Inspecting and Interacting with Meaningful Music Representations using VAE

Ruihan Yang
Music X Lab
NYU Shanghai
ry649@nyu.edu

Tianyao Chen
Music X Lab
NYU Shanghai
tc2709@nyu.edu

Yiyi Zhang
Center for Data Science
New York University
yz2092@nyu.edu

Gus Xia
Music X lab
NYU Shanghai
gxia@nyu.edu

ABSTRACT

Variational Autoencoders [9] (VAEs) have already achieved great results on image generation and recently made promising progress on music generation. However, the generation process is still quite difficult to control in the sense that the learned latent representations lack meaningful music semantics. It would be much more useful if people can modify certain music features, such as rhythm and pitch contour, via latent representations to test different composition ideas. In this paper, we propose a new method to inspect the pitch and rhythm interpretations of the latent representations and we name it *disentanglement by augmentation*. Based on the interpretable representations, an intuitive graphical user interface is designed for users to better direct the music creation process by manipulating the pitch contours and rhythmic complexity.

Author Keywords

Representation learning, Disentanglement, Music generation, Controlled generation

1. INTRODUCTION

Representation learning has become an essential tool to gain an in-depth view of the data. Bengio [1] pointed out that good data representations "make it easier to extract useful information when building classifiers or other predictor". We have also seen that representation learning dramatically boosted the effectiveness of generative models for visual arts and music style transfer [4, 2]. In particular, the general encoder-decoder architecture of Variational Autoencoders (VAEs) [9] (and in the same sense, the generator-discriminator architecture of Generative Adversial Network [5]) provides a way to generate data by sampling from the distribution of the latent representation, rather than directly sampling from the data distribution or generating one token or pixel at one time. More recently, the pioneer work of MusicVAE [10] incorporated sequence modeling with VAEs by building both the encoder and decoder with Long Short-Term Memory networks (LSTMs) [8]. However, since the learned latent representations lack semantic interpretations, the generation process is still quite difficult to control. From a practical standpoint, it would be helpful if users could manipulate meaningful music features via latent representations during the music creation process in a similarly way of tuning the radio with turning knobs, in order to test different composition ideas efficiently.

We aim to gain a better semantic interpretation of the learned latent representations by disentanglement, i.e., to inspect which part (dimensions) of the representations connects with which features of music composition, while providing an intuitive interface to control the disentangled features. In this paper, we adopt the MusicVAE model [10] and focus on disentangling and interacting with *pitch contour* and *rhythm* representations of symbolic melodies. We propose a new method named *disentanglement by augmentation* and conduct the experiment in two ways: one way only transpose the pitch contour, and the other way only split the note duration, both gradually. We discover that encoded latent representations change approximately linearly along with the changing of pitch contour and rhythm. Moreover, only a small portion of dimensions changes significantly comparing to the entire large space. Among them, almost no dimension contributes to both pitch contour changes and rhythm changes. Therefore, pitch contour and rhythm are considered to be disentangled. By fixing one of these sets of dimensions and releasing the other one, we can make the original music transformed to new music in a desired way. Meanwhile, we provide an intuitive interface for users to create music via directly interacting with or interpolating the significant dimensions of pitch contour and rhythm representations. Available at: `https://github.com/cdyrhjohn/representation_demo`

In the rest of the paper, Section 2 gives a background knowledge about the VAEs and MusicVAE models, Section 3 discusses how to inspect the latent space, Section 4 introduces the interface we design for music composition, and the last section summaries our study and discusses possible future work.

2. BACKGROUND

Our study is built on VAEs [9]. In this section, we review how VAEs work and how to adjust the model architecture to deal with the representation learning of symbolic music sequences.

2.1 Variational Autoencoder

The VAE [9] is a classical generative model that can learn a low-dimensional latent vector $\boldsymbol{z}$ from a high-dimensional data $\boldsymbol{x}$. The latent vector $\boldsymbol{z}$ can be used to generate new data or to reconstruct the original data. The prior distribution of $\boldsymbol{z}$ is $p(\boldsymbol{z})$. Therefore, the generated data $\boldsymbol{x}$ need to satisfy the distribution $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{z})$. The encoder in the VAE uses a distribution $q(\boldsymbol{z}|\boldsymbol{x})$ to approximate $p(\boldsymbol{z}|\boldsymbol{x})$ and the decoder parameterizes the distribution $p(\boldsymbol{x}|\boldsymbol{z})$. The objective is to minimize the KL divergence between $q(\boldsymbol{z}|\boldsymbol{x})$

NIME'19, June 3-6, 2019, Federal University of Rio Grande do Sul, Porto Alegre, Brazil.

and $p(\boldsymbol{z})$ by maximizing the evidence lower bound:

$$\mathbb{E}[\log p(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}(q(\boldsymbol{z}|\boldsymbol{x}) \mid\mid p(\boldsymbol{z})) \leq \log p(\boldsymbol{x}) \qquad (1)$$

Generally speaking, both encoder and decoder are neural networks, and $p(\boldsymbol{z})$ is parameterized as a diagonal covariance Gaussian, i.e., $\boldsymbol{z} \sim \mathcal{N}(\mu, \sigma)$.

In our model, after the training period, we take the mean of distribution $p(\boldsymbol{z})$, making $\boldsymbol{z} = \mu$, as the latent representation of the input data.

2.2 Music Variational Autoencoder

MusicVAE [10] is an application of variational autoencoder on music. The format of the input music required by this model is in the form of a 2-bar MIDI sequence with a 4/4 meter. Each 2-bar music clip is represented by a 32 × 130 matrix, with 32 time steps in the unit of sixteenth note, and 130 states on each time step, including 128 onset states for each MIDI pitch, a holding state, and a rest state. To deal with this time-series data, MusicVAE uses bidirectional Long Short-Term Memory networks (LSTMs) [8] recurrent neural networks for both encoder and decoder. In this paper, we use Gated Recurrent Units (GRUs) [3] as a replacement of LSTMs for more efficient training processes. The model architecture is shown in Figure 1. The learned low-dimensional latent vector $\boldsymbol{z}$ can be seen as a compact and continuous latent representation, based on which we can smoothly transform from one music clip to another one through interpolation or adding certain features via attribute vector arithmetic [7].

Note that the original MusicVAE model has a hierarchical structure to handle longer sequences, but results on music with two bars are not yet convincing. Therefore, we only adopt the model to learn short sequences. When dealing with longer sequences, we cut them into 2-bar clips, process them one by one, then concatenate them together.

3. METHODS

Our goal is to allow music composers to manipulate latent representations of a music melody easily with an effective and intuitive interface. Two improvements are needed based on MusicVAE [10]. First, we need a more compact latent representation. To this end, we shrink the dimensionality of the latent space from 512 to 128. Second, we need to disentangle the latent representation in a meaningful way. Each disentangled part of the representation should coincide with some explicit music concept, such as pitch or rhythm, thus human composers can manipulate these music features by changing the values on the latent representations while knowing the consequences. As an analogy, the sound produced by a radio can be controlled by tuning labelled knobs on a radio, while each label, such as volume or frequency, assigned to the knobs tells users the consequences of tuning them. The meaningful disentangled representation makes the design of the interface more effective and efficient for human interaction and testing creative composition ideas.

To help the model extract representations more precisely, the KL divergence is re-weighted by a value β, which is set as 0.1 in our case to help the model focus more on reconstruction rather than innovation. We refer readers to [6] for more information on β-VAE.

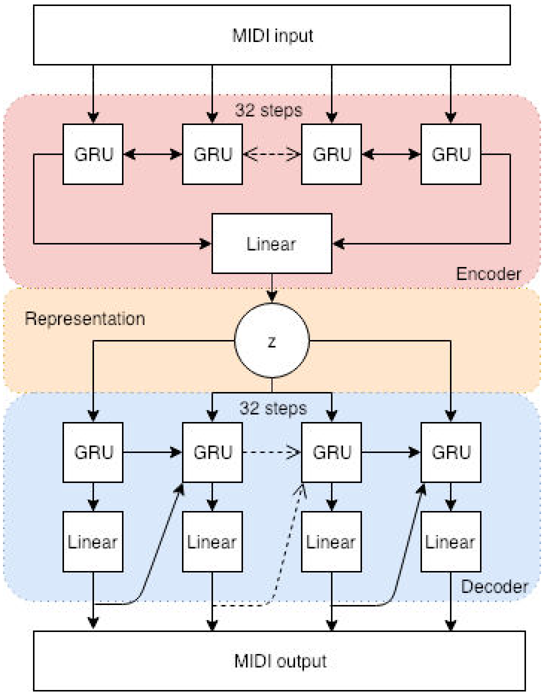


Figure 1: An illustration of the VAE model to learn music representation

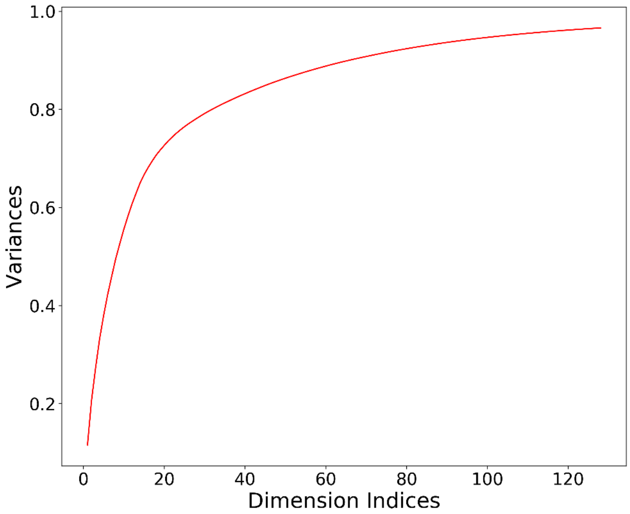


Figure 2: Sorted variance in the 128-dimension vector after PCA

3.1 A More Compact Representation

We first examined the size of the latent space. The dimensionality of the latent space designed by the original MusicVAE [10] is too large for an intuitive and controllable interface. MusicVAE uses a Euclidean space with 512 dimensions. We train a MusicVAE model on our dataset, of which the encoder maps each music clips to a 512-dimension vector in the latent space.

We performed *principal components analysis (PCA)* [12] on latent vectors of music clips across the dataset. As shown in figure 2, we observe that more than 99% of the aggregated eigenvalues are covered by the first 128 dimensions. Meanwhile, the sorted variance in figure 2 indicates that even less than 128 dimensions have major variances. To be cautious, we select 128 as our latent space size to ensure that the model could learn a more complete representation. In the following sections, we use a MusicVAE with only 128 latent space dimensions pretrained on the dataset.

3.2 Disentanglement by Augmentation

We propose a new method named *disentanglement by augmentation*, which can be considered a special case of *analysis by synthesis* [13]. Start from a well-trained MusicVAE, of which the training process follows the original MusicVAE paper [10], which consists of an ENCODER function and a DECODER function. The ENCODER maps a certain observed music clip $\mathbf{M}_i$ to a latent vector $\boldsymbol{z}_i$, while the DECODER maps the latent vector $\boldsymbol{z}_i$ back into the original music clip $\mathbf{M}_i$.

Now, consider a data augmentation function F, which can directly transform the input music clips. Theoretically, for any F, there would be a corresponding function f in the latent space, which moves the latent vector of the original music clip to the latent vector of the transformed music clip. Formally,

$$f(\text{ENCODER}(\mathbf{M}_i)) = \text{ENCODER}(F(\mathbf{M}_i)) \tag{2}$$

For a latent vector $\boldsymbol{z}_i$ of music clip $\mathbf{M}_i$, transformation f can be in the form of

$$f(\boldsymbol{z}_i) = \boldsymbol{z}_i + \Delta_f \boldsymbol{z}_i. \tag{3}$$

Since $p(\boldsymbol{z})$ satisfies a multivariate independent normal distribution, an ideal assumption about f is that, given a fixed transformation F and its corresponding latent space transformation f, for all music clips $\mathbf{M}_i$, the differences between latent vectors before and after the transformation, $\Delta_f \boldsymbol{z}_i$, are non-zero in certain components, while the rest are kept zero.

The assumption is obviously too strong. Instead, we have another practical hypothesis that, latent vectors have significant changes only on a few dimensions. Given an orthonormal basis of the latent space, $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n\}$, we assume that

$$\left|\Delta_f \boldsymbol{z}_i \cdot \boldsymbol{e}_{d_j}\right| \geq t \tag{4}$$

on dimensions $\{d_1, d_2, \ldots, d_k\}$, where t is a threshold determined manually.

We further assume that on those dimensions, latent vectors have significant changes on the same direction, either positive correlated or negative correlated. Therefore, we can use the average of the differences to select the dimensions of significant change,

$$\left|\overline{\Delta_f \boldsymbol{z}} \cdot \boldsymbol{e}_{d_j}\right| \geq t \quad \overline{\Delta_f \boldsymbol{z}} = \sum_{i=1}^{N} \Delta_f \boldsymbol{z}_i \tag{5}$$

For simplicity, we choose the standard basis as our orthonormal basis of the latent space.

3.3 Inspecting Pitch Representations

To inspect which dimensions of the latent space contribute the most to pitch information, we defined a set of pitch augmentation function F_p^{pitch}, which transposes the pitches of all notes of a music clip up by p semitones . (For instance, F_3^{pitch} means to transpose the pitches up by a minor third.) For each p, we calculated a set of $\Delta \boldsymbol{z}_i^p$ and its average $\overline{\Delta_f \boldsymbol{z}^p}$. In practice, a batch size of 10k is used, and p ranges from 1 (a half step) to 12 (an octave).

Figure 3 shows the top five latent dimensions that change most for different p, where the twelve lines corresponds to the twelve values of p. We see that, for each p, only 2 dimensions of the latent representations change significantly, while other dimensions almost keep the same. In addition, the sets of significant dimensions for different p overlap a lot, and the top 2 dimensions always remain the same for different p. This discovery supports our assumption in section 3.2. We conclude that only a small portion (only two dimensions) of the latent representations contributes to pitch variation, and we refer to these two dimensions as *pitch representations*.

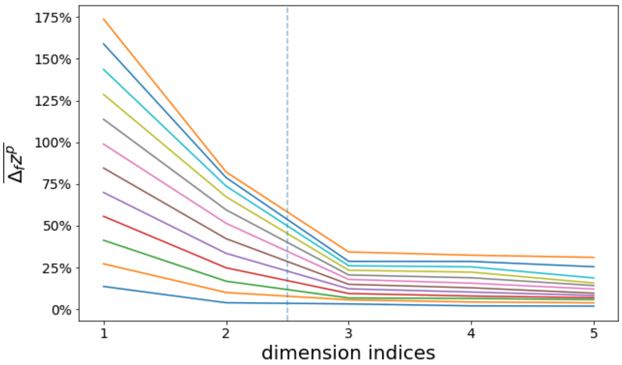


Figure 3: An illustration of only two latent dimensions contribute the most to pitch variation.

3.4 Inspecting Rhythm Representations

To inspect which dimensions of the latent space contribute the most to rhythm information, we define a set of rhythm augmentation function F_n^{rhythm}, which splits each note into n equal-length ones with the same pitch. For instance, F_2^{rhythm} splits a quarter note into two consecutive eighth notes. This augmentation cuts the melody but the rough pitch trend still remains. For each n, we get a set of Δz_i^n and its average $\overline{\Delta_f z^n}$. In practice, we build a dataset which has 16483 2-bar music clips, each is composed of two whole notes. For n, we take $n = 2$, $n = 4$, …, $n = 16$.

Figure 4 shows the 10 most significant dimensions in the latent space. Curves with different color corresponds to different values of n. We observe that the influence of different n may vary from dimensions to dimensions but the most significantly changed ones concentrate in only 5 dimensions, while other dimensions almost keep intact. This discovery proves our assumption in section 3.2. We conclude that only a small portion of the latent representation contributes to rhythm variation, and we refer to these dimensions as *rhythm representations.*

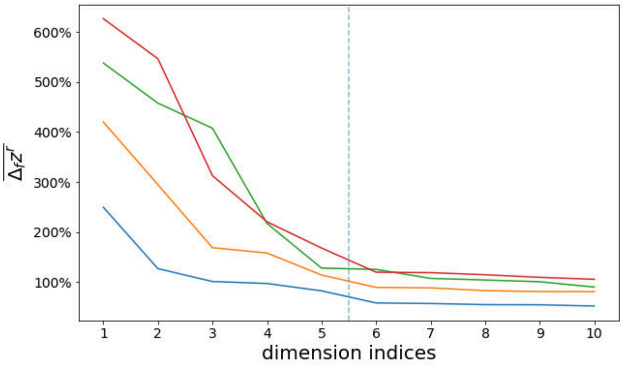


Figure 4: An illustration of only five latent dimensions contribute the most to rhythm variation

3.5 Pitch-Rhythm Disentanglement

In sum, the top 2 dimensions selected in Section 3.3 are referred to as the latent pitch representation, and the top

5 dimensions selected in Section 3.4 are referred to as the latent rhythm representation. We observe that there is no overlap between these two groups of representations.

In our following experiments of interactive composition, pitch and rhythm are modulated only via their corresponding latent representations, while keeping other dimensions fixed.

4. CONTROLLED INTERACTION

Latent space representations encode full information of original music clips, so that modulating latent vectors will result in changes of music clips. Based on the pitch-rhythm disentanglement achieved in section 3, we can interact with the music representation in a human-interpretable and nontrivial ways.

In this section, we use the theme of the "Twelve Variations on 'Ah vous dirai-je, Maman'" (Twinkle, Twinkle, Little Star) by Mozart to illustrate the process of the human-computer interactive composition by controlling the pitch and rhythm representations. Figure 5 shows the piano roll of this music theme.

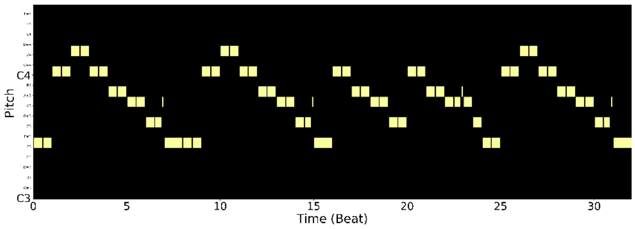


Figure 5: The piano representation of the original sample.

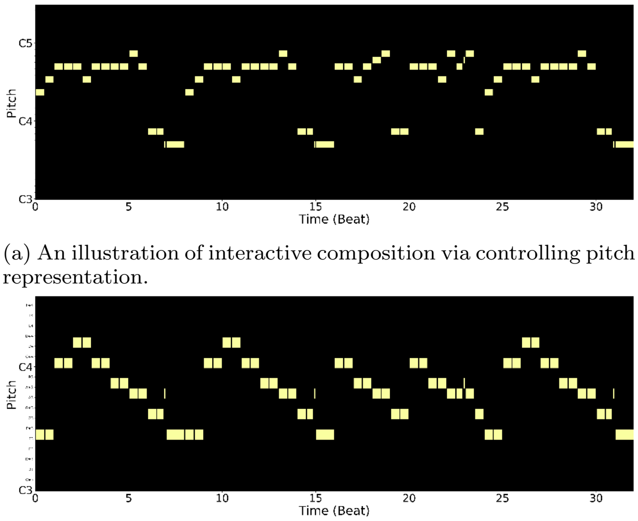
4.1 Pitch Interaction

By performing transformation on pitch representation introduced in section 3.3, we will get the most significant dimensions of the latent space for pitch variation. To create new music via precisely interacting with the pitch representation of the original music, the interface performs the following operations:

1. cut the original music into consecutive 2-bar clips,
2. encode the music clips into latent representation vectors,
3. for each latent vector, only modulate the pitch representations (the most significant dimensions for pitch variation),
4. decode the modified representations back to new music clips, and
5. concatenate the new clips to form a new piece of music.

Figure 6(a) displays the new music theme created via increasing the values of the 2-dimensional pitch representation of the original theme, while figure 6(b) is another created theme by applying the same amount of increment on 2 random dimensions of the latent representation that neither belong to pitch nor rhythm representation.

We can see that the new piece 6(a) significantly increases the overall pitch registration based on the original sample without much modification on the rhythm. In comparison, piece 6(b) hardly changes anything of the original theme. This difference indicates that the our pitch disentanglement is successful. Moreover, piece 6(a) is not merely transposing the original pitches but also creating a new melody with higher pitches and a slightly different pitch contour. An audio version of piece 6(a) can be found at `soundcloud.com/user-705441005/increase-pitch`. This discovery indicates that the interface successfully helps us generate new melodic ideas by controlling the pitch representation.



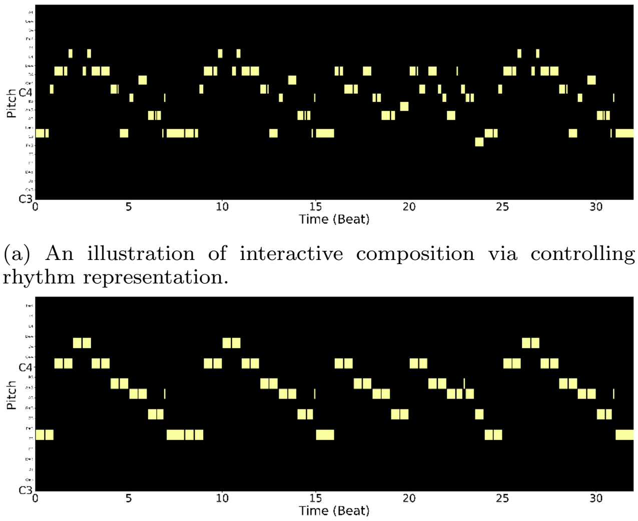
(a) An illustration of interactive composition via controlling pitch representation.

(b) An illustration of interactive composition via controlling 2 random unrelated representation dimensions.

Figure 6: Interactive composition on pitch.

4.2 Rhythm Interaction

To create new music via precisely interacting on the rhythm representation of the original music, the interface performs almost the same five steps as shown in section 4.1. The only difference lies in the third step, where the focus now shifts to the rhythm representation.



(a) An illustration of interactive composition via controlling rhythm representation.

(b) An illustration of interactive composition via controlling 5 random unrelated representation dimensions

Figure 7: Interactive composition on rhythm.

Figure 7(a) displays the new music theme created via increasing the values of the 5-dimensional rhythm representation of the original theme, while figure 7(b) is another created theme by applying the same amount of increment on 5 random dimensions of the latent representation that neither belong to pitch nor rhythm representation.

We see that the rhythm of new piece 7(a) is significantly changed. Another observation is that the pitches are not exactly the same as before. Instead, the pitches change smoothly and follow the original melody trend. This result is reasonable because the disentanglement is meant to happen at the representation level, not the observation level. The smooth transfer also indicates that our model has comprehended the intrinsic nonlinear relationship between pitch and rhythm and thus merges them in an organic way. In comparison, piece 7(b) hardly changes anything of the original theme. This difference indicates that the our rhythm disentanglement is successful. Moreover, the algorithm does not merely cut the notes to produce the new piece 7(a). Instead, it creates a new melody that retains the original melody contour. The music actually sounds like an electronic game version of the original theme, whose audio can be found at `soundcloud.com/user-705441005/increase-note-density`. This discovery indicates that the interface successfully helps us generate new melodic ideas by controlling the rhythm representation.

4.3 Extra Interaction

Besides interacting with pitch and rhythm representations, which leads to interpretable results, we provide an option to modulate the latent dimensions that do not have significant impacts either on pitch or on rhythm. Note that non-significance is not equivalent to uselessness. They indeed contribute to data reconstruction but just in a manner which is hard to be interpreted. Figure 8 shows the result via increasing the value of the non-significant latent dimensions. We see that both melody contour and rhythm patterns are modified, but the result is far less musical compared to ones shown in figure 7a and figure 6a. The audio file can be found at `soundcloud.com/user-705441005/rest-dim-modulate`.

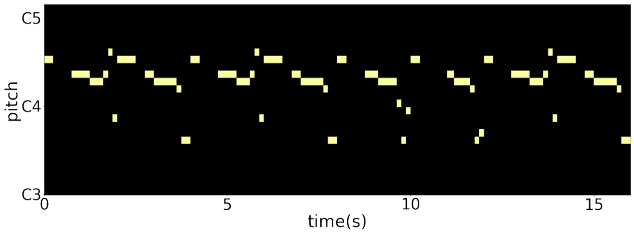


Figure 8: An illustration of interactive composition via controlling non-significant representations

4.4 Two-way Style-transfer Interpolation

Based on the pitch-rhythm disentanglement, the interface also enables a two-way interpolation from one music clip to another in a meaningful way associated with music style. Figure 9 illustrates a grid view of the 2-way interpolation using the SLERP [11] method, where the top-left corner is the *source* and the bottom-right one is the *target*. The pitch interpolation is performed from the top to the bottom, the rhythm interpolation is performed from the left the the right, and the non-significant representations are also interpolated according to the Manhattan distances between a certain location to the target and source, respectively.

Note that the music mainly changes the rhythmic style horizontally and mainly changes the pitch style vertically. We can see this two-way interpolation as a powerful interface for composers to discover new music ideas with intuitive and precise controls on pitch and rhythm styles.

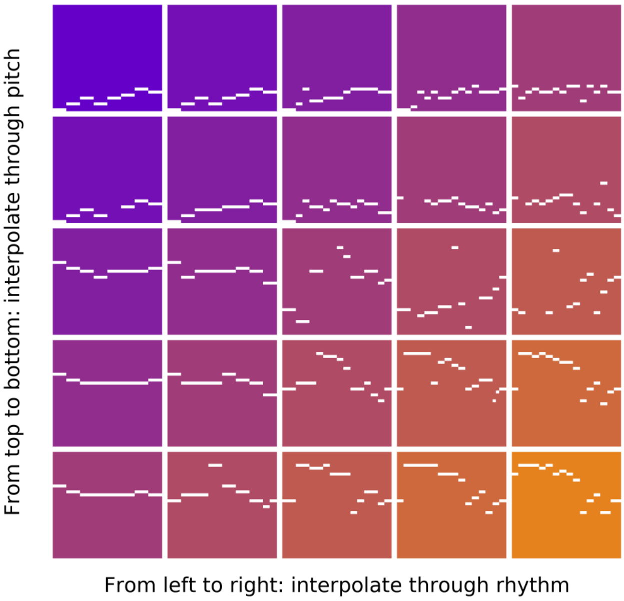


Figure 9: An illustration of two-way pitch-rhythm interpolation

5. CONCLUSION AND FUTURE WORK

In this paper, we inspect the latent space learned by a compressed MusicVAE model, and show that the space can be disentangled, with some dimensions corresponding to pitch change and some other dimensions corresponding to rhythm change. By tweaking values on those selected dimensions, we can modulate average pitch and rhythm complexity, or interpolate two music clips base on pitch or rhythm separately. We further design a user interface to control music creation by performing these operations. Our method still has some limitations. First, it cannot be used to determine the consequences of more complex operations on the latent representation. Second, our inspection is built through manual disentanglement. It would be helpful if an automatic disentangling model for music is designed and this should be a work in the future.

6. REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[2] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer. MIDI-VAE: modeling dynamics and instrumentation of music with applications to style transfer. *CoRR*, abs/1809.07600, 2018.

[3] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[4] S. Dai, Z. Zhang, and G. Xia. Music style transfer issues: A position paper. *CoRR*, abs/1803.06841, 2018.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[6] 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. 2016.

[7] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] A. Roberts, J. Engel, and D. Eck. Hierarchical variational autoencoders for music. In *NIPS Workshop on Machine Learning for Creativity and Design*, 2017.

[11] A. Watt and M. Watt. Advanced animation and bendering techniques. 1992.

[12] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[13] A. Yuille and D. Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308, 2006.