ENHANCING INTO THE CODEC: NOISE ROBUST SPEECH CODING WITH VECTOR-QUANTIZED AUTOENCODERS

Jonah Casebeer∗† Vinjai Vale∗♭ Umut Isik♭
Jean-Marc Valin♯ Ritwik Giri♭ Arvindh Krishnaswamy♯

† University of Illinois at Urbana-Champaign,♭ Stanford University
♯ Amazon Web Services

ABSTRACT

Audio codecs based on discretized neural autoencoders have recently been developed and shown to provide significantly higher compression levels for comparable quality speech output. However, these models are tightly coupled with speech content, and produce unintended outputs in noisy conditions. Based on VQ-VAE autoencoders with WaveRNN decoders, we develop compressor-enhancer encoders and accompanying decoders, and show that they operate well in noisy conditions. We also observe that a compressor-enhancer model performs better on clean speech inputs than a compressor model trained only on clean speech.

Index Terms—speech enhancement, speech coding, audio compression

1. INTRODUCTION

Audio codecs compress speech signals by eliminating redundant and unnecessary information, with their design often leveraging extensive domain expertise to keep compression rates high, while keeping artifacts at a minimum. The most popular codecs, like the Opus codec in wideband mode, can produce high-quality speech compression at around 9 kb/s [1]. Recently, there have been successful efforts in building learned codecs; starting with replacing the decoders with learned encoder-decoders and accompanying decoders, and show that they operate well in noisy conditions. We also observe that a compressor-enhancer model performs better on clean speech inputs than a compressor model trained only on clean speech.

We measure the performance of our models using Mean Opinion Scores (MOS) from a crowd-sourced study on Amazon Mechanical Turk. Across a range of compression rates and noise levels, we compare our model to a non-enhancing learned compressor both with and without additional enhancement preprocessing, as well as the LPCNet neural codec [6]. We find that the proposed model performs well in noisy scenarios, compared to both non-enhancing codecs, as well as the composition, with comparable total compute cost, of a separate speech enhancer and a non-enhancing codec. We also find that the compressor-enhancer VQ-VAE codec performs significantly better than a clean-speech-trained VQ-VAE codec on clean speech inputs.
2. METHOD

The task of speech enhancement is to recover a clean speech signal $s$ from a noisy and possibly reverberant mixture $x = s + h + n$ where $n$ represents some additive noise signal and $s + h$ represents the convolution of a room impulse response $h$ with the speech signal. The goal of speech compression is to reconstruct a speech signal $s$ after encoding it to a smaller representation $\hat{E}$ and decoding it to the reconstruction $\hat{s}$. A neural compression model is composed of an encoder network $\mathcal{N}_e$, a coding step $C$ and decoder network $\mathcal{N}_d$ which balance a trade-off between reconstruction fidelity and the size of $\hat{E}$.

Thus, we define the joint compression-enhancement task. In the joint compression-enhancement task, the model receives a noisy input $x$, which encodes to a smaller representation $\hat{E}$ and then decodes to an estimated clean and decompressed speech signal $\hat{s}$. The full procedure is therefore

$$\hat{s} = \mathcal{N}_d(C(\mathcal{N}_e(x))).$$  \hspace{1cm} (1)

2.1. Model

The proposed autoencoder model (Fig. 1) is comprised of a convolutional encoder, a VQ-VAE bottleneck, and a recurrent decoder. These correspond to $\mathcal{N}_e$, $C$, and $\mathcal{N}_d$ respectively. The model takes as input a 16 kHz noisy speech signal which is processed by the encoder and quantized by the bottleneck. The decoder autoregressively reconstructs the original 16 kHz waveform using the quantized speech and speaker encodings. As such, the encoder is encouraged to produce a compressed representation that gives the most information for the decoder to conditionally model the clean speech signal.

2.1.1. Encoder

The encoder first computes a log-Mel representation and then applies a series of 1D convolutional layers, treating the mel bins as features. Each convolutional layer is followed by a batch normalization and then a ReLU non-linearity. The stride of the log-Mel representation and convolutional layers are selected to produce encodings at a rate of 50 Hz. The encoder also estimates one additional “speaker embedding” vector by performing a simple average across time over the output of a separate set of encoding layers. The output of theses steps is called $E$.

2.1.2. Vector-Quantized Bottleneck

The vector quantized bottleneck quantizes the outputs of the encoding layer using a set of codebooks. Where a separate codebook, constant over the entire input, is used to quantize the speaker embedding. We represent a codebook containing $K$ codes by $C = \{e_1, \cdots e_K\}$. In the forward pass, the encoder outputs $E = \{e_1, \cdots e_K\}$ are quantized by replacing each $e_i$ with the closest $c_j$ to get the quantized encoding $\hat{e}_i$, where $j = \text{arg min}_k ||e_i - c_k||^2_2$. Due to the non-differntiability of the $\text{arg min}$ operation, VQ-VAE uses an additional two loss terms. The terms encourage each encoding $e_i$ to be close to the selected $c_j$ and for each code $c_j$ to minimize the quantization error incurred by any encodings that selected it. These are summarized below using the stopgradient operator $\text{sg}$ which is identity at the forward pass but stops gradients in the reverse pass. In practice, we optimize the second term using an exponential moving average $k$-means. For additional details, see [9].

$$\mathcal{L}_{\text{vq}} = \lambda \|\text{sg}[\hat{E} - E]\|^2_2 + \|\text{sg}[E] - \hat{E}\|^2_2.$$  \hspace{1cm} (2)

2.1.3. Autoregressive Decoder

We use an RNN based model to synthesize raw 16 kHz audio. The model, which is described in [5], contains two Gated Recurrent Units (GRU) and two dense layers. We first concatenate the quantized speaker embedding to the quantized encoding and pass the resulting tensor through the first GRU. Then, we up-sample the GRU output to match the desired output length (in raw audio samples) and pass the upsampled tensor through the second GRU, and two final dense layers. We apply softmax to the final dense layer and train the model to predict a distribution over 8-bit mu-law quantized values.

2.1.4. Loss

The final forward pass procedure is composed of passing raw noisy audio $x$ to the encoder, quantizing the resulting encodings and speaker embedding, and running an autoregressive model to produce an estimated clean waveform $\hat{s}$. The full loss function shown below is composed of $\mathcal{L}_{\text{vq}}$ and a cross-entropy term $\mathcal{L}_{\text{ce}}(s, \hat{s})$, which measures the KL-divergence between the predicted distribution and the one-hot value of the mu-law quantized clean speech $s$.

$$\mathcal{L} = \mathcal{L}_{\text{vq}} + \mathcal{L}_{\text{ce}}(s, \hat{s}).$$  \hspace{1cm} (3)

3. EXPERIMENTS AND RESULTS

We train the Codec Only and Enhancing Codec models at two different kb/s and for the highest kb/s Enhancing Codec we also experiment with modifying the training setup to use higher SNR mixtures.

3.1. Model Details

The encoder first computes an 80 bin log-mel representation with a hop-size of 10ms and a window size of 250ms on 16kHz audio. These are passed to the speech encoder which has five convolutional layers each with 768 filters. The first, second, fourth and fifth layers use a stride of 1 and a kernel
3.2. Datasets

3.2.1. Training

To generate a training mixture, we retrieve clean speech data from the LibriSpeech dataset in [18], and noise data from AudioSet [19]. When selecting noise clips from AudioSet, we avoid any clips with speech related tags. To increase the prevalence of challenging noise we sample noise clips with non-stationary noise more frequently. The noisy mixtures are created with a random SNR between $-5$ and $25$ dB. Finally, all room impulse responses are synthetically generated using the image-source method. For additional details consult [16].

3.2.2. Evaluation

To evaluate, we use the test mixtures from the VCTK dataset [20]. It contains mixtures with SNRs of $2.5$dB, $7.5$dB, $12.5$dB and $17.5$dB across a variety of speakers and noises. When evaluating our models on clean speech we use the clean speech samples from the VCTK test set.

3.3. Subjective Quality Evaluation

Since the compression based models in this paper resynthesize waveforms, their performance is not aptly measured by standard numerical metrics; we therefore measure model performance using a Mean Opinion Score (MOS) from a crowd-sourced study on Mechanical Turk that uses the P808 evaluation method [21].

3.4. Results

We compare the compressor-enhancer model with a compressor-only counterpart of identical architecture and size at bandwidths of $1.35$ kb/s and $0.9$ kb/s. The compressor only model is trained on the clean speech setup described in section 3.2.1. We also experiment with the combination of a speech enhancement model (RNNoise) [17] and the compression only model. We refer to these baselines as “Codec Only”, and “Enhancement, then codec” respectively. As a final point of comparison we also evaluate a pre-trained LPCNet vocoder [6].

3.4.1. MOS vs Bandwidth in Noisy Speech

Fig. 2 displays the MOS scores of the models at different bandwidths. Within the examined bandwidths, our enhancement-trained compression model (blue diamonds and circle) scores about .6 MOS above its non-enhancing counterpart (black square). Interestingly, sequential speech enhancement and compression scores worse than compression alone. We suspect this stems from the compression only model being susceptible to out of distribution errors. Finally, the LPCNet model serves as another comparison point for a compression only neural codec. The model denoted as “Enhancing Codec LN” is identical to “Enhancing Codec” in

![Block Diagram of the compressor-enhancer. The speech and speaker encoders are made up of several convolutional layers with batch normalization and ReLU. The VQ bottleneck has separate quantizers for the speech and speaker encodings. In our experiments the speech quantizer is made up of several codebooks of size 512 and the speaker quantizer is made up of a single codebook of size 512. The decoder is a WaveRNN based model and uses the quantized speech and speaker information first to reconstruct a coarse and then reconstruct a fine waveform. The waveform output is mu-law quantized.](image-url)
Fig. 2. MOS scores on the VCTK test set compared across models running at a range of kb/s. Our proposed joint compressor-enhancer models outperform the compression only baselines as well as the sequential enhancement-compression baselines. The LN suffix denotes a model trained on mixtures with lower noise content. The results are statistically significant with 95% confidence intervals of approximately 0.03.

Table 1. MOS comparison across SNRs on VCTK at 1.35 kb/s. The largest performance difference is in low SNR scenes.

<table>
<thead>
<tr>
<th>Model</th>
<th>SNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Codec Only</td>
<td>2.89</td>
</tr>
<tr>
<td>Enhancing Codec LN</td>
<td>3.27</td>
</tr>
<tr>
<td>Enhancement, then Codec</td>
<td>2.77</td>
</tr>
</tbody>
</table>

omit the sequential enhancement then compression model since the speech is already clean. The enhancement-trained model outperforms the compression only model, leading us to suspect that training with noise helps the model learn a more robust bottleneck, and thus generalize better.

3.5. Comparison with a two-stage approach

We also attempted a joint compression and enhancement approach where we trained, first, a compression only model and then an encoder-only enhancement model trained to, given noisy speech, output the discretized latent representation of clean speech. With the goal being improvements to out-of-domain errors for speech enhancement models, as a two-stage approach would mean a fewer number of parameters needing to be trained on enhancement. However, we found that these models do not perform well, possibly because the clean-trained autoregressive decoder is too sensitive to out-of-domain inputs from the enhancer-encoder, making the simultaneous training of the decoder a key component of joint compression and enhancement.

4. CONCLUSION

In this work we presented a model that does joint compression and enhancement of a noisy speech signal using a VQ-VAE with a convolutional encoder and a WaveRNN decoder. Through a set of mean opinion score based experiments, we found that joint compression and enhancement performs better in the presence of noise, including in low SNR scenarios, than stand-alone compression; and also outperforms a sequential combination of speech enhancement and a compression only neural codec. We also found that enhancement training improves codec performance on clean speech signals.

5. REFERENCES


