Source-Aware Neural Speech Coding for Noisy Speech Compression

Haici Yang\textsuperscript{1}, Seungkwon Beack\textsuperscript{2}, Minje Kim\textsuperscript{1},

\textsuperscript{1}Indiana University, Department of Intelligent Systems Engineering, Bloomington, IN, USA
\textsuperscript{2}Electronics and Telecommunications Research Institute, Daejeon, South Korea
hy1@iu.edu, skbeack@etri.re.kr, minje@indiana.edu

Abstract

This paper introduces a novel neural network-based speech coding system that can handle noisy speech effectively. The proposed source-aware neural audio coding (SANAC) system harmonizes a deep autoencoder-based source separation model and a neural coding system, so that it can explicitly perform source separation and coding in the latent space. An added benefit of this system is that the codec can allocate different amount of bits to the underlying sources, so that the more important source sounds better in the decoded signal. We target the use case where the user on the receiver side cares the quality of the non-speech components in the speech communication, while the speech source still carries the most important information. Both objective and subjective evaluation tests show that SANAC can recover the original noisy speech in a better quality than the baseline neural audio coding system, which is with no source-aware coding mechanism.

Index Terms: speech enhancement, speech coding, source separation

1. Introduction

Breakthroughs made in the fields of deep learning for the past decade have shown phenomenal performance improvements in various pattern recognition tasks, including media compression and coding. Seminal work was proposed in the lossy image compression domain, where deep autoencoders were employed. Autoencoders are a natural choice as their encoder part converts the input signal into a latent feature vector, followed by the decoder that recovers the input from the latent space \cite{1, 2}. The compression is achieved if the total number of bits to represent the latent feature (or code) is smaller than that of the raw input signal. Although the computational complexity of the encoding and the decoding processes is still more complex than the traditional coding systems, the deep autoencoding systems did show superior compression performance to the traditional technology.

Neural speech coding is an emerging research area, too. Autoregressive models, such as WaveNet \cite{3}, have shown a transparent perceptual performance in a very low bitrate \cite{4, 5}, surpassing the performance of the traditional coders at the same bitrate. Another branch of neural speech coding systems take a frame-by-frame approach, where a time-domain waveform signal is fed to an end-to-end autoencoding network. Kankanahalli proposed a simpler model consists of fully convolutional layers to control dimension reduction, quantization, and entropy control tasks \cite{6}. Cross-module residual learning (CMRL) inherited the convolutional pipeline, and proposed a cascading structure, where multiple autoencoders are concatenated to work on the residual signal produced by the preceding ones \cite{7}. CMRL is then harmonized with trainable linear predictive coding (LPC) module \cite{8}. It further improved the performance and lowered the model complexity down to 0.45 million parameters.

These waveform codecs were able to outperform AMR-WB \cite{9}.

In this work we widen the scope of applications of speech codec by taking into account the noisy speech as input. While most of the speech coding systems are designed to work only for clean single-talker speech, the real-world speech signals are often accompanied by additional sound sources, which needs to be addressed, too. However, traditional speech codecs are based on the speech production models \cite{10, 11}, thus lacking the ability to model the non-speech components mixed in the input signal. This kind of ideas have been partly reflected in the MPEG unified speech and audio coding (USAC) standard \cite{11, 12}. USAC explicitly tackles speech signals in the mixture condition by switching between different tools defined for different kinds of signals, such as speech and music. However, the switching decision was on every frame rather than addressing the fact that mixing happens within the frame. Meanwhile, AMR-WB’s discontinuous transmission (DTX) mode also considers the mixed nature of input speech by deactivating the coding process for the non-speech periods \cite{9}. Lombard et al. improved DTX by generating artificial comfort noise that smooths out the discontinuity \cite{13}. However, for the frames where both speech and non-speech sources co-exist, it is difficult to effectively control the bitrate using DTX. Similar ideas have been used in transform coders for audio compression, where the dynamic bit allocation algorithm based on psychoacoustic models can create a spectral hole in low bitrate cases. Intelligent noise gap filling can alleviate the musical noise generated from this quantization process \cite{14, 15}, while it is to reduce the artifact generated from the coding algorithm, rather than to model the non-stationary noise source separately from the main source.

In this work we propose source-aware neural audio coding (SANAC) to control the amount of bits for each source differently. We begin with the noisy speech signals, since the speech source is obviously more interesting than the other noise sources. However, SANAC does not seek a speech-only reconstruction, e.g., by denoising the noisy input while coding it simultaneously \cite{16}. We target the use case where the user still wants the code to convey the non-speech components for a better understanding of the transmitter’s acoustic environment. Moreover, we also show that the sources in the mixture can be assigned with unbalanced bitrates depending on their perceptual or applicational importance as well as their entropy in the latent space, leading to a better perceptual quality.

2. Model Description

The proposed SANAC system harmonizes a source separation module into the neural coding system. Our model performs explicit source separation in the coded space to produce source-specific feature maps, which are subsequently quantized and decoded back to the sources, respectively. The source-specific code vectors can be learned using a masking-based approaches.
as in TasNets [17][18], while we propose to utilize the orthogonality assumption between the source-specific code vectors, because in that way we can drop the separator module in the TasNet architecture and reduce the encoder complexity. SANAC also employs soft-to-hard quantization [2] to quantize these source-specific codes, which are then decoded source-wise, too. The architecture is carefully designed so that the quantization process and the bitrate control on it can work on individual sources independently.

2.1. Orthogonal code vectors for separation

As a source separation system, the model consists of an encoder that converts a time-domain mixture frame \( x \in \mathbb{R}^N \) into a code vector \( z \in \mathbb{R}^D \): \( z \leftarrow \mathcal{F}_{\text{enc}}(x) \). We assume that there are \( K \) mask vector \( m^{(k)} \in \mathbb{R}^D \) that can decompose the code vector into \( K \) components: \( \sum_{k=1}^{K} m^{(k)} = 1 \). Note that the mask values are probability over the \( K \) sources.

We further assume that this probabilistic code assignments to the sources are a hard decision so that the masking process assigns each code value to only one source, i.e.,

\[
m_{d}^{(k)} \left\{ \begin{array}{ll} 1 & \text{if } \arg \max \alpha, m^{(j)} = k \\
0 & \text{otherwise}
\end{array} \right.
\]

In the TasNet architectures, similar masking vectors are estimated via a separate neural network module, which led to the state-of-the-art separation performance. In there, the estimated mask values are indeed somewhat drastically distributed to either near zero or one making the masked code vectors nearly orthogonal from each other, although its sigmoid-activated masks near zero or one making the masked code vectors nearly or-

2.2. Soft-to-hard quantization

Quantization is a mapping process that replaces a continuous variable into its closest discrete representative. Since it is not a differentiable process, incorporating it in a neural network requires a careful consideration. Soft-to-hard quantization showed successful performance both in image and speech compression models [4][6][2]. The idea is to formulate this cluster assignment process as a softmax classification during the feedforward process, which finds the nearest one among the \( M \) total representatives \( \mu_{m} \), for the given code vector \( y \) as follows:

\[
d_{m} = \mathcal{E}(y\|\mu_{m}), \quad p = \text{Softmax}(\alpha d),
\]

Testing: \( y = \mu_{\arg \max_{m} p_m} \). Training: \( y = \sum_{m=1}^{M} p_m \mu_{m} \),

where the algorithm first computes the Euclidean distance vector \( d \) against all the representatives (i.e., the cluster means), whose negative value works like a similarity score for the softmax function. Using the softmax result, the probability vector \( p \) of the cluster membership, we can construct the quantized code vector \( \hat{y} \). During test time, simply choosing the closest one will do a proper quantization. However, since this arg max operation is not differentiable, for training we do a convex combination of the cluster centroids to represent the quantized code. The discrepancy between training and testing is reduced by controlling the scaling hyperparameter \( \alpha \), which makes the softmax probabilities more drastic (e.g., a one-hot vector in the extreme case) once it is large enough. Note that it also learns the cluster centroids \( \mu \) as a part of the learnable network parameters rather than employing a separate clustering process to define them.

In previous work the quantization has been on scalar vari-
ables, i.e., \( y \in \mathbb{R}^1 \). In this work, the soft-to-hard quantization performs vector quantization (VQ). We denote the CNN encoder output by \( Z \in \mathbb{R}^{L \times P} \), which consists of \( K = 2 \) code blocks: \( Z = [Z^{(1)}, Z^{(2)}] \). Then, each code vector for quantization is defined by the \( p \)-th feature out of \( P \) total features spanning over \( L \) channels: \( y = Z_{k}^{(p)} \), having \( L = 6 \) as the VQ dimension in our case.

2.3. Source-wise entropy control

The theoretical lower bound of the bitrate, as a result from Huffman coding, can be defined by the entropy of the quantized codes. The frequency of the cluster means defines the entropy of the source-specific codes: \( \hat{H}(\mu^{(k)}) = - \sum_{m=1}^{M} q_m^{(k)} \log q_m^{(k)} \), where \( q_m^{(k)} \) denotes the frequency of \( m \)-th mean for the \( k \)-th source. Meanwhile, the entropy for the code of the mixture signal is smaller than or equal to the sum of the entropy of all sources: \( \hat{H}(\tilde{x}) = \sum_{k=1}^{K} \hat{H}(\mu^{(k)}) \). The \( \tilde{x} \) is the mixture vector learned directly from the mixture signals. Therefore, in theory, the proposed source-specific coding scheme cannot achieve a better coding gain than a codec that works directly on the mixture.

However, SANAC can still benefit from the source-wise coding, especially by exploiting the perceptual factors. As our main assumption in this work is that the perceptual importance differs by the sources, we envision a coding system that is able to assign different bitrates to different sources. For noisy speech, for example, we will try to assign more bits to the speech source. Consequently, although the user eventually listens to the recovered mixture of speech and noise \( \hat{x} \) (a) the perceptual quality of the speech component is relatively higher than that of the noise source reconstruction \( \hat{n} \)

This argument is based on the code's ability to control the entropy of the source-specific codes. In SANAC, we adopt the entropy control mechanism proposed in [2], but by setting up a per-source loss between the target \( \xi_{\text{tot}} \) and the actual entropy values: \( (\xi_{(k)} - \hat{H}(\mu^{(k)}))^2 \). While this loss does not guarantee the exact bitrate during the test time, in practice, we observe that the actual bitrate is not significantly different from the target.

2.4. Decoding and the final loss

The source-specific truncated codes, after the quantization, \( Z^{(k)} \), is fed to the decoder part of the network. The decoder function works similar to Conv-TasNet [18] in that the decoder runs \( K \) times to predict \( K \) individual source reconstructions by taking \( K \) source-specific feature maps as the input. However, SANAC is different from Conv-TasNet's decoding due to the fact that decoder input features are quantized codes. In addition, we found that having an additional feature transformation block helps the decoding performance before calling the individual decoders (the “Transformation” module in Figure 1). Finally, our model cares the quality of the recovered mixture, not only the separation quality.

Considering all these goals, our training loss consists of the ordinary mean squared error (MSE) in the time domain and the entropy control terms. More specifically, for the noisy speech case \( x = s + n \) (\( k = 1 \) for speech and \( k = 2 \) for noise), the MSE loss is for the speech source reconstruction \( \hat{s} \) and the mixture reconstruction \( \hat{x} \), while the noise source reconstruction \( \hat{n} \) is implied in there. We regularize the total entropy as well as the ratio between the two source-wise entropy values:

\[
\mathcal{L} = \lambda_{\text{MSE}} (\mathcal{E}_{\text{MSE}}(s|\hat{s}) + \mathcal{E}_{\text{MSE}}(x|\hat{x})) \\
+ \lambda_{\text{EntTot}} (\mathcal{E}_{\text{MSE}}(s|\hat{s}) + \mathcal{E}_{\text{MSE}}(x|\hat{x})) \\
+ \lambda_{\text{Entropy}} (\mathcal{H}(\mu^{(1)}) - \mathcal{H}(\mu^{(2)}))^2 \\
+ \lambda_{\text{Ratio}} (\psi - \frac{\mathcal{H}(\mu^{(1)})}{\mathcal{H}(\mu^{(2)})})^2,
\]

where \( \xi \) and \( \psi \) are the target total entropy and the target ratio, respectively.

3. Experiment

3.1. Dataset

500 and 50 utterances are randomly selected from TIMIT corpus [19], and then contaminated by ten non-stationary noise sources, \{bird singing, casino, cicadas, typing, chip eating, frogs, jungle, machine gun, motorcycle, ocean\} used in [20]. Every noisy speech waveform is segmented into frames of 512 samples (32ms), with overlap of 64 samples. We apply a Hann window of size 128 samples only to the overlapping periods. Since there are 16000/448 frames per second and each frame produces \( P \) code vectors for VQ, for the entropy of a source-specific codebook \( \xi \), the bitrate is \( 16000P\xi/488 \), e.g., 9.14kpbs when \( P = 256 \) and \( \xi = 1 \).

3.2. Training Process

Adam optimizer with an initial learning rate 0.0001 trains the models [21]. Both SANAC and the baseline are trained in three stages. Every jump to next stage is triggered when the validation loss stops improving in 3 consecutive epochs. We stop the training updates after validation loss does not improve for 10 epochs.

- **Stage 1**: For the first three epochs, the model learns to denoise without any coding in between. This stage is to learn decent initialization of quantization vector centroids for the two sources. For the encoder, we employ a few bottleneck blocks that are commonly used in ResNet [22], which is a small convolutional autoencoder that reduces the depth of the input feature map from 30 to 10, and then recover back to 30 for the output. That way, the input and output feature maps can be connected via an identity shortcut. The encoder module also employs a 1-d convolutional layer to downsamples the feature map from 512 to 256, followed by another bottleneck block and channel changing layer to yield two sets of code map of 6 \( \times \) 256 each. In this stage, we do not perform the soft-to-hard quantization yet. Instead, we directly feed each source-specific code map to the channel changer and upsample, to prepare the original 30 \( \times \) 512 feature map for the final decoding. The upsampling layer interlaces two adjacent channels into one, doubling the number of features up to 512 while halving the number of channels. It is a sub-pixel convolution technique introduced in [23], which showed its merit in speech coding, too [6][7]. Finally, the feature maps go through a few more ResNet blocks that strengthens the separation (the “Transformation” module in Figure 4), followed by the source-wise decoders, which share the same parameters with each other. The “Transformation” module and the decoder consist of two bottleneck blocks with different input channels, 60 and 30, respectively.

- **Stage 2**: In this stage the model starts to quantize the encoder output using the soft-to-hard VQ mechanism. The VQ is done
with $M = 128$ centroids. We set the scale of softmax function $\alpha = 10$, and increase it exponentially until it reaches 500. Meanwhile, the other modules in the network are also updated accordingly to absorb the quantization error.

- **Stage 3**: As Stage 2 stabilizes, we introduce entropy control terms into the loss function by setting up the regularization weights $\lambda_{\text{EntTot}} = 1/5$ and $\lambda_{\text{Ratio}} = 1/60$. We set $\xi$ to be 1, 2, and 3, which correspond to three bitrates 9.14, 18.29, and 27.43 kbps. The target ratio between speech and noise bitrates is set to be $\psi = 3$.

Our baseline system is similar to the proposed architecture, except that (a) encoder produces $L \times F$ code map (b) there is no control of entropy ratio as there is only one kind of codes for the mixture (c) the decoder runs only once to recover the mixture (d) the loss for speech source reconstruction does not exist.

3.3. Objective Evaluation

We evaluate the model based on the scale-invariant signal-to-distortion ratio (SISDR) [24] and short-time objective intelligibility (STOI) [25] for both reconstructed speech and mixture signals. For speech in particular, we report SISDR improvement against the 0 dB and 5 dB input mixture. Figure 3 shows that our model outperforms the baseline model on both mixture and speech reconstruction in most of the case, except for the 5 dB input. More specifically, Figure 2c shows that the speech source in the recovered mixture is more intelligible than the baseline, thanks to the more bits assigned to the speech codes. This trend is more prominent in the lower bitrates. When it comes to the speech reconstruction shown in Figure 2b, we see similar trend in the 5dB input SNR case, while for 0 dB, the imperfect separation quality starts to degrade the intelligibility of the proposed SANAC system. SANAC showed better mixture reconstruction performance in terms of SISDR, too (Figure 2c) and the SISDR improvement for the speech source reconstruction (Figure 2d). In both SISDR experiments, the performance gap is more salient in 5dB input SNR cases. Overall, we note that SANAC clearly outperforms the baseline in the lower bitrates, if the noise intervention is not too severe.

3.4. Subjective Evaluation

Eight audio experts participated in the subjective test on the perceptual quality of mixture reconstructions. From a randomly shuffled pair of SANAC and the baseline examples, listeners are asked to choose one that sounds more similar to the reference signal, which is the corresponding uncompressed input mixture. The test consists of three sessions with three different bitrates. Each session has 10 trials with 10 gender-balanced utterances contaminated by 10 distinct noise types. Y-axis in Figure 3 indicates the number of samples that are considered preferable in the given session per subject. The results are in line with the STOI score, where people are more fond of SANAC in low bitrate cases. However, note that we did not specifically asked the subjects to evaluate based on the intelligibility, but to compare the decoded signals to the reference. We also observe that, in the case of high bitrate, the test results are dependent on the noise types, potentially due to the different denoising results.

4. Conclusion

In this work, we proposed SANAC for source-aware neural speech coding. In the model, we harmonized a Tasnet-like masking-based separation approach into an end-to-end neural coding network and subsequently perform quantization on the source-specific codes. SANAC showcased superior performance in both objective and subjective tests to the baseline model with a similar architecture except for source-specific coding. We believe that SANAC opens a new possibility of widening audio coding on mixture signals by being able to control individual sources differently. The sound examples and source code are available online.

5. Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (2017-0-00072, Development of
6. References

[1] L. Theis, W. Shi, A. Cunningham, and F. Huszar, “Lossy image compression with compressive autoencoders,” 2017.

[2] E. Agustsson, F. Mentzer, M. Tschanzen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, “Soft-to-hard vector quantization for end-to-end learning compressible representations,” in Advances in Neural Information Processing Systems (NIPS), 2017, pp. 1141–1151.

[3] A. van den 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.

[4] W. B. Kleijn, F. S. C. Lim, A. Luebs, J. Skoglund, F. Stimberg, Q. Wang, and T. C. Walters, “WaveNet based low rate speech coding,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2018, pp. 676–680.

[5] J.-M. Valin and J. Skoglund, “A real-time wideband neural vocoder at 1.6 kbps using LPCNet,” in Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech), 2019.

[6] S. Kankanhalli, “End-to-end optimized speech coding with deep neural networks,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2018.

[7] K. Zhen, J. Sung, M. S. Lee, S. Beack, and M. Kim, “Cascaded cross-module residual learning towards lightweight end-to-end speech coding,” in Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech), 2019.

[8] K. Zhen, M. S. Lee, J. Sung, S. Beack, and M. Kim, “Efficient and scalable neural residual waveform coding with collaborative quantization,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2020.

[9] B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Jarvinen, “The adaptive multirate wideband speech codec (AMR-WB),” IEEE Transactions on Speech and Audio Processing, vol. 10, no. 8, pp. 620–636, 2002.

[10] M. Schroeder and B. Atal, “Code-excited linear prediction (CELP): High-quality speech at very low bit rates,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 10, 1985, pp. 937–940.

[11] ISO/IEC DIS 23003-3, “Information technology – mpeg audio technologies – part 3: Unified speech and audio coding,” 2011.

[12] ISO/IEC 14496-3:2009/PDAM 3, “Transport of unified speech and audio coding (USAC),” 2011.

[13] A. Lombard, S. Wilde, E. Ravelli, S. Dhla, G. Fuchs, and M. Dietz, “Frequency-domain comfort noise generation for discontinuous transmission in evs,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2015, pp. 5893–5897.

[14] J. Herre, J. Hilpert, A. Kunz, and J. Plogsties, “Mpeg-h audio—the new standard for universal spatial/3D audio coding,” Journal of Audio Engineering Society, vol. 62, no. 12, pp. 821–830, 2015.

[15] S. Dirsch, A. Niedermeier, C. R. Helmrich, C. Neukam, K. Schmidt, R. Geiger, J. Leconte, F. Glode, F. Nagel, and B. Adler, “Intelligent gap filling in perceptual transform coding of audio,” in Audio Engineering Society Convention 141, Sep. 2016.

[16] F. S. C. Lim, W. Bastiaan Kleijn, M. Chinen, and J. Skoglund, “Robust low rate speech coding based on cloned networks and wavenet,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2020, pp. 6769–6773.

[17] Y. Luo and N. Mesgarani, “Tasnet: time-domain audio separation network for real-time, single-channel speech separation,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2018.

[18] ——, “Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 27, no. 8, pp. 1256–1266, 2019.

[19] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, “TIMIT acoustic-phonetic continuous speech corpus,” Linguistic Data Consortium, Philadelphia, 1993.

[20] Z. Duan, G. J. Mysore, and P. Smaragdis, “Online PLCA for real-time semi-supervised source separation,” in Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA), 2012, pp. 34–41.

[21] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[23] W. Shi, J. Caballero, F. Huszâr, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1874–1883.

[24] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “SDR - half-baked or well done?” arXiv preprint arXiv:1811.02508, 2018.

[25] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, “A short-time objective intelligibility measure for time-frequency weighted noisy speech,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2010.