Vali, Mohammadhassan; Bäckström, Tom

Interpretable Latent Space Using Space-Filling Curves for Phonetic Analysis in Voice Conversion

Published in:
Proceedings of Interspeech Conference

DOI:
10.21437/Interspeech.2023-1549

Published: 01/01/2023

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Vali, M., & Bäckström, T. (2023). Interpretable Latent Space Using Space-Filling Curves for Phonetic Analysis in Voice Conversion. In *Proceedings of Interspeech Conference* (Vol. 2023-August, pp. 306-310). (Interspeech). International Speech Communication Association (ISCA). https://doi.org/10.21437/Interspeech.2023-1549

Interpretable Latent Space Using Space-Filling Curves for Phonetic Analysis in Voice Conversion

Mohammad Hassan Vali, Tom Bäckström

Department of Information and Communications Engineering, Aalto University, Finland

mohammad.vali@aalto.fi, tom.backstrom@aalto.fi

Abstract

Vector quantized variational autoencoders (VQ-VAE) are well-known deep generative models, which map input data to a latent space that is used for data generation. Such latent spaces are unstructured and can thus be difficult to interpret. Some earlier approaches have introduced a structure to the latent space through supervised learning by defining data labels as latent variables. In contrast, we propose an unsupervised technique incorporating space-filling curves into vector quantization (VQ), which yields an arranged form of latent vectors such that adjacent elements in the VQ codebook refer to similar content. We applied this technique to the latent codebook vectors of a VQ-VAE, which encode the phonetic information of a speech signal in a voice conversion task. Our experiments show there is a clear arrangement in latent vectors representing speech phones, which clarifies what phone each latent vector corresponds to and facilitates other detailed interpretations of latent vectors.

Index Terms: interpretable latent space, phonetic analysis, space-filling curves, vector quantization, voice conversion

1. Introduction

As a well-known type of deep generative models, vector quantized variational autoencoders (VQ-VAE) [1] have recently gained success in a wide range of applications such as voice conversion [2, 3], image generation [4], speech and audio coding [5], music generation [6], and text-to-speech synthesis [7, 8]. They can model complex high-dimensional data to an abstract latent space that captures the statistical properties of the data. Hence, they are useful for data generation, feature extraction, and data compression purposes while generating high-quality reconstructions from the latent space. However, this latent space has neither interpretable nor explorable structure. For example, it is not clear what information each latent codebook vector captures, or what property each latent space dimension expresses in a form understandable by human users.

In generative models, the term *interpretability* for the latent space is considered from two different perspectives as follows:
Introducing structure into the latent space: In this line of work, there are two different types of approaches; supervised and unsupervised. By using labeled data, supervised methods [9, 10, 11] employ data labels to define an isolated subgroup in the latent space for each specific label. After training, they have a learned subgroup of the latent representation for each class of data, which is easy to interpret and use for data manipulation and generation. However, these methods require human labeling, and they might critically prevent the learned representations to capture some inherent structures existing in the data, which the human labeler is unaware of [12]. On the other hand, unsupervised methods [13, 14, 15] aim to learn a disentangled latent space. These methods try to model the generative factors existing in the data in a way that changes in each latent dimension only modify a single property of the generated data while keeping other generative factors invariant to these changes. In other words, they force different dimensions of the latent space to be independent of each other. Hence, they render an interpretable latent space such that each specific dimension corresponds to a unique generative factor. While learning a disentangled representation, these models are however less efficient in achieving high quality and diverse generations [12].

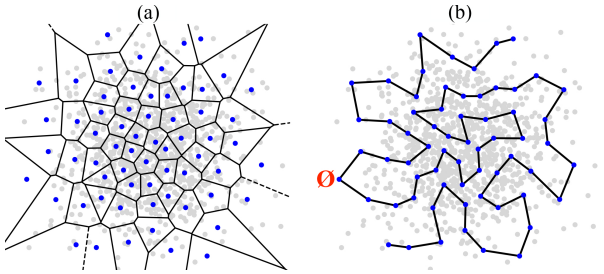

Figure 1: *(a) A vector quantizer for a Gaussian distribution (gray points) with $K = 64$ codebook vectors (blue points) and the corresponding Voronoi regions (black lines). (b) A space-filling vector quantizer (thick black line) for the same data (the point marked by Ø in red shows an outlier corner point).*

Exploring meaningful directions in the latent space: The methods proposed in [12, 16, 17] aim to discover directions in the latent representations such that moving on these directions yields humanly interpretable image transformations such as a change in aging, gender, hairstyle, lighting and etc. They usually detect these directions by analyzing the latent space of pretrained generative models and their interpretability implies achieving a controllable generation process by manipulation of latent vectors along the detected directions.

In this paper, we propose a novel tool to explore the underlying structure in the latent space of a vector quantized variational autoencoder (VQ-VAE) used in the voice conversion task presented in [2]. To this end, we incorporate space-filling curves into the vector quantization module such that codebook vectors of the latent space are the actual corner points of a space-filling curve. Since space-filling curves have inherent structure, they automatically create a shapely arrangement for the latent codebook vectors in an unsupervised learning manner. Hence, contrary to the supervised approaches [9, 10, 11], our method does not incur any human labeling and manual restrictions on the learned latent space. To make our method interpretable, it requires the user to study the learned latent space entirely only once by observation. Experiments demonstrate

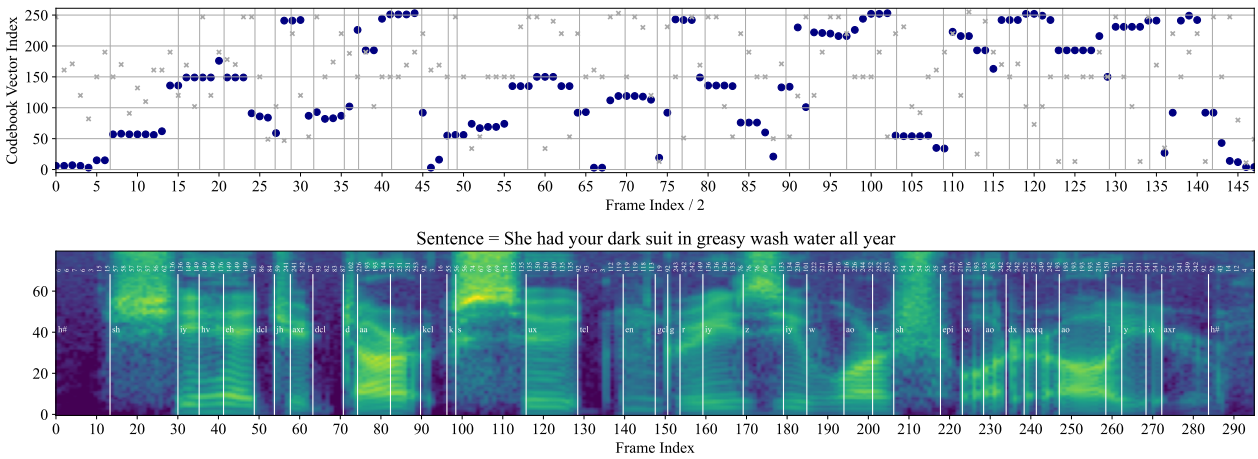

Figure 2: *(top) Codebook vector indices for the speech signal using our proposed space-filling vector quantizer (in dark blue circles) and the ordinary vector quantizer in [2] (in gray crosses). (bottom) Mel-spectrogram of the speech signal including codebook vector indices corresponding to speech frames.*

that our proposed space-filling vector quantizer achieves a coherently structured and easily interpretable representation for latent codebook vectors, which represent phonetic information of the input speech. Accordingly, there is an obvious distinction of various phonetic groupings such as {consonants vs. vowels}, {fricatives vs. nasals vs. approximants vs. ...}, and we can simply tell apart which phone each codebook vector represents. In addition, similar phones within a specific phonetic group are encoded next to each other in the latent codebook space. Specifically, the novelties of this work are twofold:

1. Introducing space-filling vector quantizer; a new tool for unsupervised modeling of a data distribution in a structured way.
2. Applying the proposed tool for interpretation and exploration of latent space (specifically phonetic analysis in this study).

2. Methods

Vector quantization [19] is a data compression technique (similar to k-means) that is used in a wide range of applications such as speech coding [20], voice conversion [2, 3], and image compression [21]. For an input vector $\mathbf{x} \in \mathbb{R}^{1\times D}$ and codebook matrix $\mathbf{C} \in \mathbb{R}^{K\times D}$, vector quantization is defined as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{c}_k} \|\mathbf{x} - \mathbf{c}_k\|^2, \quad 0 \leq k < K, \tag{1}$$

where $\|\cdot\|^2$ is the Euclidean distance and $\mathbf{c}_k \in \mathbb{R}^{1\times D}$ is the closest codebook vector from $\mathbf{C}$ to the input vector $\mathbf{x}$ in terms of Euclidean distance. In other words, vector quantization maps the input vector to the closest codebook vector.

Space-filling curve is a piece-wise continuous line generated with a recursive rule and gets bent until it completely fills a multi-dimensional space when repeated infinitely. A common type of space-filling curve is known as the Hilbert curve [22], in which the corner points are defined using a specific formulation at each iteration. Getting intuition from it, we can thus think of vector quantization as mapping input data points on a space-filling curve (rather than only mapping the input exclusively on codebook vectors). Therefore, we incorporate vector quantization into space-filling curves, such that our proposed space-filling vector quantizer models a D-dimensional data distribution by continuous piece-wise linear curves whose corner points are vector quantization codebook entries. Fig. 1 illustrates a vector quantizer and its space-filling variant applied to a Gaussian distribution.

Proposed space-filling vector quantizer (training): To map the input vector $\mathbf{x}$ to a space-filling line during training, we use a dithering technique that prevents codebook vectors from diverging. To this end, we define a dithered codebook by interpolating at random points between subsequent vectors of the current base codebook. This dithered codebook is then used as in conventional vector quantization. It means, we map the input vector to the closest codebook vector from the dithered codebook as

$$\hat{\mathbf{x}} = (1-\lambda)\mathbf{c}_j + \lambda\mathbf{c}_{j+1}, \tag{2}$$

where $\mathbf{c}_j$ and $\mathbf{c}_{j+1}$ are the vectors from the base codebook which their interpolation is the closest dithered codebook vector to the input vector, and λ is the interpolation (dithering) factor. To apply dithering, we sample λs from a uniform distribution of $U[0, 1)$ during training that guarantees interpolating at random points between subsequent base codebook vectors. Such interpolation enforces an assumption of continuity between subsequent codebook elements and randomizing the interpolation implements a dithering or regularization effect. If we intend to train the space-filling vector quantizer module exclusively (independent from any other module that requires training) on a distribution, we can use mean squared error (MSE) between the input vector and its quantized version as the loss function

$$\text{MSE}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - (1-\lambda)\mathbf{c}_j - \lambda\mathbf{c}_{j+1}\|^2. \tag{3}$$

In this case, we do not need to pass gradients through the non-differentiable *argmin* function in Eq. (1) (gradient collapse problem [23]), since $\hat{\mathbf{x}}$, and as a result, the MSE loss are a function of leaf variables $(\mathbf{c}_j, \mathbf{c}_{j+1})$ which need gradients. However, if we intend to train the space-filling vector quantizer jointly with other modules, we need to pass the gradients through *argmin* function using our recently proposed noise substitution in vector quantization (NSVQ) technique [23] through which the final quantized input vector is defined as

$$\tilde{\mathbf{x}} = \mathbf{x} + \|\mathbf{x} - \hat{\mathbf{x}}\| \cdot \frac{v}{\|v\|}, \tag{4}$$

where v is a vector sampled from a zero-mean, unit-variance normal distribution ($\mathcal{N}(0, 1)$).

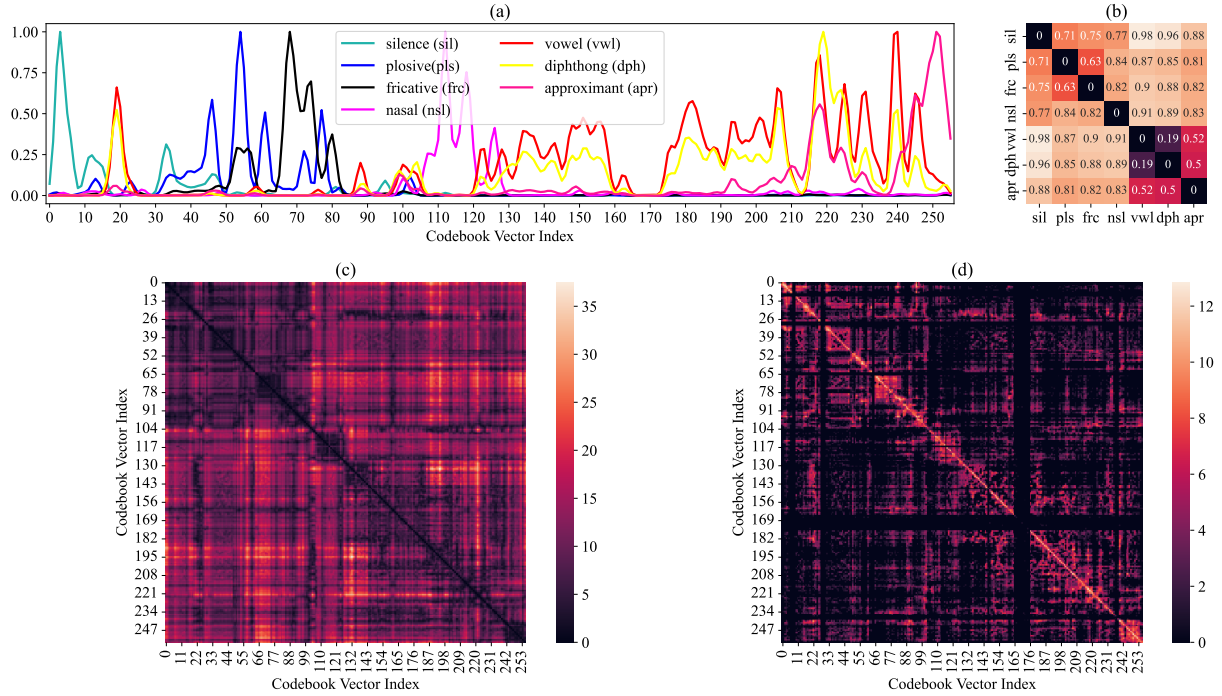

Figure 3: *(a) Histogram of codebook vector indices for phonetic groups. (b) Confusion matrix representing the Jensen-Shannon distance [18] between phonetic groups. (c) Heatmap of Euclidean distance between all codebook vectors. (d) Heatmap representing the number of times each codebook vector (each column) occurs immediately after a specific codebook vector (each row).*

Proposed space-filling vector quantizer (inference): Through analytical minimization of the mean squared error (MSE) in Eq. (3) between $\mathbf{x}$ and $\hat{\mathbf{x}}$ in the interval between i and $i+1$, we find the optimal λ as

$$\lambda_{optimal} = \frac{(\mathbf{c}_{i+1} - \mathbf{c}_i)^T(\mathbf{x} - \mathbf{c}_i)}{\|\mathbf{c}_{i+1} - \mathbf{c}_i\|^2}. \quad (5)$$

At inference, to find where an input vector $\mathbf{x}$ is mapped on the space-filling curve, we first find the index k^* of the closest codebook vector to $\mathbf{x}$ using Eq. (1). Then, we consider two possible intervals of $\{\mathbf{c}_{k^*-1}, \mathbf{c}_{k^*}\}$ and $\{\mathbf{c}_{k^*}, \mathbf{c}_{k^*+1}\}$ for mapping. By using Eq. (5) and Eq. (2), we calculate $\lambda_{optimal}$ and $\hat{\mathbf{x}}$ for each interval, respectively. Then, we calculate the MSE between input vector $\mathbf{x}$ and two calculated mappings $\hat{\mathbf{x}}$ for both intervals and map $\mathbf{x}$ to the interval which gives a smaller MSE value.

3. Experiment

To evaluate how our proposed space-filling vector quantizer (SFVQ) finds the structure in the latent space, we applied it to a voice conversion task based on a vector quantized variational autoencoder (VQ-VAE) network [2]. The vector quantization (VQ) module in VQ-VAE is an information bottleneck that learns a discrete representation of speech that captures phonetic content and discards the speaker-related information. In other words, VQ codebook vectors are expected to collect only the phone-related content of the speech. We trained the VQ-VAE network from scratch using the same settings as in the original paper [2]. The only difference is that we reduced the number of codebook vectors from 512 to 256 for the VQ module, since we wanted to impose a more strict constraint to the bottleneck, to make sure that there is no speaker-related information leaked to the codebook vectors [24]. We trained the model on the *ZeroSpeech 2019 Challenge* English dataset [25] (containing about 15 h of speech from 102 speakers) for 500 k training batches (batch size = 52) using Adam optimizer with the initial learning rate of $4 \cdot 10^{-4}$. We halved the learning rate after 300 k and 400 k training batches. After training, we discarded the trained VQ and decoder modules and kept only the trained encoder which learned to extract only phone-level information of input speech signal. Then, we froze the encoder parameters and trained our proposed SFVQ (with 256 codebook vectors or corner points) on what the encoder generates from the entire *ZeroSpeech* dataset. The PyTorch implementation of our proposed SFVQ is publicly available[1].

4. Results and discussion

We evaluate the performance of our space-filling vector quantizer (SFVQ) on its ability to find the structure in the latent codebook vectors representing phonetic information in a voice conversion task [2]. For our evaluations, we used the TIMIT dataset [26] (both train and test sets), since it contains phone-wise labeled data using the phone set from [27]. It also indicates the consistency of our model by employing different datasets for training and evaluation. For our experiments, we use the phonetic grouping based on [28] as

- Plosives (stops): {p, b, t, d, k, g, jh, ch}
- Fricatives: {f, v, th, dh, s, z, sh, zh, hh, hv}
- Nasals: {m, em, n, nx, ng, eng, en}
- Vowels: {iy, ih, ix, eh, ae, aa, ao, ah, ax, ax-h, uh, uw, ux}
- Diphthongs: {ey, aw, ay, oy, ow}
- Approximants (semi-vowels): {l, el, r, er, axr, w, y}
- Silence: {h#}.

[1] https://gitlab.com/speech-interaction-technology-aalto-university/sfvq

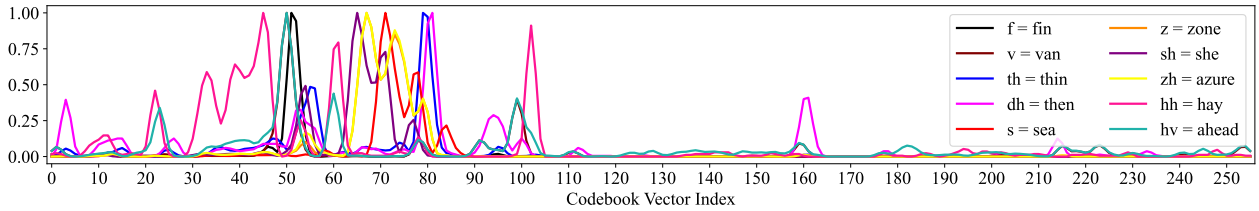

Figure 4: *Histogram of codebook vector indices for fricative phones.*

To analyze the performance of our proposed SFVQ, we pass the labeled TIMIT speech files through the trained encoder and SFVQ modules, respectively, and extract the codebook vector indices corresponding to all existing phones in the speech. As explained in section 2, we expect our SFVQ to map similar phonetic contents next to each other. To examine this expectation, in Fig. 2 we visualize the Mel-spectrogram of the sentence "*She had your dark suit in greasy wash water all year*", and its corresponding codebook vector indices for the ordinary vector quantizer (VQ) in [2] and our proposed SFVQ. We observe that the indices of the ordinary VQ [2] does not have any particular structure. However, when using our proposed SFVQ, there is a clear structure for the codebook vector indices. The indices for the frames containing phones {sh, s, z} within the words {she, wash, suit, greasy} are uniformly distributed next to each other throughout the frames. In addition, silence frames containing phone {h#} and some other low energy frames containing {kcl, tcl, gcl: k, t, g closures} within the words {dark, suit, greasy} are uniformly located next to each other in the range [0-20]. Notice that we have informally found that Fig. 2 remains consistent for sentences with the same phonetic content, even across speakers with different genders, speech rhythms, and dialects.

Fig. 3.a demonstrates the histogram of codebook indices for each phonetic group. Histograms are first smoothed by a Blackman windowing with window size 7 and then normalized by their maximum values. At first glance, we observe that consonants:{silence, plosives, fricatives, nasals} and vowels:{vowels, diphthongs, approximants} can be separated around index 125 (apart from the peak near index 20). We also observe that the most prominent peaks of different groups are separated in different parts of the histogram. There are two exceptions; First, the silence part in the range [30-50], refers to frames labeled as silence, but still contain a little energy in their Mel-spectrogram, making them sound similar to fricatives of {hh, hv} and some plosives. Second, the peak near index 20 (for vowel, diphthong, and approximant groups) refers to phones containing sounds similar to the approximant {y} ({iy, ih, ix} from vowels, and {ey, ay, oy} from diphthongs).

To investigate how well our proposed SFVQ preserves similarity for adjacent codebook vectors, Fig. 3.c presents the heatmap of Euclidean distance between codebook vectors. It is expected that the black squares (neighboring codebook vectors) along the diagonal axis represent the phonetic elements that share similar properties. Based on subjective observations, we could identify such black squares at: [0-20]=[silence:{h#}], [30-50]=[fricatives:{hh,hv}, plosives], [64-81]=[fricatives:{s, z, sh, zh}], [105-125]=[nasals], [126-143]=[approximant:{y}, vowels:{iy, ih, ix}, diphthongs:{ey, ay, oy}], [210-230]=[vowels:{ao, aa, ah, ax, ax-h, uw, ux, uh}, diphthongs:{aw, ow}, approximants:{l, el, w}], [244-255]=[approximants:{r ,er, axr}]. In addition, the big black squares at [0-104]=[consonants (except nasals):{silence, fricatives, plosives}] and [144-255]=[vowels:{vowels, diphthongs, approximants}] show a clear separation between consonants and vowels.

To quantify the distance between different phonetic groups, we calculated the Jensen-Shannon distance (JSD) [18] between each pair of histograms (see Fig. 3.b). Given two histograms, the larger the JSD value, the more dissimilar the corresponding phonetic groups are. Based on the JSD values, we again observe a clear distinction between consonants and vowels. Moreover, the most disparate histograms are silence vs. vowels, and the most similar ones are vowels vs. diphthongs.

In Fig. 2 we can observe that subsequent frames often feature similar phonemes. To characterize such similarity over time, Fig. 3.d shows the number of times (in $\log_2$-magnitude) each codebook vector (each column) occurs immediately after a specific codebook vector (each row). A few important observations; 1) Vowels:{vowels, diphthongs, approximants} are less probable to happen immediately after silence (specifically silence indices [0-10]). 2) Nasals are less probable to happen immediately after plosives and fricatives. 3) Plosives and fricatives are less probable to happen immediately after nasals, except fricatives of {s, th} like in the words {hamster, fancy, warmth, length} and plosives of {k, p} like in the words {bank, link, input, jump}. 4) According to histograms in Fig. 3.a and due to the blank black area in the range [165-175], it is inferred that the codebook vectors in this range do not model any phonetic content (they are outliers which do not pass through the data distribution similar to the corner point marked by Ø in Fig. 1.b).

As an example for distribution of phones within a phonetic group, Fig. 4 illustrates the histograms of all fricatives. Histograms are first smoothed by a Blackman windowing with window size 7 and then normalized by their maximum values. By observing the most prominent peak for each phone, we find out the peaks of similar phones are located next to each other. To elaborate, we listed similar phones and their corresponding peak index here as {f:51, v:50}, {th:78, dh:80}, {s:71, z:67, sh:65, zh:67}, {hh:46, hv:50}. Except {hh, hv} phones, fricatives are mainly located in the range [50-85]. Further structures can be readily identified from all provided figures by visual inspection.

5. Conclusions

Vector quantized variational autoencoders (VQ-VAE) are well-known deep generative models that have an unstructured latent space that is difficult to interpret. In this paper, we proposed the space-filling vector quantizer (SFVQ) which reveals an unsupervised, coherently structured latent space in a VQ-VAE-based voice conversion task, enabling detailed phonetic interpretations of the latent space. The proposed SFVQ can also assist interpretation of the latent space for other types of data (e.g. images) in other applications. By using the proposed SFVQ, we can thus interpret and explain how unsupervised deep neural networks internally represent information which in turn can help us better understand and trust machine learning models.

6. References

[1] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proceedings of NeurIPS*, 2017, pp. 6309–6318.

[2] B. van Niekerk, L. Nortje, and H. Kamper, "Vector-quantized neural networks for acoustic unit discovery in the zerospeech 2020 challenge," in *Proceedings of Interspeech*, 2020, pp. 4836–4840.

[3] S. Ding and R. Gutierrez-Osuna, "Group latent embedding for vector quantized variational autoencoder in non-parallel voice conversion," in *Proceedings of Interspeech*, 2019, pp. 724–728.

[4] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," in *Proceedings of NeurIPS*, 2019, pp. 14 866–14 876.

[5] C. Gârbacea, A. v. den Oord, Y. Li, F. S. C. Lim, A. Luebs, O. Vinyals, and T. C. Walters, "Low bit-rate speech coding with VQ-VAE and a Wavenet decoder," in *Proceedings of ICASSP*, 2019, pp. 735–739.

[6] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: a generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[7] 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 *Proceedings of Interspeech*, 2019, pp. 1118–1122.

[8] X. Wang, S. Takaki, J. Yamagishi, S. King, and K. Tokuda, "A vector quantized variational autoencoder (VQ-VAE) autoregressive neural F_0 model for statistical parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 157–170, 2020.

[9] J. Klys, J. Snell, and R. Zemel, "Learning latent subspaces in variational autoencoders," in *Proceedings of NeurIPS*, 2018, pp. 6445–6455.

[10] Y. Xue, M. Ding, and X. Lu, "Supervised vector quantized variational autoencoder for learning interpretable global representations," *arXiv preprint arXiv:1909.11124*, 2019.

[11] S. An, H. Choi, and J.-J. Jeon, "EXoN: Explainable encoder network," *arXiv preprint arXiv:2105.10867*, 2021.

[12] A. Voynov and A. Babenko, "Unsupervised discovery of interpretable directions in the gan latent space," in *Proceedings of International Conference on Machine Learning*, 2020, pp. 9786–9796.

[13] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Proceedings of NeurIPS*, 2016.

[14] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *Proceedings of International Conference on Learning Representations*, 2017.

[15] W. Lee, D. Kim, S. Hong, and H. Lee, "High-fidelity synthesis with disentangled representation," in *Proceedings of European Conference on Computer Vision*, 2020, pp. 157–174.

[16] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "Ganspace: Discovering interpretable gan controls," in *Proceedings of NeurIPS*, 2020, pp. 9841–9850.

[17] H. Yang, L. Chai, Q. Wen, S. Zhao, Z. Sun, and S. He, "Discovering interpretable latent space directions of gans beyond binary attributes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 177–12 185.

[18] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.

[19] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Springer Science & Business Media, 2012, vol. 159.

[20] J. Casebeer, V. Vale, U. Isik, J.-M. Valin, R. Giri, and A. Krishnaswamy, "Enhancing into the codec: Noise robust speech coding with vector-quantized autoencoders," in *Proceedings of ICASSP*, 2021, pp. 711–715.

[21] D. Báscones, C. González, and D. Mozos, "Hyperspectral image compression using vector quantization, pca and jpeg2000," *Remote Sensing*, vol. 10, 2018.

[22] H. Sagan, *Space-filling curves*. Springer Science & Business Media, 2012. [Online]. Available: https://doi.org/10.1007/978-1-4612-0871-6

[23] M. H. Vali and T. Bäckström, "NSVQ: Noise substitution in vector quantization for machine learning," *IEEE Access*, vol. 10, pp. 13 598–13 610, 2022.

[24] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *Proceedings of International Conference on Machine Learning*, 2019, pp. 5210–5219.

[25] "Zerospeech 2019 challenge dataset," [Online; accessed 06.03.2023]. [Online]. Available: https://download.zerospeech.com

[26] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, *The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CDROM*. Linguistic Data Consortium, 1993.

[27] C. Lopes and F. Perdigao, "Phoneme recognition on the TIMIT database," in *Speech Technologies*. IntechOpen, 2011, ch. 14. [Online]. Available: https://doi.org/10.5772/17600

[28] P. Scanlon, D. P. Ellis, and R. B. Reilly, "Using broad phonetic group experts for improved speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 803–812, 2007.