TRANSFORMER VAE: A HIERARCHICAL MODEL FOR STRUCTURE-AWARE AND INTERPRETABLE MUSIC REPRESENTATION LEARNING

Junyan Jiang[1,2], *Gus G. Xia*[2], *Dave B. Carlton*[3], *Chris N. Anderson*[3], *Ryan H. Miyakawa*[3]

[1]School of Computer Science, Carnegie Mellon University
[2] Music X Lab, New York University Shanghai
[3] Hooktheory, LLC

ABSTRACT

Structure awareness and *interpretability* are two of the most desired properties of music generation algorithms. Structure-aware models generate more natural and coherent music with long-term dependencies, while interpretable models are more friendly for human-computer interaction and co-creation. To achieve these two goals simultaneously, we designed the *Transformer Variational AutoEncoder*, a hierarchical model that unifies the efforts of two recent breakthroughs in deep music generation: 1) the Music Transformer and 2) Deep Music Analogy. The former learns long-term dependencies using attention mechanism, and the latter learns interpretable latent representations using a disentangled conditional-VAE. We showed that Transformer VAE is essentially capable of learning a context-sensitive hierarchical representation, regarding local representations as the context and the dependencies among the local representations as the global structure. By interacting with the model, we can achieve *context transfer*, realizing the imaginary situation of "what if" a piece is developed following the music flow of another piece.

Index Terms— Representation learning, VAE, Transformer, music structure

1. INTRODUCTION

With the development of deep generative models [1], we have recently seen a lot of progress in computational creativity [2] regarding signals of various modalities, including image [3, 4], text [5, 6], and music [7, 8]. For music generation models, two of the most desired properties are *structure awareness* and *interpretability*. Structure-aware models have the potential to generate natural and coherent music with long-term dependencies, including repetition and variation at different hierarchies [9]. Interpretability, on the other hand, is the key to turn complex computational models into controllable interfaces for interactive music performance [10] or co-composition purposes.

To this end, two research mainstreams are observed. On the one hand, the deep music analogy study [11, 12] deals with model interpretability under a Variational Auto-Encoder (VAE) framework [8] via *multi-task constraints* and *conditioning*. However, VAE-based models cannot yet handle time-series structures well, and the performance decreases significantly for long-term (e.g., phrase-level) music [8]. On the other hand, attentive generative models such as Music Transformer [7] and MuseNet [13] are built to deal with long-term structures. However, such models' latent states are not human interpretable. Some works try to introduce VAEs to attentive models [14], but not for the purpose of interpretability.

Our goal is to achieve *both* structure awareness and interpretability via unifying the approaches above. In this study, we contribute *Transformer Variational AutoEncoder*, a hierarchical model for long-term melody representation learning. The model is composed of two layers. The bottom layer consists of multiple local encoders working in parallel, each learning the latent representation of a measure (bar). The top layer uses masked attention blocks (by letting latter bars pay attention to earlier ones) to extract the global structure, i.e., dependencies among the bars. Therefore, all local representations essentially serve as the "contextual condition" for succeeding bars. This mechanism frees the VAE from memorizing redundant information, and essentially offers:

1. An effective way to memorize structured long-term signals;
2. a *context-sensitive* representation learning method;
3. an interactive generation process via *context transfer*, helping us realize the imaginary situation of "what if" a piece is developed following the structure of another piece.

Experiment results show that our method not only achieves satisfied reconstructions for phrase-level music but also is capable of transferring melodic and rhythmic contexts from one phrase to another.

2. PROPOSED METHOD

2.1. Transformer VAE

Fig. 1 shows an overview of our proposed model. The model is heavily based on the vanilla Transformer proposed by [6].

The vanilla Transformer specializes in seq-2-seq prediction, including machine translation [15] and music generation [7]. We made some changes to make the vanilla Transformer model capable of *representation learning* under a VAE setting, i.e., the model input and output are identical. Specifically, (1) we add Gaussian noises to the output of the encoder before feeding it to the decoder, and (2) we adopt a masked attention mechanic for both the encoder and the decoder.

Formally, let $x_{1..T}$ denotes the original melody with T bars (measures) in length, where x_i represents the melody of the i-th bar. First, every bar goes through a local encoder E_{local} with shared parameters to get the bar-level representations $h^e_{1..T}$, where $h^e_i = E_{\text{local}}(x_i)$. Then, the Transformer encoder takes $h^e_{1..T}$ as inputs and calculate the parameters of the latent code z:

$$g^e_{1..T} = E_{\text{Transformer}}(h^e_{1..T}) \tag{1}$$

$$[\mu_i, \log(\sigma_i^2)] = \mathbf{W}_i g^e_i + b_i \tag{2}$$

$$z_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \tag{3}$$

The latent variable z is then fed into the Transformer decoder $D_{\text{Transformer}}$, together with the representation of previously decoded bars to decode the new bars sequentially. Formally,

$$g^d_{1..i} = D_{\text{Transformer}}(z_{1..i}, h^d_{0..i-1}) \tag{4}$$

$$\hat{x}_i = D_{\text{local}}(g^d_i) \tag{5}$$

where the $\hat{x}_{1..T}$ are the decoded bars and $h^d_i = E_{\text{local}}(\hat{x}_i)$ are their bar-level representation. Here, h^d_0 is a special embedding for the start of music, which functions similar to the *Start Of Sentence* (SOS) tag in natural language processing.

2.2. Context-sensitive Representation

The self-attention mechanism [6] of the Transformer encoder allows the representation z_i to contain not only information in h^e_i but also the contextual information from other bars, $h^e_j, j \neq i$. From the perspective of representation learning, such contextual information can help reduce the redundancy in $z_{1..T}$ and learn a context-sensitive representation. For example, when the input melody satisfies $x_1 = x_5$ (i.e., exact repetition), the model can capture this information when encoding the 5th bar by attending to the 1st bar. If the bar's content is already stored in z_1, z_5 does not have to store the same information again, but rather a simple structural description such as "x_5 *is the same as the* x_1".

It is worth noting that z_5 itself now becomes insufficient for reconstructing x_5, but the attention layers in the decoder will help the 5th bar fetch back the information from the 1st bar and complete the reconstruction. In general, if we change the context of some bars, we expect the whole reconstructed music to be changed accordingly. In other words, we can achieve *context transfer*, realizing imaginary aesthetic questions such as "what if the 1st bar of piece A is developed following the music flow (structure) of piece B".

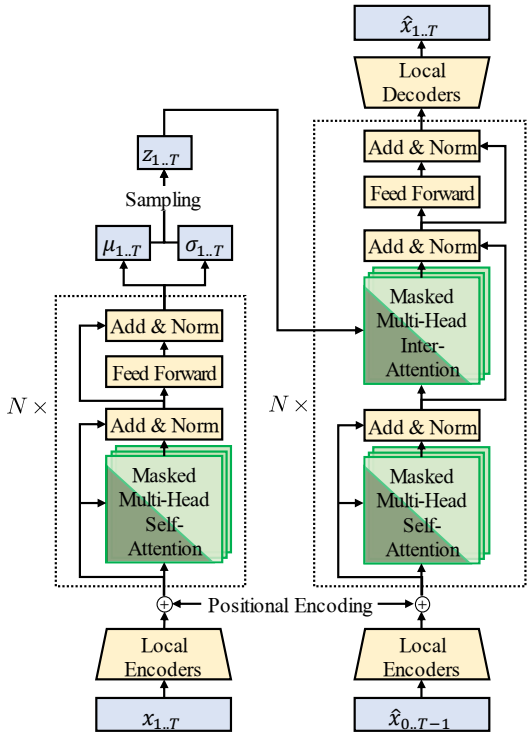

Fig. 1: An overview of the model.

2.3. Dependency Control via Masked Attention

The attention mask plays an important role in dependency control of variables. The Transformer VAE uses the same attention calculation methods as the original Transformer [6]:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}(\mathbf{M} \circ \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}})\mathbf{V}, \tag{6}$$

Here, $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ represent the query, key and value matrix, and d_k represents the row count of $\mathbf{K}$. $\mathbf{M}$ is the attention mask and "$\circ$" denotes the element-wise multiplication. Under the context of Transformer, $\mathbf{K}$ and $\mathbf{V}$ are some linear transformations of the same value vectors $v_{1..T}$, i.e., $\mathbf{K} = [v_1, ..., v_T]^\top \mathbf{W}_k$, $\mathbf{V} = [v_1, ..., v_T]^\top \mathbf{W}_v$, and $\mathbf{Q} = [q_1, ..., q_T]^\top \mathbf{W}_q$ where $q_{1..T}$ are the query vectors of the same length. The output vectors are given by $[o_1, ...o_T] = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})^\top$. When no attention mask is applied ($\mathbf{M} = \mathbf{1}_{T\times T}$), each output token o_i is influenced by all values in $v_{1..T}$ (Fig. 2 (a)). To control the dependency, a masking function is introduced to set some of the entries to $-\infty$. Fig. 2 shows some common attention masks.

A common way to use masking, as proposed in the original Transformer [6], is to multiply the decoder's self-attention by an upper triangular matrix (i.e., Fig. 2 (b)). This operation prevents the information leakage of future results to previous steps during teacher-forcing training. However, if our model only adopts this mask, the interpretation of the latent code $z_{1..T}$ will be ambiguous. Referring to the $x_1 = x_5$ example

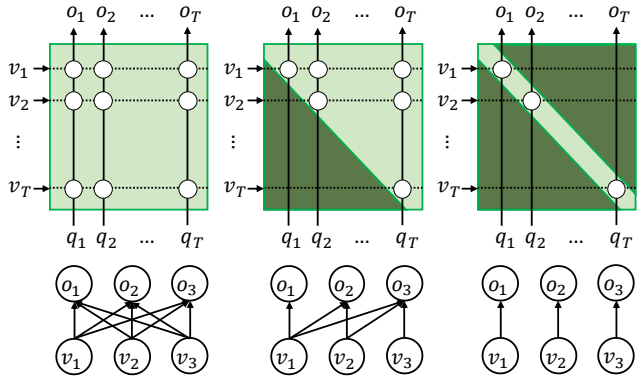

Fig. 2: Three types of attention masks: (a) no mask, (b) upper triangular mask and (c) diagonal mask (degenerated attention). The figures below show the corresponding dependency graphs when $T = 3$.

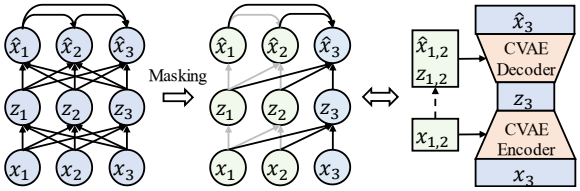

Fig. 3: A conditional VAE view of the Transformer VAE when $T = 3$. The left shows the change of the dependency graph after we apply the attention mask. The right shows a theoretically equivalent conditional VAE for the 3^{rd} bar, where the previous bars serve as conditions.

used in the last section, though we know that the information will be stored merely once, we have *no* clue of whether it will be stored in z_1, z_5, or half in z_1 and half in z_5.

We resolve this ambiguity issue by introducing upper triangular masks (Fig. 2 (b)) for all attention layers in the encoder and the decoder. This operation serves as a *dependency control*: for the i-th bar, (1) the encoder can only access the context to its left ($x_{1..i}$), and (2) the decoder can only access the context and decoded bars to its left ($\hat{x}_{1..i-1}$ and $z_{1..i}$). Hence, the model will learn to store the information of the repeated bars only on its first occurrence, which makes the structure interpretation unambiguous.

2.4. A Conditional VAE View

To get a deeper understanding of the Transformer VAE, one can adopt a conditional VAE's viewpoint. The *condition* of a VAE is some information provided at both the encoding and decoding phases [16]. Ideally, to encode the signal x, a VAE can utilize the conditional information c without memorizing it in the latent representation z. In a similar sense, we can regard the model as the combination of T distinct 1-bar VAEs, with the first one unconditioned and all remaining ones conditioned on their previous context (see Fig. 3 for an example). This is the reason why the redundancy in the latent representation $z_{1..T}$ is explicitly removed.

Also, if different conditions are provided at the decoding phase, the decoded results will be changed even if the latent code z_i remains the same. Therefore, the decoding of each bar is context-sensitive.

3. EXPERIMENTS

3.1. Dataset

We use a dataset provided by Hooktheory[1], a crowd-annotated data source for popular music transcription for the experiment. Our dataset contains 16,142 8-bar pieces with a 4/4 meter (corresponding to the most common format on Hooktheory). We use 80% songs for training and 20% for testing.

The dataset does not cover all possible MIDI pitches or have very few samples for certain MIDI pitches. Therefore, we first perform data augmentation on the training set within the range of -4 to 4 semitones. Also, we adopt a valid pitch range from MIDI pitch 40 to 84. After data augmentation, all notes not included in the pitch range are removed. The pieces are represented by a sequence of tokens. Each token represents a time step of a sixteenth note (16 tokens per bar). Possible token values include 45 onset states (each corresponds to a valid MIDI pitch), 1 sustain state and 1 silence state.

3.2. Model Hyperparameters and Training

We implemented the Transformer VAE model with a latent dimension $\dim(z_i) = 64$ for each bar. The Transformer encoder (decoder) consists of $N = 3$ stacked identical encoder (decoder) layers with a hidden dimension $\dim(h_i^e) = \dim(h_i^d) = 256$. The local encoder and the local decoder both contain 3 fully-connected layers with the hidden dimension of $[512, 384]$ for the local encoder and $[384, 512]$ for the local decoder. Tokens are converted into the one-hot format for input. A softmax function is applied to the output of the local decoder to calculate token probabilities.

We adopt a β-VAE loss function [17] for the model, as shown in Eqn. 7.

$$\mathcal{L}_\theta(x, z, \hat{x}) = \mathcal{L}_{\text{recons}}(x, \hat{x}) + \beta \text{KL}[q(z|x)||p(z)] \quad (7)$$

Here, $\mathcal{L}_{\text{recons}}$ is the reconstruction loss. We use an average cross entropy loss for each token here. The value β controls the degree of KL divergence penalty. We choose $\beta = 1$ for our models. The model is trained using the Adam optimizer [18] with a batch size of 16. The model is trained for 60 epochs with a scheduled learning rate (30 epochs for 1e-4, 20 epochs for 1e-5 and 10 epochs for 1e-6).

[1] https://www.hooktheory.com/

	Recons. (Teacher-Forcing)	Recons. (Sampling)	KL Loss
Proposed+A	0.9765	0.9716	**0.2592**
Proposed-A	0.9761	0.9753	0.3366
1×8-bar LSTM	0.8859	0.8307	0.3132
8×1-bar LSTM	**0.9962**	**0.9912**	0.3244

Table 1: Comparitive Results. “Recons." denotes the reconstruction accuracy on the test set. “KL Loss" denotes the average KL distance term on the test set.

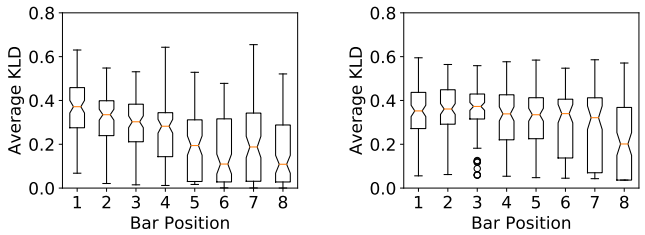

Fig. 4: Box plots of KL loss on different bar positions of the proposed model (a) with attention; (b) without attention.

3.3. Comparative Results

We first performed a comparative evaluation over the following methods: (1) **Proposed+A**: the Transformer VAE with regular upper-triangular-masked attention. (2) **Proposed-A**: The Transformer VAE without the attention mechanism. In other words, We use a diagonal mask (as in Fig. 2 (c)) to degenerate the attention matrix. (3) **1×8-bar LSTM**: The baseline MusicVAE model [8] with a hierarchical Long Short-Term Memory (LSTM) decoder, which directly encodes an 8-bar melody into a 512-dimension latent codes. In other words, this model aims to memorize a long-term piece in a brute-force way. (4) **8×1-bar LSTM**: 8 short-term plain MusicVAEs [8] working in parallel, each encoding a 1-bar melody into a 64-dimensional latent code. All four models try to encode an 8-bar melody into a 512-dimensional latent code. The last two baseline methods do not produce context-sensitive representation; they only serve as general VAE baselines.

We evaluated how well these models can reconstruct the original melody with minimal information stored in z (i.e., minimal KL distance). The results are shown in Table 1. All models except the 8-bar LSTM can achieve high accuracy on reconstructing, while Transformer VAE with the attention mechanism has the lowest KL loss. It shows that the proposed method can effectively reduce the redundancy in the latent code z. We further compared the average KL loss of different bars between the models with attention and without attention (Fig. 4). With the help of attention, our model can store the information of bar $2..T$ in a more concise way.

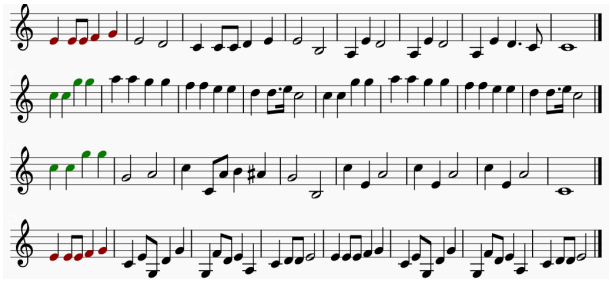

Fig. 5: An analogy example. From top to bottom: (a) original song $x^{(1)}$; (b) original song $x^{(2)}$; (c) generated song $\tilde{x}^{(1)}$; (d) generated song $\tilde{x}^{(2)}$.

3.4. Interactive Generation via Context Transfer

We further evaluated our system on the task of deep music analogy [11] to see how generated music piece will change according to the context[2]. As an example, we choose two 8-bar sample songs $x^{(1)}_{1..T}$ and $x^{(2)}_{1..T}$ as shown in Fig. 5 and encode them into the latent representation $z^{(1)}_{1..T}$ and $z^{(2)}_{1..T}$. Then we swap the representation of the first bar between these two songs and generate two new melody lines: $\tilde{x}^{(1)} = \text{Decode}(z^{(2)}_1, z^{(1)}_2, ...z^{(1)}_T), \tilde{x}^{(2)} = \text{Decode}(z^{(1)}_1, z^{(2)}_2, ...z^{(2)}_T)$. The results are shown in Fig. 5.

Although we only change the representation of the first bar, the generated songs change in every bar. Several interesting trends can be observed. (1) **Global structure remains**: the global structure and repetitions of the two pieces remain the same. $\tilde{x}^{(2)}$ keeps an A-A structure as in $x^{(2)}$, and $\tilde{x}^{(1)}$ keeps the repeated segments from the 5th to the 7th bar; (2) **Rhythmic pattern changes**: the generated piece $\tilde{x}^{(2)}$ begins to mimic the rhythmic pattern of the new first bar; (3) **Pitch range changes**: the pitch range of $x^{(2)}$ is higher than $x^{(1)}$. After swapping, some notes in $\tilde{x}^{(1)}$ rises to be consistent with the pitch range of the new first bar.

One limitation we observed is that the model focuses too much on exact repetitive patterns. It is still difficult to capture more complex relationships such as tonal sequence, retrograde and inversion. The problem might be solved by introducing additional inductive bias.

4. CONCLUSION

In this paper, we proposed the Transformer VAE model for phrase-level melody representation learning. The model combines the advantages of structure awareness from the Transformer model and interpretability from the deep music analogy framework. With the help of the controlled dependency via masked attention, the model can summarize the music using concise context-sensitive latent representation, and produce new analogous examples by context transfer.

[2]More demos are available at https://drive.google.com/open?id=1Su-8qrK__28mAesSCJdjo6QZf9zEgIx6

5. REFERENCES

[1] Ian Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[2] Mary Lou Maher, "Computational and collective creativity: Who's being creative?," in *ICCC*. Citeseer, 2012, pp. 67–71.

[3] 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*, 2014, pp. 2672–2680.

[4] Leon A Gatys, Alexander S Ecker, and Matthias Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[5] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[7] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck, "Music transformer," in *International Conference on Learning Representations*, 2019.

[8] 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.

[9] Ke Chen, Weilin Zhang, Shlomo Dubnov, Gus Xia, and Wei Li, "The effect of explicit structure encoding of deep neural networks for symbolic music generation," in *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*. IEEE, 2019, pp. 77–84.

[10] Nicolas E Gold and Roger B Dannenberg, "A reference architecture and score representation for popular music human-computer music performance systems," in *New Interfaces for Musical Expression*. Department of Musicology, University of Oslo/Norwegian Academy of Music, 2011.

[11] Ruihan Yang, Dingsu Wang, Ziyu Wang, Tianyao Chen, Junyan Jiang, and Gus Xia, "Deep music analogy via latent representation disentanglement," in *the 20th International Society for Music Information Retrieval Conference, ISMIR*, 2019.

[12] Jesse Engel, Matthew Hoffman, and Adam Roberts, "Latent constraints: Learning to generate conditionally from unconditional generative models," *arXiv preprint arXiv:1711.05772*, 2017.

[13] Christine Payne, "Musenet," `https://openai.com/blog/musenet/`, 2019.

[14] Tianming Wang and Xiaojun Wan, "T-cvae: Transformer-based conditioned variational autoencoder for story completion," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019, pp. 5233–5239.

[15] Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N Gomez, Stephan Gouws, Llion Jones, Lukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al., "Tensor2tensor for neural machine translation," *arXiv preprint arXiv:1803.07416*, 2018.

[16] Carl Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[17] 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.," *ICLR*, vol. 2, no. 5, pp. 6, 2017.

[18] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.