multimedia experience (QoMEX)*. IEEE, 2020, pp. 1–6.

[35] W. A. Jassim, J. Skoglund, M. Chinen, and A. Hines, "WARP-Q: Quality prediction for generative neural speech codecs," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 401–405.

[36] ITU-R, "Recommendation BS.1534-1: Method for the subjective assessment of intermediate quality level of coding systems," *International Telecommunications Union, Geneva, Switzerland*, vol. 2, 2001.