



Unsupervised Learning of Disentangled Speech Content and Style Representation

Andros Tjandra^{1,*}, Ruoming Pang², Yu Zhang^{2,3}, Shigeki Karita³

¹ Nara Institute of Science and Technology, Japan

² Google Research, Brain Team, USA

³ Google LLC, Japan

andros.tjandra@gmail.com, {rpang,ngyuzh,karita}@google.com

Abstract

Speech is influenced by a number of underlying factors, which can be broadly categorized into linguistic contents and speaking styles. However, collecting the labeled data that annotates both content and style is an expensive and time-consuming task. Here, we present an approach for unsupervised learning of speech representation disentangling contents and styles. Our model consists of: (1) a local encoder that captures per-frame information; (2) a global encoder that captures per-utterance information; and (3) a conditional decoder that reconstructs speech given local and global latent variables. Our experiments show that (1) the local latent variables encode speech contents, as reconstructed speech can be recognized by ASR with low word error rates (WER), even with a different global encoding; (2) the global latent variables encode speaker style, as reconstructed speech shares speaker identity with the source utterance of the global encoding. Additionally, we demonstrate a useful application from our pre-trained model, where we can train a speaker recognition model from the global latent variables and achieve high accuracy by fine-tuning with as few data as one label per speaker.

Index Terms: Unsupervised, Disentanglement, Speech, VQ-VAE

1. Introduction

A speech signal is a one-dimensional time-domain representation where each value represents the sound intensity across different time-step. Behind its simple representation, there is a lot of high-level information such as phoneme, tone, emotion, gender, and speaker are mixed underneath. Learning disentangled latent representations from speech has a wide set of applications in generative tasks, including speech synthesis, data augmentation, voice transfer, and speech compression. Downstream tasks such as speech recognition [1] and speaker classification [2] can also benefit from such learned representations.

Because of the cost, complexity, and privacy concerns around collecting labeled speech data, there has been a lot of interest in unsupervised representation learning for speech. Of particular interest is to learn representations for speech styles from unsupervised data due to the difficulty in describing prosody with human labels.

Some previous works aim to learn *global* representations from entire speech sequences. Global style tokens [3] learn a dictionary of embeddings from speech without prosody labels. As another example, Hsu et al. [4] model disentangled speech styles with a hierarchy of variational autoencoder (VAE) [5].

Hu et al. [6] proposed a content and style separation model by pre-training on a single-speaker dataset with text transcription and minimizing mutual information (MI) between the content and style representation. Unfortunately, all of those approaches required datasets with text labels, either in the training or inference stage.

Other works try to learn fine-grain *localized* representations of speech. [2, 7, 8] apply self-supervised learning to unlabeled speech data and extract localized latent representations that can be fine-tuned for speech recognition. Those models produce high-quality content representation over time-step, but the global representation is not captured or disentangled by the model itself.

Factorized Hierarchical Variational Autoencoder (FH-VAE) [9] learns a sequence of high-level and low-level features by utilizing two-level hierarchical Gaussian VAE. However, the Gaussian VAE-based model has drawbacks such as the evidence lower-bound (ELBO) trade-off between model generation quality and enforcing simple posteriors [10]. There are also several works based on the generative adversarial network (GAN) model [11]. For example [12, 13] used CycleGAN [14] to perform voice conversion without any paired data. While GAN-based models are powerful, they can be hard to train and require careful hyperparameter tuning as well [15, 16]. [17, 18, 19] leverages vector-quantized VAE [20] (VQVAE) to learn a discrete sequence representation of speech. Still, both approaches require the speaker information given during the training stage.

We propose a simple framework to learn both global and localized representation of speech. In order to disentangle the content and style representations, we apply (1) a local encoder with a VQ layer to learn a discrete per-timestep representation of the speech that captures the linguistic contents and (2) a global VAE to extract per-utterance representations to reflect the speech styles. We further disentangle the local and global representations with a mutual information loss [21]. We evaluate the quality of linguistic and style representations by running speech and speaker recognition models on the reconstructed speech. We also show that the global representation captures the speaker information well enough that we can obtain a speaker classification model by training a linear projection layer on top of the global representation with only one example per speaker.

2. Model architecture

Our proposed model is illustrated in Figure 1. On the encoder side, we have two different encoders with different architectures, encoding content and style respectively. On the decoder side, we have a decoder that takes both content and style latent variables to produce speech features.

* work done during internship at Google Brain

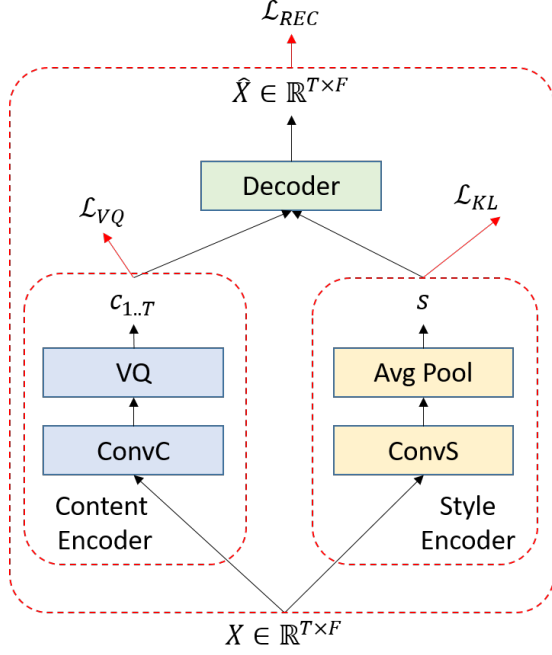


Figure 1: Proposed model: disentangling content and style auto-encoder.

2.1. Content Encoder

We represent the speech feature as a matrix $X \in \mathbb{R}^{T \times F}$ where T is the frame time-step and F is the feature dimension. The content encoder $EncC(\cdot)$ aims to extract a sequence of latent variables $C = [c_1, \dots, c_T] \in \mathbb{R}^{T \times D_C}$ that only represent the content from the input speech X . It is built on top of several convolutional layers to represent the content information on the relatively short window time-step.

On the top of the content encoder, we apply information bottleneck via vector quantization (VQ) [20] with straight-through gradients. By limiting the number of output possibilities from the content encoder, we force the content encoder and VQ to capture only necessary localized information such as phoneme or subword-like representation and discard their variations (e.g., the speaking style variation). Without using an information bottleneck or strong regularization method such as VQ layer, the content encoder might capture both content and style inside $c_{1..T}$ and the decoder will ignore the style latent variable s from the style encoder.

2.2. Style Encoder

The style encoder $EncS(\cdot)$ aims to extract a global latent variable as a vector $s \in \mathbb{R}^{D_S}$ that only represent the speaking style from the input speech X . It is built on top of several convolutional layers. We summarize the output from the convolutional layer into a single vector with a global average pooling operation across the time-axis. We put a variational layer with Gaussian posterior [5] on the top of convolutional layer output. During training, the style variable s is sampled from Eq. 3. In the inference step, we directly use s_μ to represent the style information.

2.3. Decoder

The decoder $Dec(\cdot, \cdot)$ takes both content and style latent variables $[c_{1..T}, s]$ and reconstruct the features $\hat{X} \in \mathbb{R}^{T \times F}$ during training stage. For the inference step, we can combine $c_{1..T}$ and s from different input speech and generate a new speech feature that has same content as C and same style as s .

2.4. Loss

We optimize our model with multiple loss functions:

1. **Reconstruction loss:** $\mathcal{L}_{REC} = \|X - \hat{X}\|_1^2 + \|X - \hat{X}\|_2^2$. The reconstruction loss minimizes both L1 and L2-norm squared distance between groundtruth X and \hat{X} .

2. **Content VQ loss:**

$$i = \arg\min_j \|ConvC(X) - E_j\|_2 \quad (1)$$

$$\mathcal{L}_{VQ} = \|ConvC(X) - sg(E_i)\|_2^2 \quad (2)$$

where $sg(\cdot)$ is stop gradient operations. The content VQ loss encourages the encoder output z to minimize the distance between itself and the nearest codebook E_i .

3. **Style regularization loss:**

$$s_\mu, s_\sigma^2 = EncS(X); \quad s \sim \mathcal{N}(s_\mu, s_\sigma^2) \quad (3)$$

$$\mathcal{L}_{KL} = \mathbb{D}_{KL}(\mathcal{N}(s_\mu, s_\sigma^2) \parallel \mathcal{N}(0, 1)) \quad (4)$$

The style regularization loss minimizes the KL divergence between the Gaussian posterior $\mathcal{N}(s_\mu, s_\sigma^2)$ with the unit Gaussian prior $\mathcal{N}(0, 1)$.

The final loss is a combined sum:

$$\mathcal{L} = \mathcal{L}_{REC} + \gamma \mathcal{L}_{VQ} + \mathcal{L}_{KL}. \quad (5)$$

2.5. Mutual Information Estimation and Minimization

Section 2.1 describes the importance of information bottleneck via VQ layer on the content encoder sides. However, the quantized variables $c_{1..T}$ may still manage to carry non-content information such as speaking style.

To further decouple content and style representations without explicit labels, we estimate and minimize the mutual information (MI) from the output between content $c_{1..T}$ with style s . By minimizing the mutual information, we reduce the amount of correlation between two random variables [22].

Measuring mutual information directly from high dimensional random variables is a difficult problem [23]. Here, we use InfoNCE [2] to estimate the lower bound MI between content and style:

$$C_i = AvgPool(ConvC(X_i)) \quad (6)$$

$$S_i = AvgPool(ConvS(X_i))$$

$$I(C; S) \geq \mathbb{E} \left[\frac{1}{K} \sum_{i=1}^K \log \frac{e^{Sc(C_i, S_i)}}{\frac{1}{K} \sum_{j=1}^K e^{Sc(C_i, S_j)}} \right] \triangleq I_{NCE}.$$

where K is the number of samples (across different utterances), $Sc(\cdot, \cdot)$ is a neural network scorer, C_i is the representation from content encoder prior to the quantization, and S_i is the representation from style encoder. The neural network scorer $Sc: \mathbb{R}^{D_C} \times \mathbb{R}^{D_S} \rightarrow \mathbb{R}$ is used to model the density ratio

$$Sc(C_i, S_i) \propto \frac{p(C|S)}{p(C)} \quad (7)$$

between content vector C_i and style vector S_i .

Finally, we simultaneously try to maximize I_{NCE} w.r.t S_C and minimize I_{NCE} and \mathcal{L} w.r.t $(\theta = \{EncC, EncS, Dec\})$:

$$\max_{S_C} \min_{EncC, EncS, Dec} (\mathcal{L} + I_{NCE}). \quad (8)$$

We apply adaptive gradient scaling [21] to stabilize the loss Eq. 8 during training. For each step, the model parameters are updated with:

$$g_\theta = \partial \mathcal{L} / \partial \theta; \quad g_a = \partial I_{NCE} / \partial \theta \quad (9)$$

$$g_b = \min(\|g_a\|_2, \|g_\theta\|_2) \frac{g_a}{\|g_a\|_2} \quad (10)$$

$$\theta \leftarrow \theta - (g_\theta + g_b) \quad (11)$$

$$S_C \leftarrow S_C + \partial I_{NCE} / \partial S_C \quad (12)$$

3. Experiments

3.1. Dataset

To evaluate our proposed model effectiveness on disentangling content and style, we use the Librispeech [24] corpus for all following experiments. To train the generative model, we use all training sets: train-clean 100 hours (contains 251 unique speakers), train-clean 360 hours (contains 921 unique speakers) and train-other 500 hours (contains 1166 unique speakers). We use dev-clean (contains 40 unique speakers) as the development set and test-clean (contains 40 unique speakers) for evaluation. In total, Librispeech contains 2484 unique speakers without overlap between different sets. For the feature extraction, we extract 80 dimensions log Mel-filterbank with window size 25 ms and step size 10 ms. All features are normalized into zero mean and unit standard deviation based on the statistics from the entire dataset.

3.2. Hyperparameters

For the content encoder, we have 10 1D convolutional layers with 768 dimension and residual connections. We use time stride 2 in the 3rd layer. After we fed the input X into the content encoder, the final output length was compressed from T to $T/2$.

For the style encoder, the input X is processed by 6 residual 1D convolution layers with 256 dimension. We set the time stride=2 on three different layers, resulted in $8 \times$ time-length reduction and followed by average pooling operation.

For the decoder, we use 10 1D convolution layers with 768 dimension and residual connection and we feed the style information by concatenating s in the channel axis on $\{1,3,5,7\}$ -th layers. Following [20], we set the $\gamma = 0.25$ in the final loss Eq. 5.

We evaluate variations of the model along two dimensions. First, we measure the model performance with different VQ-VAE codebook sizes with $K \in \{256, 512, 1024\}$. Second, we compare the disentanglement performance between models trained with vs. without the mutual information minimization loss. All models are optimized with Adam [25] optimizer up to 800,000 steps. In total, our model has ± 30 million parameters.

3.3. Experiments on Content Preservation

This set of experiments aim to measure how well the content encoder extracts and preserves the original content from the input speech. There are two different scenarios:

1. No shuffle: feeds speech X_i into both content and style encoder and predict the speech features \hat{X} with the decoder. Since each frame is encoded by 8 to 10 bits of VQVAE codes, this setup tests how well the model compresses speech.
2. Shuffle: feeds speech X_i into the content encoder and speech X_j ($i \neq j$) into the style encoder and predict the speech features \hat{X} with the decoder. This setup measures whether the speech contents are preserved by the VQVAE codes.

We use test-clean for both content and style encoder in order to test the performance of our system with unseen content and speaking style. To evaluate how similar the predicted speech content compared to the original content from speech X_i , we utilize a state-of-the-art pre-trained Conformer L [26] to transcribe the predicted features \hat{X} . After that, we calculate the word error rate for the transcription result with the ground-truth text from input speech X_i on the content encoder side.

Based on Table 1, we observed that increasing the codebook size allows the model to better preserve speech contents, as manifested by the decreasing WER for both shuffle and no shuffle scenarios. Adding the mutual information loss further improves content preservation with decreased WER across all code book sizes, for both shuffle and no-shuffle scenarios.

Table 1: Predicted features WER between different codebook sizes and with or without mutual information (MI) loss. The Conformer L model’s WER on the original test-clean set is 2.1%.

Codebook Size	MI Loss	WER (%)	
		No shuffle	Shuffle
256	No	4.6	7.7
	Yes	4.4	7.0
512	No	4.2	7.0
	Yes	3.8	6.8
1024	No	3.7	6.3
	Yes	3.6	6.1

Additionally, we trained a WaveRNN [27] model to reconstruct the new speech utterance from different contents and styles without any paired data. The WaveRNN is trained with input content C_i and style S_i , extracted from audio X_i with pre-trained content and style encoders. The objective is to minimize the reconstruction loss to the target waveform from audio X_i . To synthesize a new speech utterance from WaveRNN that has the same content from utterance X_i and speaking style from utterance X_j , we feed the content C_i and style S_j as the input for WaveRNN. We put the generated audio in the following URL: <https://unsup-content-style-demo-v1.netlify.app>.

3.4. Experiments on Style Disentanglement

This set of experiments measure how well the style encoder extracts and preserves the original speaking style from the input speech. We feed different input speech X_i into the content encoder and input speech X_j ($i \neq j$) into the style encoder and predict speech features \hat{X} based on the content and style latent variables.

Next, we evaluate the speaker similarity between the generated speech and X_i and X_j , the input speeches for the content and style encoders, respectively:

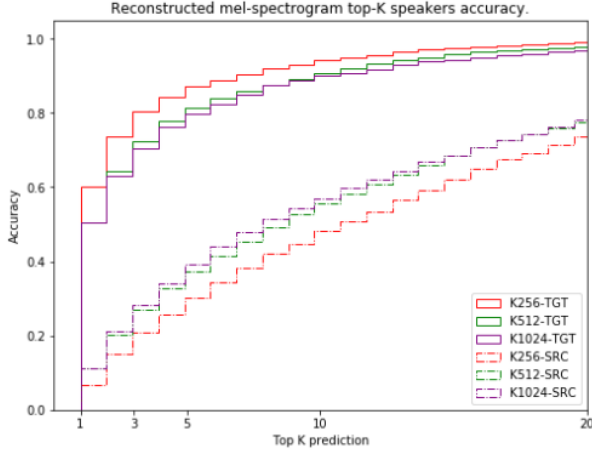


Figure 2: Top-K speaker accuracy compared to content (SRC) speaker ID and style (TGT) speaker ID. In this figure, we varied the codebook size and all models are trained with MI loss.

1. Train a speaker recognition model using ground-truth speaker labels;
2. Predict the softmax output on the predicted features \hat{X} .

Based on the softmax output predictions, we calculate several metrics such as: average ranking, top-1, top-3, and top-5 accuracy compared to the ground truth speaker on X_j . As shown in Figure 2, the generated speech has a speaker style much closer to X_j , the input to the style encoder, than X_i , the input to the content encoder.

Table 2 further shows how the speaker style accuracy for X_j changes as we vary codebook sizes and whether to enable mutual information loss. We can see that the style and content information becomes more entangled as we increase the codebook size. The mutual information loss helps improving the disentanglement, esp. for the smaller codebooks. Table 1 and Table 2 together highlight the tradeoffs between content reconstruction and style disentanglement.

Table 2: Predicted features top-K speaker ranks between different K codebook size and with or without mutual information loss.

Codebook	MI	Avg.	Top-1	Top-3	Top-5
256	No	3.9	0.5	0.72	0.81
	Yes	2.9	0.6	0.8	0.87
512	No	3.9	0.5	0.71	0.8
	Yes	3.7	0.5	0.73	0.81
1024	No	4.0	0.52	0.72	0.8
	Yes	4.0	0.51	0.71	0.8

3.5. Experiments on Few-Shot Speaker Recognition

Recently, there are many success stories for pre-training a model with a large amount of unlabeled data and fine-tune it with a small labeled data [1, 28]. In this experiment, we fine-tune a pre-trained style encoder for the few-shot speaker recognition task. For the dataset, we split the train-dev-test from train set 1000h (with a total unique speaker 2338) by sampling 1 or 3 utterance per speaker for the train, 5% for the dev and 5% for

the test without overlap. As shown in Figure 3, we add a linear projection layer on the top of the style encoder to compute the logits for speaker classification. During the fine-tuning, we freeze all parameters except the final softmax projection layer.

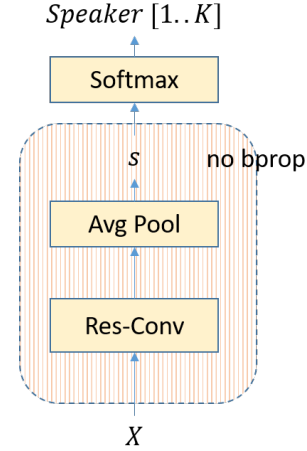


Figure 3: Using pre-trained style encoder as a speaker recognition by appending and fine-tuning a softmax layer.

We consider three scenarios for fine-tuning a speaker recognition model: 1) train from scratch without pre-trained weight (without freezing any weight), 2) initialize with a style encoder pre-trained without MI loss (freeze style encoder weight), 3) initialize with a style encoder pre-trained with MI loss (freeze style encoder weight). We also vary the amount of fine-tuning data with 1 or 3 examples per speaker, termed as “1-shot” and “3-shot”, respectively, in Table 3. The results show that we can achieve $> 90\%$ accuracy with 1-shot fine-tuning, compared with $< 11\%$ without a pre-trained encoder, and $> 95\%$ accuracy with 3 examples per speaker, compared with 63% without pre-training. Further, the use of mutual information loss brings small improvements to the results.

Table 3: 1-shot and 3-shot accuracy based on three difference scenarios. Model M1: model trained from scratch, M2: pre-trained style encoder without MI loss, M3: pre-trained style encoder with MI loss.

Model	1-shot acc (%)	3-shot acc (%)
M1	10.8	63.0
M2	90.5	95.9
M3	91.4	96.4

4. Conclusion

We present a framework to learn disentangled representations of speech with unlabeled data. The framework includes a local VQ encoder to extract a discrete sequence representation of the speech contents and a global VAE encoder to learn a continuous representation of speech styles. Our evaluation shows that the discrete sequence representation effectively captures the linguistic contents while the continuous global representation encapsulates the speaker style. Additionally, we also show the application from our pre-trained model, in which we successfully train a speaker recognition system with high accuracy ($> 90\%$) only with one sample per speaker.

5. References

- [1] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised pre-training for speech recognition,” *Proc. Interspeech 2019*, pp. 3465–3469, 2019.
- [2] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *CoRR*, vol. abs/1807.03748, 2018. [Online]. Available: <http://arxiv.org/abs/1807.03748>
- [3] Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, “Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis,” in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 5167–5176. [Online]. Available: <http://proceedings.mlr.press/v80/wang18h.html>
- [4] W. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen, P. Nguyen, and R. Pang, “Hierarchical generative modeling for controllable speech synthesis,” in *7th International Conference on Learning Representations, ICLR 2019*, 2019. [Online]. Available: <https://openreview.net/forum?id=rygk305YQ>
- [5] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
- [6] T.-Y. Hu, A. Shrivastava, O. Tuzel, and C. Dhir, “Unsupervised style and content separation by minimizing mutual information for speech synthesis,” in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3267–3271.
- [7] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised Pre-Training for Speech Recognition,” in *Proc. Interspeech 2019*, 2019, pp. 3465–3469. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2019-1873>
- [8] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [9] W.-N. Hsu, Y. Zhang, and J. Glass, “Unsupervised learning of disentangled and interpretable representations from sequential data,” in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 1878–1889.
- [10] Y. Yacoby, W. Pan, and F. Doshi-Velez, “Failure modes of variational autoencoders and their effects on downstream tasks,” *arXiv preprint arXiv:2007.07124*, 2020.
- [11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 2672–2680. [Online]. Available: <https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html>
- [12] T. Kaneko and H. Kameoka, “CycleGAN-vc: Non-parallel voice conversion using cycle-consistent adversarial networks,” in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 2100–2104.
- [13] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, “CycleGAN-vc2: Improved cyclegan-based non-parallel voice conversion,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6820–6824.
- [14] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.
- [15] L. Mescheder, A. Geiger, and S. Nowozin, “Which training methods for GANs do actually converge?” in *International conference on machine learning*. PMLR, 2018, pp. 3481–3490.
- [16] M. Lucic, K. Kurach, M. Michalski, O. Bousquet, and S. Gelly, “Are GANs created equal? a large-scale study,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 698–707.
- [17] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, “Unsupervised speech representation learning using wavenet autoencoders,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2041–2053, 2019.
- [18] A. Tjandra, B. Sisman, M. Zhang, S. Sakti, H. Li, and S. Nakamura, “VQVAE unsupervised unit discovery and multi-scale Code2Spec inverter for zerospeech challenge 2019,” in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association*. ISCA, 2019, pp. 1118–1122. [Online]. Available: <https://doi.org/10.21437/Interspeech.2019-3232>
- [19] A. Tjandra, S. Sakti, and S. Nakamura, “Transformer VQ-VAE for unsupervised unit discovery and speech synthesis: Zerospeech 2020 challenge,” *Proc. Interspeech 2020*, pp. 4851–4855, 2020.
- [20] A. Van Den Oord, O. Vinyals *et al.*, “Neural discrete representation learning,” in *Advances in Neural Information Processing Systems*, 2017, pp. 6306–6315.
- [21] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, “Mutual information neural estimation,” ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 531–540. [Online]. Available: <http://proceedings.mlr.press/v80/belghazi18a.html>
- [22] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Copyright Cambridge University Press, 2003.
- [23] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker, “On variational bounds of mutual information,” in *International Conference on Machine Learning*, 2019, pp. 5171–5180.
- [24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.
- [25] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [26] A. Gulati, C.-C. Chiu, J. Qin, J. Yu, N. Parmar, R. Pang, S. Wang, W. Han, Y. Wu, Y. Zhang, and Z. Zhang, “Conformer: Convolution-augmented transformer for speech recognition,” in *to appear Interspeech*, 2020.
- [27] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, “Efficient neural audio synthesis,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.
- [28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *ACL 2019*, Jun. 2019, pp. 4171–4186. [Online]. Available: <https://www.aclweb.org/anthology/N19-1423>