

# DEEP MUSIC ANALOGY VIA LATENT REPRESENTATION DISENTANGLEMENT

Ruihan Yang<sup>1</sup> Dingsu Wang<sup>1</sup> Ziyu Wang<sup>1</sup> Tianyao Chen<sup>1</sup> Junyan Jiang<sup>1,2</sup> Gus Xia<sup>1</sup>

<sup>1</sup> Music X Lab, NYU Shanghai    <sup>2</sup> Machine Learning Department, Carnegie Mellon University

<sup>1</sup>{ry649,dw1920,zz2417,tc2709,gxia}@nyu.edu, <sup>2</sup>junyanj@cs.cmu.edu

## ABSTRACT

*Analogy-making* is a key method for computer algorithms to generate both natural and creative music pieces. In general, an analogy is made by partially transferring the music abstractions, i.e., high-level representations and their relationships, from one piece to another; however, this procedure requires *disentangling* music representations, which usually takes little effort for musicians but is non-trivial for computers. Three sub-problems arise: extracting latent representations from the observation, disentangling the representations so that each part has a unique semantic interpretation, and mapping the latent representations back to actual music. In this paper, we contribute an explicitly-constrained variational autoencoder (EC<sup>2</sup>-VAE) as a unified solution to all three sub-problems. We focus on disentangling the *pitch* and *rhythm* representations of 8-beat music clips conditioned on chords. In producing music analogies, this model helps us to realize the imaginary situation of “*what if*” a piece is composed using a different pitch contour, rhythm pattern, or chord progression by borrowing the representations from other pieces. Finally, we validate the proposed disentanglement method using objective measurements and evaluate the analogy examples by a subjective study.

## 1 Introduction

For intelligent systems, an effective way to generate high-quality art is to produce analogous versions of existing examples [15]. In general, two systems are analogous if they share common abstractions, i.e., high-level representations and their relationships, which can be revealed by the paired tuples  $A : B :: C : D$  (often spoken as A is to B as C is to D). For example, the analogy “the hydrogen atom is like our solar system” can be formatted as *Nucleus : Hydrogen atom :: Sun : Solar system*, in which the shared abstraction is “a bigger part is the center of the whole system.” For generative algorithms, a clever shortcut is to make analogies by solving the problem of “ $A : B :: C : ?$ ”. In the context of music generation, if A is the rhythm pattern of a very lyrical piece B, this analogy can help us realize the

imaginary situation of “what if B is composed with a rather rapid and syncopated rhythm C” by preserving the pitch contours and the intrinsic relationship between pitch and rhythm. In the same fashion, other types of “what if” compositions can be created by simply substituting A and C with different aspects of music (e.g., chords, melody, etc.).

A great advantage of *generation via analogy* is the ability to produce both *natural* and *creative* results. Naturalness is achieved by reusing the representations (high-level concepts such as “image style” and “music pitch contour”) of human-made examples and the intrinsic relationship between the concepts, while creativity is achieved by recombining the representations in a novel way. However, making meaningful analogies also requires *disentangling* the representations, which is effortless for humans but non-trivial for computers. We already see that making analogies is essentially transferring the abstractions, not the observations — simply copying the notes or samples from one piece to another would only produce a casual re-mix, not an analogous composition [11].

In this paper, we contribute an explicitly-constrained conditional variational autoencoder (EC<sup>2</sup>-VAE), a conditional VAE with explicit semantic constraints on intermediate outputs of the network, as an effective tool for learning disentanglement. To be specific, the encoder extracts latent representations from the observations; the semantic constraints disentangle the representations so that each part has a unique interpretation, and the decoder maps the disentangled representations back to actual music while preserving the intrinsic relationship between the representations. In producing analogies, we focus on disentangling and transferring the *pitch* and *rhythm* representations of 8-beat music clips when chords are given as the condition (an extra input) of the model. We show that EC<sup>2</sup>-VAE has three desired properties as a generative model. First, the disentanglement is *explicitly coded*, i.e., we can specify which latent dimensions denote which semantic factors in the model structure. Second, the disentanglement does not sacrifice much of the reconstruction. Third, the learning does not require any analogous examples in the training phase, but the model is capable of making analogies in the inference phase. For evaluation, we propose a new metric and conduct a survey. Both objective and subjective evaluations show that our model significantly outperforms the baselines.



## 2 Related Work

### 2.1 Generation Via Analogy

The history of generation via analogy can trace back to the studies of non-parametric “image analogies” [15] and “playing Mozart by analogy” using case-based reasoning [29]. With recent breakthroughs in artificial neural networks, we see a leap in the quality of produced analogous examples using deep generative models, including music and image style transfer [7, 13], image-to-image translation [18], attribute arithmetic [3], and voice impersonation [12].

Here, we distinguish between two types of analogy algorithms. In a *broad* sense, an analogy algorithm is any computational method capable of producing analogous versions of existing examples. A common and relatively easy approach is supervised learning, i.e., to directly learn the mapping between analogous items from labeled examples [18, 27]. This approach requires little representation learning but needs a lot of labeling effort. Moreover, supervised analogy does not generalize well. For example, if the training analogous examples are all between lyrical melodies (the source domain) and syncopated melodies (the target domain), it would be difficult to create other rhythmic patterns, much less the manipulation of pitch contours. (Though improvements [1, 21, 32] have been made, weak supervision is still needed to specify the source and target domains.) On the other hand, a *strict* analogy algorithm requires not only learning the representations but also disentangling them, which would allow the model to make domain-free analogies via the manipulation of any disentangled representations. Our approach belongs to this type.

### 2.2 Representation Learning and Disentanglement

Variational auto-encoders (VAEs) [22] and generative adversarial networks (GANs) [14] are so far the two most popular frameworks for music representation learning. Both use encoders (or discriminators) and decoders (or generators) to build a bi-directional mapping between the distributions of observation  $x$  and latent representation  $z$ , and both generate new data via sampling from  $p(z)$ . For music representations, VAEs [2, 9, 24, 30] have been a more successful tool so far compared with GANs [31], and our model is based on the previous study [30].

The motivation of representation disentanglement is to better interpret the latent space generated by VAE, connecting certain parts of  $z$  to semantic factors (e.g., age for face images, or rhythm for melody), which would enable a more controllable and interactive generation process. InfoGAN [5] disentangles  $z$  by encouraging the mutual information between  $x$  and a subset of  $z$ .  $\beta$ -VAE [16] and its follow-up studies [4, 20, 30] imposed various extra constraints and properties on  $p(z)$ . However, the disentanglement above are still *implicit*, i.e., though the model separates the latent space into subparts, we cannot define their meanings beforehand and have to “check it out” via *latent space traversal* [3]. In contrast, the disentanglement in Style-based GAN [19], Disentangled Sequential Autoen-

coder [23], and our EC<sup>2</sup>-VAE are *explicit*, i.e., the meanings of different parts of  $z$  are defined by the model structure, so that the controlled generation is more precise and straightforward. The study Disentangled Sequential Autoencoder [23] is most related to our work and also deals with sequential inputs. Using a partially time-invariant encoder, it can approximately disentangle dynamic and static representations. Our model does not directly constrain  $z$  but applies a loss to intermediate outputs associated with latent factors. Such an indirect but explicit constraint enables the model to further disentangle the representation into pitch, rhythm, and any semantic factors whose observation loss can be defined. As far as we know, this is the first disentanglement learning method tailored for music composition.

## 3 Methodology

In this section, we introduce the data representation and model design in detail. We focus on disentangling the latent representations of pitch and rhythm, the two fundamental aspects of composition, over the duration of 8-beat melodies. All data come from the Nottingham dataset [10], regarding a  $\frac{1}{4}$  beat as the shortest unit.

### 3.1 Data Representation

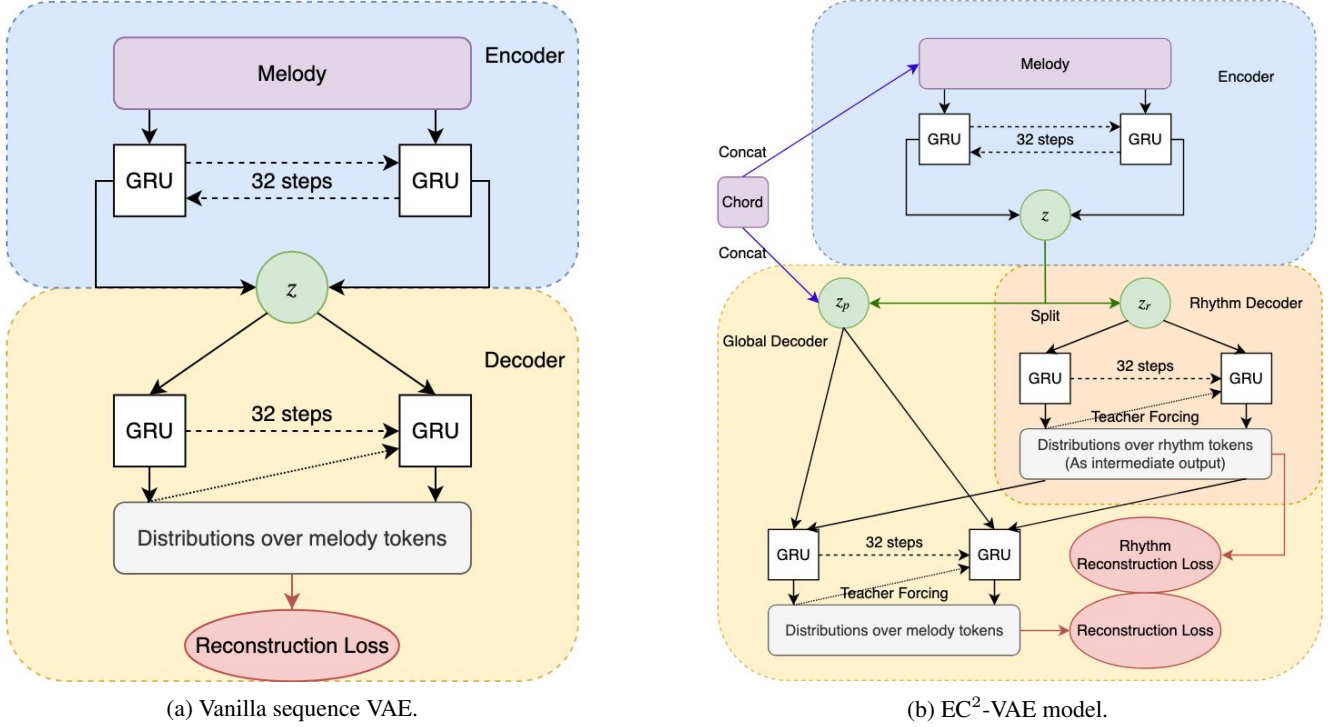
Each 8-beat melody is represented as a sequence of 32 one-hot vectors each with 130 dimensions, where each vector denotes a  $\frac{1}{4}$ -beat unit. As in [24], the first 128 dimensions denote the *onsets* of MIDI pitches ranging from 0 to 127 with one unit of duration. The 129<sup>th</sup> dimension is the *holding* state for longer note duration, and the last dimension denotes *rest*. We also designed a rhythm feature to constrain the intermediate output of the network. Each 8-beat rhythm pattern is also represented as a sequence of 32 one-hot vectors. Each vector has 3 dimensions, denoting: an onset of any pitch, a holding state, and rest.

Besides, chords are given as a condition, i.e., an extra input, of the model. The chord condition of each 8-beat melody is represented as a chromagram with equal length, i.e., 32 multi-hot vectors each with 12 dimensions, where each dimension indicates whether a pitch class is activated.

### 3.2 Model Architecture

Our model design is based on the previous studies of [24, 30], both of which used VAEs to learn the representations of fixed-length melodies. Figure 1 shows a comparison between the model architectures, where Figure 1(a) shows the model designed in [30] and Figure 1(b) shows the model design in this study. We see that both use bi-directional GRUs [6] (or LSTMs [17]) as the encoders (in blue) to map each melody observation to a latent representation  $z$ , and both use uni-directional GRUs(LSTMs) (with teacher forcing [26] in the training phrase) as the decoders (in yellow) to reconstruct melodies from  $z$ .

The key innovation of our model design is to assign a part of the decoder (in orange) with a specific subtask: to disentangle the latent rhythm representation  $z_r$  from the overall  $z$  by explicitly encouraging the intermediate output of  $z_r$  to match the rhythm feature of the melody. The other



**Figure 1:** A comparison between vanilla sequence VAE [30] and our model with condition and disentanglement.

part of  $z$  is therefore everything but rhythm and interpreted as the latent pitch representation,  $z_p$ . Note that this explicitly coded disentanglement technique is quite flexible — we can use multiple subparts of the decoder to disentangle multiple semantically interpretable factors of  $z$  simultaneously as long as the intermediate outputs of the corresponding latent factors can be defined, and the model shown in Figure 1(b) is the simplest case of this family.

It is also worth noting that the new model uses chords as a condition for both the encoder and decoder. The advantage of chord conditioning is to free  $z$  from storing chord-related information. In other words, the pitch information in  $z$  is “detrended” by the underlying chord for better encoding and reconstruction. The cost of this design is that we cannot learn a latent distribution of chord progressions.

### 3.2.1 Encoder

A single layer bi-directional GRU with 32 time steps is used to model  $Q_\theta(z|x, c)$ , where  $x$  is the melody input,  $c$  is the chord condition, and  $z$  is the latent representation. Chord conditions are given by concatenating with the input at each time step.

### 3.2.2 Decoder

The global decoder models  $P_\phi(x|z, c)$  by multiple layers of GRUs, each with 32 steps. For disentanglement, we split the latent representation  $z$  into two halves  $z_p$  and  $z_r$ , each being a 128-dimensional vector. As a subpart of the global decoder, the rhythm decoder models  $P_{\phi_r}(r(x)|z)$  by a single layer GRU, where  $r(x)$  is the rhythm feature of the melody. Meanwhile, the rhythm is concatenated with  $z_p$  and chord condition as the input of the rest of the global decoder to reconstruct the melody. We used cross-entropy

loss for both rhythm and melody reconstruction. Note that the overall decoder is supposed to learn non-trivial relationships between pitch and rhythm, rather than naively cutting a pitch contour by a rhythm pattern.

### 3.3 Theoretical Justification of the ELBO Objective with Disentanglement

One concern about representation disentanglement techniques is that they sometimes sacrifice reconstruction power [20]. In this section, we prove that our model does not suffer much of the disentanglement-reconstruction paradox, and the likelihood bound of our model is close to that of the original conditional VAE, and in some cases, equal to it.

Recall the Evidence Lower Bound (ELBO) objective function used by a typical conditional VAE [8] constraint on input sample  $x$  with condition  $c$ :

$$\begin{aligned} \text{ELBO}(\phi, \theta) &= \mathbb{E}_Q[\log P_\phi(x|z, c)] \\ &\quad - \mathbb{KL}[Q_\theta(z|x, c) \| P_\phi(z|c)] \leq \log P_\phi(x|c) \end{aligned}$$

For simplicity,  $\mathcal{D}$  denotes  $\mathbb{KL}[Q_\theta(z|x, c) \| P_\phi(z|c)]$  in the rest of this section. If we see the intermediate rhythm output in Figure 1(b) as hidden variables of the whole network, the new ELBO objective of our model only adds the rhythm reconstruction loss based on the original one, resulting in a lower bound of the original ELBO. Formally,

$$\begin{aligned} \text{ELBO}^{\text{new}}(\phi, \theta) &= \mathbb{E}_Q[\log P_\phi(x|z, c)] - \mathcal{D} + \mathbb{E}_Q[\log P_{\phi_r}(r(x)|z_r)] \\ &= \text{ELBO}(\phi, \theta) + \mathbb{E}_Q[\log P_{\phi_r}(r(x)|z_r)] \end{aligned}$$

where  $\phi_r$  denotes parameters of the rhythm decoder. Clearly,  $\text{ELBO}^{\text{new}}$  is a lower bound of the original ELBO because  $\mathbb{E}_Q[\log P_{\phi_r}(r(x)|z_r)] \leq 0$ .

Moreover, if the rest of global decoder takes the original rhythm rather than the intermediate output of rhythm decoder as the input, the objective can be rewritten as:

$$\begin{aligned}
 & \text{ELBO}^{\text{new}}(\phi, \theta) \\
 &= \mathbb{E}_Q[\underbrace{\log P_\phi(x|r(x), z_p, c) + \log P_\phi(r(x)|z_r, c)}_{\text{with } x \perp\!\!\!\perp z_r | r(x), c \text{ and } r(x) \perp\!\!\!\perp z_p}] - \mathcal{D} \\
 &= \mathbb{E}_Q[\log P_\phi(x, r(x)|z, c)] - \mathcal{D} \\
 &= \mathbb{E}_Q[P_\phi(x|z, c) + \log P_\phi(r(x)|x, z, c)] - \mathcal{D} \\
 &= \text{ELBO}(\phi, \theta)
 \end{aligned}$$

The second equal sign holds for a perfect disentanglement, and the last equal sign holds since  $r(x)$  is decided by  $x$ , i.e.,  $P_\phi(r(x)|x, z, c) = 1$ . In other words, we show that under certain assumptions ELBO<sup>new</sup> with disentanglement is identical to the ELBO.

## 4 Experiments

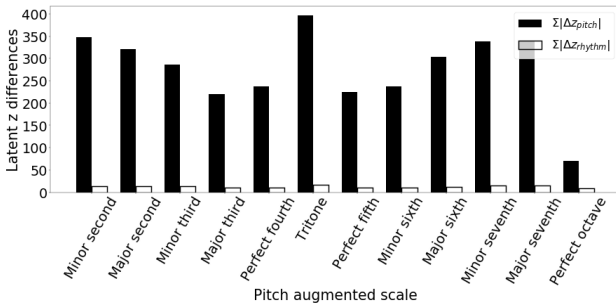
We present the objective metrics to evaluate the disentanglement in Section 4.1, show several representative examples of generation via analogy in Section 4.2, and use subjective evaluations to rate the artistic aspects of the generated music in Section 4.3.

### 4.1 Objective Measurements

Upon a successful pitch-rhythm disentanglement, any changes in pitch of the original melody should not affect the latent rhythm representation much, and vice versa. Following this assumption, we developed two measurements to evaluate the disentanglement: 1)  $\Delta z$  after transposition, which is more qualitative, and 2) F-score of an augmentation-based query, which is more quantitative.

#### 4.1.1 Visualizing $\Delta z$ after transposition

We define  $F_i$  as the operation of transposing all the notes by  $i$  semitones, and use the  $L_1$ -norm to measure the change in  $z$ . Figure 2 shows a comparison between  $\Sigma|\Delta z_p|$  and  $\Sigma|\Delta z_r|$  when we apply  $F_i$  to a randomly chosen piece (where  $i \in [1, 12]$ ) while keeping the rhythm and underlying chord unchanged.



**Figure 2:** A comparison between  $\Delta z_p$  and  $\Delta z_r$  after transposition.

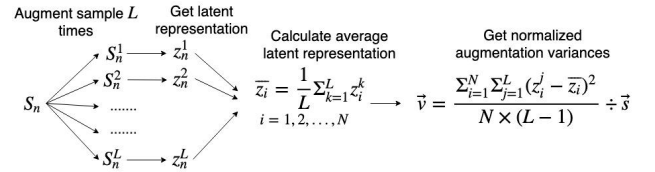
Here, the black bars stand for  $\Sigma|\Delta z_p|$  and the white bars stand for the  $\Sigma|\Delta z_r|$ . It is conspicuous that when augmenting pitch, the change of  $z_p$  is much larger than the change of  $z_r$ , which well demonstrates the success of the disentanglement.

It is also worth noting that the change of  $z_p$  to a certain extent *reflects human pitch perception*. Given a chord, the change in  $z_p$  can be understood as the “burden” (or difficulty) to memorize (or encode) a transposed melody. We see that such burden is large for tritone (very dissonant), relatively small for major third, perfect fourth & fifth (consonant), and very small for perfect octave.

Due to the space limit, we only show the visualization of the latent space when changing the pitch. According to the data representation in Section 3.1, changing the rhythm feature of a melody would inevitably affect the pitch contour, which would lead to complex behavior of the latent space hard to interpret visually. We leave the discussion for future work but will pay more attention to the effect of the rhythm factor in Section 4.3.

#### 4.1.2 F-score of Augmentation-based Query

The explicitly coded disentanglement enables a new evaluation method from an *information-retrieval* perspective. We regard the pitch-rhythm split in  $z$  defined by the model structure as the *reference* (the ground truth), the operation of factor-wise data augmentation (keeping the rhythm and only changing pitch randomly, or vice versa) as a *query* in the latent space, and the actual latent dimensions having the largest variance caused by augmentation as the *result set*. In this way, we can quantitatively evaluate our model in terms of precision, recall, and F-score.



**Figure 3:** Evaluating the disentanglement by data augmentation.

	Pitch			Rhythm		
	pre.	rec.	F-s.	pre.	rec.	F-s.
EC <sup>2</sup> -VAE	<b>0.88</b>	<b>0.88</b>	<b>0.88</b>	<b>0.80</b>	<b>0.80</b>	<b>0.80</b>
Random	0.5	0.5	0.5	0.5	0.5	0.5

**Table 1:** The evaluation results of pitch- and rhythm-wise augmentation-based query.

Figure 3 shows the detailed query procedure, which is a modification of the evaluation method in [20]. After pitch or rhythm augmentation for each sample,  $\vec{v}$  is calculated as the average (across the samples) variance (across augmented versions) of the latent representations, normalized by the total sample variance  $\vec{s}$ . Then, we choose the first half (128 dimensions) with the largest variances as the result set. This precision, recall and F-score of this augmentation-based query result is shown in Table 1. (Here, precision and recall are identical since the size of the result set equals the dimensionality of  $z_p$  and  $z_r$ .) As this is the first tailored metric for explicitly coded disentanglement, we use random guess as our baseline.

## 4.2 Examples of Generation via Analogy

We present several representative “what if” examples by swapping or interpolating the latent representations of different pieces. Throughout this section, we use the following example (shown in Figure 4), an 8-beat melody from the Nottingham Dataset [10] as the source, and the target rhythm or pitch will be borrowed from other pieces. (MIDI demos are available at <https://github.com/cdyrhjohn/Deep-Music-Analogy-Demos>.)



Figure 4: The source melody.

### 4.2.1 Analogy by replacing $z_p$

Two examples are presented. In both cases, the latent pitch representation and the chord condition of the source melody are replaced with new ones from other pieces. In other words, the model answers the analogy question: “source’s pitch : source melody :: target’s pitch : ?”

Figure 5 shows the first example, where Figure 5(a) shows the piece from which the pitch and chords are borrowed, and Figure 5(b) shows the generated melody. From Figure 5(a), we see the target melody is in a different key (D major) with a larger pitch range than the source and a big pitch jump in the beginning. From Figure 5(b), we see the generated new melody captures such pitch features while keeping the rhythm of the source unchanged.



(a) Target’s pitch and chord.

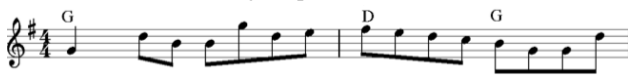


(b) The generated target music, using the pitch and chord from (a) and the rhythm from the source.

Figure 5: The 1<sup>st</sup> example of analogy via replacing  $z_p$ .



(a) Target’s pitch and chord.



(b) The generated target, using the pitch and chord from (a) and the rhythm from the source.

Figure 6: The 2<sup>nd</sup> analogy example via replacing  $z_p$ .

Figure 6 shows another example, whose subplots share the same meanings with the previous one. From Figure 6(a), we see the first measure of the target’s melody is a broken chord of Gmaj, while the second measure is the G major scale. From Figure 6(b), we see the generated new melody captures these pitch features. Moreover, it retains

the source’s rhythm and ignores the dotted eighth and sixteenth notes in Figure 6(a).

### 4.2.2 Analogy by replacing $z_r$

Similar to the previous section, this section shows two example answers to the question: “source’s rhythm : source melody :: target’s rhythm : ?” by replacing  $z_r$ . Figure 7 shows the first example, where Figure 7(a) contains the new rhythm pattern quite different from the source, and Figure 7(b) is the generated target. We see that Figure 7(b) perfectly inherited the new rhythm pattern and made minor but novel modifications based on the source’s pitch.



(a) Target’s rhythm pattern.



(b) The generated target music, using the rhythm of (a) while keeping source’s pitch and chord.

Figure 7: The 1<sup>st</sup> example of analogy via replacing  $z_r$ .



(a) Target’s rhythm pattern.



(b) The generated target music, using the rhythm of (a) while keeping source’s pitch and chord.

Figure 8: The 2<sup>nd</sup> analogy example via replacing  $z_r$ .

Figure 8 shows a more extreme case, in which Figure 8(a) contains only 16th notes of the same pitch. Again, we see the generated target in Figure 8(b) maintains the source’s pitch contour while matching the given rhythm pattern.

### 4.2.3 Analogy by Replacing Chord



(a) Changing all the chords down a semitone, resulting in the key change from G major to Bb minor.



(b) Changing the key from G major to G minor.

Figure 9: Two examples of replacing the original chord.

Though chord is not our main focus, here we show two analogy examples in Figure 9 to answer “what if” the source melody is composed using some other chord progressions. Figure 9(a) shows an example where the key is Bb minor. An interesting observation is the new melody

contour indeed adds some reasonable modification (e.g. flipping the melody) rather than simply transposing down all the notes. It brings us a little sense of Jazz. Figure 9(b) shows an example where the key is changed from G major to G minor. We see melody also naturally transforms from major mode to minor mode.

#### 4.2.4 Two-way Pitch-Rhythm Interpolation

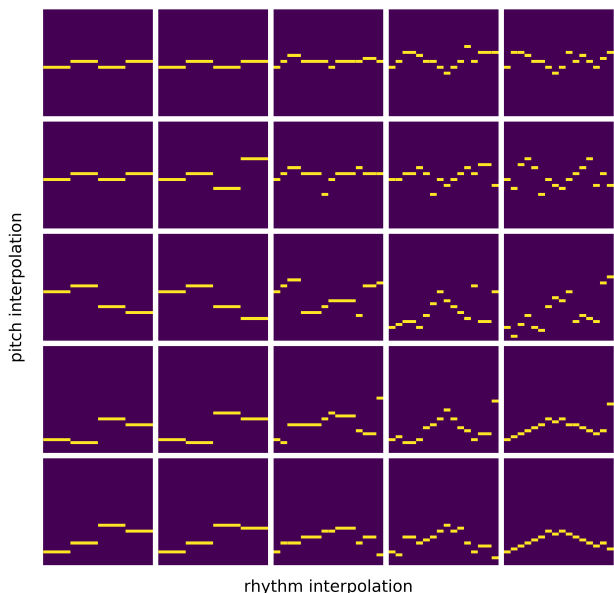


Figure 10: An illustration of two-way interpolation.

The disentanglement also enables a smooth transition from one music to another. Figure 10 shows an example of two-way interpolation, i.e., a traversal over a subspace of the learned latent representations  $z_r$  and  $z_p$  along 2 axes respectively, while keeping the chord as NC (no chord). Here, each square is a piano-roll of an 8-beat music. The top-left (source) and bottom-right (target) squares are two samples created manually and everything else is generated by interpolation using SLERP [28]. Note that the rhythmic changes are primarily observed moving along the “rhythm interpolation” axis, and likewise for pitch and the vertical “pitch interpolation” axis.

### 4.3 Subjective Evaluation

Besides objective measurement, we conducted a subjective survey to evaluate the quality of generation via analogy. We focus on changing the rhythm factors of existing music since this operation leads to an easier identification of the source melodies.

Each subject listened to two groups of five pieces each. All the pieces had the same length (64 beats at 120 bpm). Within each group, one piece was an original, human-composed piece from the Nottingham dataset, having a lyrical melody consisting of longer notes. The remaining four pieces were variations upon the original with more rapid rhythms consisting of 8<sup>th</sup> and 16<sup>th</sup> notes. Two of the variations were produced in a rule-based fashion by naively cutting the notes in the original into shorter subdivisions, serving as the *baseline*. The other two variations were gen-

erated with our EC<sup>2</sup>-VAE by merging the  $z_p$  of the original piece and the  $z_r$  decoded from two pieces having the same rhythm pattern as the baselines but with all notes replaced with “do” (similar to Figure 8(a)). The subjects always listened to the original first, and the order of the variations was randomized. In sum, we compare three versions of music: 1) the original piece, 2) the variation created by the baseline, and, 3) the variation created by our algorithm. The subjects were asked to rate each sample on a 5-point scale from 1 (very low) to 5 (very high) according to three criteria:

1. *Creativity*: how creative the composition is.
2. *Naturalness*: how human-like the composition is.
3. *Overall musicality*.

A total of 30 subjects (16 female and 14 male) participated in the survey. Figure 11 shows the results, where the heights of bars represent means of the ratings the and error bars represent the MSEs computed via within-subject ANOVA [25]. The result shows that our model performs significantly better than the rule-based baseline in terms of creativity and musicality ( $p < 0.05$ ), and marginally better in terms of naturalness. Our proposed method is even comparable to the original music in terms of creativity, but remains behind human composition in terms of the other two criteria.

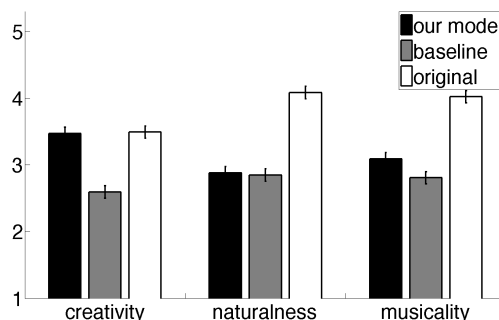


Figure 11: Subjective evaluation results.

## 5 Conclusion

In conclusion, we contributed an explicitly-constrained conditional variational autoencoder (EC<sup>2</sup>-VAE) as an effective disentanglement learning model. This model generates new music via making analogies, i.e., to answer the imaginary situation of “what if” a piece is composed using different pitch contours, rhythm patterns, and chord progressions via replacing or interpolating the disentangled representations. Experimental results showed that the disentanglement is successful and the model is able to generate interesting and musical analogous versions of existing music. We see this study a significant step in music understanding and controlled music generation. The model also has the potential to be generalized to other domains, shedding light on the general scenario of generation via analogy.

## 6 Acknowledgement

We thank Yun Wang, Zijian Zhou and Roger Dannenberg for the in-depth discussion on music disentanglement and analogy. This work is partially supported by the Eastern Scholar Program of Shanghai.

## 7 References

- [1] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [2] Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer. *arXiv preprint arXiv:1809.07600*, 2018.
- [3] Shan Carter and Michael Nielsen. Using artificial intelligence to augment human intelligence. *Distill*, 2(12):e9, 2017.
- [4] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in vaes. *NIPS*, 2018.
- [5] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.
- [6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [7] Shuqi Dai, Zheng Zhang, and Gus G Xia. Music style transfer: A position paper. *arXiv preprint arXiv:1803.06841*, 2018.
- [8] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.
- [9] Philippe Esling, Axel Chemla-Romeu-Santos, and Adrien Bitton. Bridging audio analysis, perception and synthesis with perceptually-regularized variational timbre spaces. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR*, pages 23–27, 2018.
- [10] E. Foxley. Nottingham database, 2011.
- [11] Y Gao. *Towards neural music style transfer*. PhD thesis, Master Thesis, New York University. [https://github.com/821760408-sp/the ...](https://github.com/821760408-sp/the...), 2017.
- [12] Yang Gao, Rita Singh, and Bhiksha Raj. Voice impersonation using generative adversarial networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2506–2510. IEEE, 2018.
- [13] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.
- [14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [15] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM, 2001.
- [16] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, volume 3, 2017.
- [17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.
- [20] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.
- [21] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1857–1865. JMLR. org, 2017.
- [22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [23] Yingzhen Li and Stephan Mandt. Disentangled sequential autoencoder. *arXiv preprint arXiv:1803.02991*, 2018.
- [24] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. *arXiv preprint arXiv:1803.05428*, 2018.
- [25] Henry Scheffe. *The analysis of variance*, volume 72. John Wiley & Sons, 1999.

- [26] Nikzad Benny Toomarian and Jacob Barhen. Learning a trajectory using adjoint functions and teacher forcing. *Neural Networks*, 5(3):473–484, 1992.
- [27] Christopher J Tralie. Cover song synthesis by analogy. *arXiv preprint arXiv:1806.06347*, 2018.
- [28] Alan Watt and Mark Watt. Advanced animation and bending techniques. 1992.
- [29] Gerhard Widmer and Asmir Tobudic. Playing mozart by analogy: Learning multi-level timing and dynamics strategies. *Journal of New Music Research*, 32(3):259–268, 2003.
- [30] Ruihan Yang, Tianyao Chen, Yiyi Zhang, and Gus Xia. Inspecting and interacting with meaningful music representations using vae. *arXiv preprint arXiv:1904.08842*, 2019.
- [31] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [32] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.