ATTRIBUTES-AWARE DEEP MUSIC TRANSFORMATION

Lisa Kawai[1] **Philippe Esling**[2] **Tatsuya Harada**[1,3]
[1] The University of Tokyo, Japan [2] IRCAM, France [3] RIKEN, Japan
{kawai, harada}@mi.t.u-tokyo.ac.jp, philippe.esling@ircam.fr

ABSTRACT

Recent machine learning techniques have enabled a large variety of novel music generation processes. However, most approaches do not provide any form of interpretable control over musical attributes, such as pitch and rhythm. Obtaining control over the generation process is critically important for its use in real-life creative setups. Nevertheless, this problem remains arduous, as there are no known functions nor differentiable approximations to transform symbolic music with control of musical attributes.

In this work, we propose a novel method that enables attributes-aware music transformation from any set of musical annotations, without requiring complicated derivative implementation. By relying on an adversarial confusion criterion on given musical annotations, we force the latent space of a generative model to abstract from these features. Then, reintroducing these features as conditioning to the generative function, we obtain a continuous control over them. To demonstrate our approach, we rely on sets of musical attributes computed by the jSymbolic library as annotations and conduct experiments that show that our method outperforms previous methods in control. Finally, comparing correlations between attributes and the transformed results show that our method can provide explicit control over any continuous or discrete annotation.

1. INTRODUCTION

For a long time, music composition has been considered a skill reserved for highly-trained experts. However, since the emergence of new technologies, it appears possible for non-expert and untrained users to indulge in this task. One such way is to rely on machine learning models, which train on existing sets of music to produce pieces with similar characteristics. The importance of music generation is growing in the industrial world as well, for instance, in the generation of soundtracks for video games and movies [1, 2].

In this context, models that provide control over the attributes of the generated music are of critical importance [1, 3–5]. Indeed, the simple generation of "self-contained" music through a single button rapidly becomes dull. Instead, users are rather looking to transform aspects of the music to make it fit their intent in terms of musical attributes such as pitch, rhythm, and melody. Ideally, these controls should be continuous, providing flexibility akin to the *knobs* of musical synthesizers. For instance, if a user wants to generate music with longer notes, the system should provide a way to decide *how much* the notes are extended, not just whether extend them or not.

Nevertheless, obtaining explicit controls over music generation is still an open and complex problem [5]. Although recent generative models can produce a large variety of music genres, they usually do not provide any mechanism on *how to modify* the generation with musical attributes in a controllable way. However, acquiring such controls seems to require the implementation of complicated derivatives or approximation for each attribute [6].

In this paper, we introduce a novel method to generate symbolic music similar to a given dataset, while being able to control its musical attributes continuously. Our method can be trained using raw annotations, without requiring access to their computational functions nor derivative implementations. This allows us to rely on any form of musical annotations, even high-level semantic or subjective characteristics. To demonstrate this aspect, we rely on attribute vectors computed by jSymbolic [7] as annotations to make the model learn a continuous control for each. This produces numerical descriptions of music, such as pitch ranges and mean note durations, which enables us to learn understandable control over the generation.

To empower our generative model with continuous control over non-differentiable musical attributes, we rely on adversarial learning. Our model is based on a latent encoder-decoder model, where the decoding function takes continuous musical attributes as conditioning information. As the model is trained to reconstruct input data, the latent encoding potentially contains all the information necessary for reconstruction. Hence, this would lead to a decoding function that can generate the output without reflecting changes in the conditioning. Aiming to prevent this situation, we introduce an adversarial discriminator on the latent space of the model. The role of this discriminator is to drive the encoder to remove any attribute information from the latent vector. This adversarial framework with an encoder-decoder model is similar to Fader Networks [8]. However, one major difference is that we handle continuous values instead of binary ones. Hence, a major contribution of this paper is to define an approach to train an adversarial discriminator on continuous values. Indeed, discriminators across the literature [8–15] predict

binary values such as real/fake or with/without attributes, whereas we aim to deal with continuous values. We solve this problem in two steps. First, we quantize attribute values and compute balanced class labels. Second, we extend the discriminator so that it relies on multivariate class vectors. As usual in adversarial training, the encoder is trained to prevent the discriminator from predicting the labels correctly. This produces continuous controls over the musical characteristics by shifting the condition to the decoder. This behavior stems from the need for the model to correctly utilize the decoder condition so as to reconstruct the input data. That way, we force the generated data to reflect the musical control values, for any attribute annotation.

For the purpose of evaluation, we demonstrate the ability to gain interpretable control over musical attributes by conducting extensive experiments. We show that our method outperforms previous proposals [6,16,17] and also allows us to rely on unconstrained sets of attributes. We evaluate these results by computing the correlation between control attributes and generation results. This enables us to evaluate quantitatively how the generation reflects the control. Our contributions are: (1) We propose a novel training framework to learn from ordered annotation values. (2) We solve the above problem by quantizing annotations and adversarial training to obtain controllability of musical attributes without their derivatives.

2. RELATED WORK

2.1 Generative Models

Generative models allow production of new data samples $\tilde{\mathbf{x}}$, by training on a collection of examples x, with the most popular approaches being Generative Adversarial Networks [9] and Variational Auto-Encoder (VAE) [18, 19].

2.1.1 Variational Auto-Encoder

The VAE is based on an encoder-decoder architecture. The encoder takes input data $\mathbf{x} \in \mathbb{R}^{d_\mathbf{x}}$ and outputs a compressed latent vector $\mathbf{z} \in \mathbb{R}^{d_\mathbf{z}}$. The decoder takes this latent vector $\mathbf{z}$ as input and tries to reconstruct the input data $\mathbf{x}$. Hence, this approach models two distributions: $\mathbf{z} \sim p_{enc}(\mathbf{z} \mid \mathbf{x}, \theta_{enc})$ and $\tilde{\mathbf{x}} \sim p_{dec}(\mathbf{x} \mid \mathbf{z}, \theta_{dec})$, where θ_{enc} and θ_{dec} are parameters of the encoder and the decoder. The original objective function for training a VAE approximates the distribution $p(\mathbf{x})$, that we wish to model by a lower bound (Evidence Lower Bound (ELBO)) [18, 19].

$$\mathrm{ELBO}_{\{\theta_{enc},\theta_{dec}\}} = -(\mathcal{L}_R + \mathcal{L}_{KL}) \leq \log p(\mathbf{x}) \quad (1)$$

Here, $\mathcal{L}_R$ and $\mathcal{L}_{KL}$ are mean reconstruction error and Kullback-Leibler divergence, respectively.

$$\mathcal{L}_R = -\mathbb{E}_{p_{enc}(\mathbf{z}|\mathbf{x})}\big[\log(p_{dec}(\mathbf{x} \mid \mathbf{z}))\big] \quad (2)$$

$$\mathcal{L}_{KL} = D_{KL}\big(p_{enc}(\mathbf{z} \mid \mathbf{x}) \,||\, p(\mathbf{z})\big) \quad (3)$$

Therefore, the model optimizes its parameters by minimizing the reconstruction error $\mathcal{L}_R$, while regularizing the latent space through $\mathcal{L}_{KL}$, so that the encoded latent variables match with a Gaussian prior.

2.1.2 Attribute Control in Generative Models

Several works have focused on attribute control for generative models in the image domain [8, 20–23], which enable changing properties of the generation, such as facial attributes. In some cases [21], learning is not performed on the attributes themselves, but rather through a rich pretrained model. This allows computing the mean directions of interpolation, which maximize attribute change in feature space, in order to re-apply the obtained direction.

In Fader Networks [8], an encoder-decoder-based model is trained to generate facial images given binary visual attributes through adversarial learning. As the decoder is expected to rely on control attributes for generation, this requires that the other latent vector input does not contain any information about the attributes. Thus, they introduce a discriminator on those latent vectors, which tries to classify the input latent vectors based on target binary attributes, such as wearing glasses or not. The encoder is simultaneously trained to prevent the discriminator from predicting target attributes correctly. Therefore, this adversarial criterion enables the model to learn an attributes-invariant representation by forcing the encoder to remove any attribute information from the latent vector. In a similar way, [20, 22] also proposed to gain attributes control by removing attribute information from a latent vector.

Given the success of these methods for facial image-specific components, our proposed model explores a similar approach. However, our model needs to train on continuous ordered values instead of binary attributes.

2.2 Controllable Music Generation

Several works have been focusing on music generation conditioned on composer styles [1, 24] or chord progression [10]. In contrast, our work focuses on the direct and continuous transformation of music.

2.2.1 Music Interpolation

In most music interpolation approaches, the models do not rely on annotations while training [16, 17]. Instead, they interpolate musical data by moving the latent vector in different directions. These directions are usually defined by reference points from the training set. In some cases, musical annotations are also used to learn the characteristics or certain styles of music, or even sequences of features [4, 25], which mean the models have annotation values in every time step. For instance, [4] relies on contour and fragmentation & consolidation sequences in addition to rhythmic ones, and they split the latent vector between a target attribute and the other unsupervised dimensions. Although these models are able to account for target attributes, their precise control remains unclear. Moreover, only a single attribute can be learned in each model, requiring a full model per attribute. In [26], the authors train separate models to explore the latent space based on gradient descent using binary annotations or user-defined functions.

Similarly, [6] explicitly maps the number of notes to a dimension of the latent vector with geometrical regularization. However, it requires access to a differentiable compu-

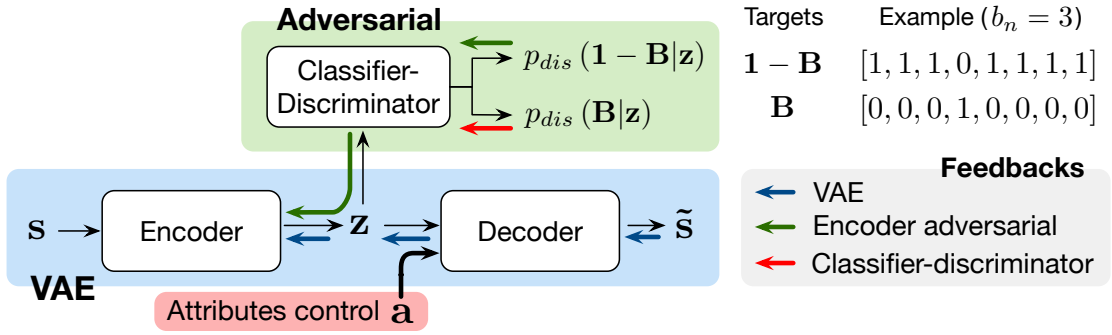

Figure 1. Overview of our proposed model. It is based on the Variational Auto-Encoder, and the decoder is conditioned on an attribute vector $\mathbf{a}$. To force the output to reflect the conditioning, our model has a classifier-discriminator in addition, which induces the encoder to remove the attribute information from the latent vector $\mathbf{z}$. The classifier-discriminator predicts the class labels $\mathbf{B}$, which are calculated by quantizing musical attributes $\mathbf{a}$, and trains the encoder in an adversarial way.

tational function of attributes. Thus, there is no guarantee that this model can deal with more complex attributes nor multiple attributes at the same time. Recent works [27, 28] also map the musical attributes to certain dimensions.

2.2.2 Music Style Transfer

A field related to our research is music style transfer. In this task, models aim to apply a *target style* while preserving the *content* of a given music piece. This task appears more complex than music interpolation as the precise definition of *style* is somewhat elusive. Depending on papers, style is either defined as genre, composer, or other undefined factors. The wide array of researches on music style transfer [15, 29–31] can be broadly separated between supervised and unsupervised approaches.

In the supervised realm, [31] relies on synthesized data of the same content in a variety of target styles, which can be considered as a ground-truth for transfer. Although this work appears to perform transfer successfully, its problem setting appears unrealistic, as we usually do not have access to paired ground truth data for every style.

Unsupervised music style transfer [15,29,30], provide a more realistic approach, by simply collecting random data with style labels and aiming to learn a model of the different musical styles. Therefore, these approaches avoid the pitfall of providing a clear definition of *content* or *style*. However, this leads to models which are extremely difficult to evaluate and does not provide controls on the precise characteristics of the generated material.

In our work, we aim to provide explicit control over a set of interpretable musical attributes, which can be defined as *musical style* collectively. Hence, we believe that our research can provide the first step towards a more grounded, interpretable music style transfer.

3. METHODOLOGY

In this paper, our goal is to devise a method for symbolic music generation, providing understandable controls over musical attributes of the generation. Furthermore, these controls should provide continuous modifications to the output, akin to the parameters of a modern synthesizer. This implies that we need to control the extent of different transformations, ensuring the quality of the generation.

3.1 Model Overview

To achieve our goal, we rely on a VAE architecture with an adversarial classifier-discriminator on the latent vector, as depicted in Figure 1. The encoder-decoder model is trained to reconstruct the input data, as in usual VAE frameworks. Moreover, as we aim to address the explicit control of musical attributes, the decoder takes these musical attributes as additional conditioning information to the latent vector. However, this latent vector potentially already contains all the information required to reconstruct the input data. This would lead to a decoder that simply does not use the information of conditioned attributes. In this degenerate situation, the model would not account for control modifications in the generation. The classifier-discriminator solves this problem, by driving the encoder to remove any information on the attribute from the latent space. Hence, the decoder is forced to use the attribute information to reconstruct the input data adequately, as this information is missing from the latent vector.

Although this architecture is similar to Fader Networks [8], our model is based on the VAE, and the attributes in our work are continuous ordered values, instead of binary indicators. This means that we cannot make direct use of existing discriminators [8–15]. To overcome this problem, we extend the discriminator mechanism so that it acts as a classifier as well. To do so, we quantize the attributes into K balanced classes and use these as targets to a multivariate discriminator. As a result of adversarial training, the encoder prevents the discriminator from predicting the class labels correctly. Hence, the resulting latent vectors should not contain any musical attributes information.

3.2 Model Architecture

We consider a monophonic pitch sequence $\mathbf{s}_{1:T}$ as input, where T represents time steps, encoded in a piano-roll representation, as a sequence of its one-hot vectors (Eq. 4).

Encoder: The encoder is simply defined as a one-layer bidirectional Gated Recurrent Unit (GRU), followed by

linear layers (denoted as MLP) to generate the mean and variance of the latent vector $\mathbf{z}$.

$$\mathbf{E}_t = \text{onehot}(s_t) \quad (4)$$

$$(\mathbf{O}, \mathbf{H}) = \text{GRU}(\mathbf{E}_{1:T}) \quad (5)$$

$$p_{enc}(\mathbf{z} \mid \mathbf{s}) = \mathcal{N}\Big(\mathbf{z} \Big| \text{MLP}(\mathbf{O}_T), \text{diag}\big(\exp(\text{MLP}(\mathbf{O}_T))\big)\Big) \quad (6)$$

where $\mathbf{O}$ is the output feature, and $\mathbf{H}$ the hidden state.

Decoder: The decoder reconstructs the input $\mathbf{E}_t$ at time t, based on the latent vector $\mathbf{z}$, the one-hot vector of the previous step $\mathbf{E}_{t-1}$, and the vector of musical attributes $\mathbf{a}$. The difference between our decoder and a conditional VAE is that this musical attributes vector is used along with the latent information, based on the premise that the latent vector does not have information about these attributes. This decoder is composed of a two-layer GRU.

$$p_{dec}(\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid 0, \mathcal{I}) \quad (7)$$

$$(\mathbf{O}_1, \mathbf{H}_1) = \text{GRU}\big([\mathbf{E}_0; \mathbf{z}; \mathbf{a}], \text{MLP}(\mathbf{z})\big) \quad (8)$$

$$(\mathbf{O}_t, \mathbf{H}_t) = \text{GRU}\big([\mathbf{E}_{t-1}; \mathbf{z}; \mathbf{a}], \mathbf{H}_{t-1}\big) \quad (9)$$

$$p_{dec}(s_t \mid \mathbf{s}_{1:t-1}, \mathbf{z}, \mathbf{a}) = \text{Cat}\big(s_t \mid \sigma(\text{MLP}(\mathbf{H}_t))\big) \quad (10)$$

where $[;]$ is a concatenation of vectors, Cat the categorical distribution, and σ the softmax function.

3.3 Attribute Control

In this section, we detail our method to provide understandable control over musical attributes in the music generation process.

3.3.1 Musical Attributes

We rely on the jSymbolic [7] software to calculate statistical musical attributes from symbolic music data. It provides 246 kinds of features, such as pitch statistics, melodies, intervals, rhythm, instrumentation, texture, and dynamics. Out of these, we picked twelve features based on their interpretability such as the total number of notes, pitch variability, and rhythmic value variability [1]. Each feature is normalized to have zero mean and unit variance. We obtain a set $\mathbf{a} = (a_1, \ldots, a_N)$ of N attributes, where each attribute is averaged across a given input example.

3.3.2 Quantize Attributes

To train our discriminator, we first quantize the attributes into class labels $b_n \in \{1, 2, 3, \ldots, K\}$, where $K = 8$ in this work. The quantization operation $b_n = \text{quantize}(a_n)$, which is a μ-law compression, leading to an equal number of training data in each class, as depicted in Figure 2.

3.3.3 Classifier-Discriminator

The classifier-discriminator takes a latent vector $\mathbf{z}$ and predicts the class which the data belongs to. Hence, the target of the classifier-discriminator can be described as follows

$$gt(n, k) = \begin{cases} 1 & (k = b_n) \\ 0 & (\text{else}) \end{cases} \quad (11)$$

[1] The detailed set of features along with their meaning is available in associated webpage of this paper (`https://lisakawai.github.io/music_transformation/`).

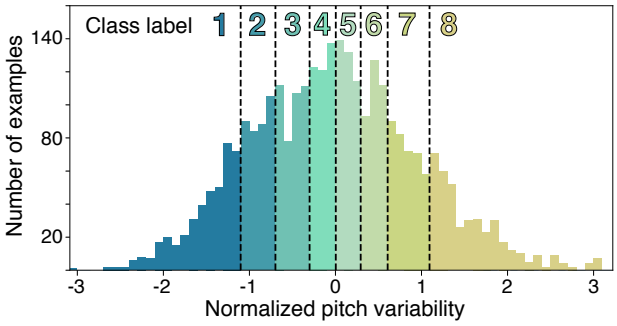

Figure 2. Data distribution of pitch variability, showing how the data is split.

where $k \in [1, K]$ is the index of the quantized class and $n \in [1, N]$ is the attribute index. In order to define the adversarial criterion, the target of the encoder is $1 - gt(n, k)$.

The classifier-discriminator is defined as linear layers followed by *tanh* activation functions for all layers, except the last layer, which uses a *sigmoid* function. Thus, this network outputs a matrix $\mathbf{B} \in \mathbb{R}^{N \times K}$. Hence, this is a major difference of our classifier-discriminator, which relies on a multivariate output, instead of a scalar one [8].

3.4 Loss Function

In this work, three separate loss functions are used to optimize the model, namely, the reconstruction loss $\mathcal{L}_R$, the Kullback-Leibler divergence $\mathcal{L}_{KL}$, and the adversarial loss $\mathcal{L}_D$. The first two functions are used to optimize the VAE, as detailed in Section 2.1.1. $\mathcal{L}_D$ is used to train the classifier-discriminator and influences the encoded latent vectors, as detailed in the following section.

3.4.1 Adversarial Loss

To compute $\mathcal{L}_D$, we rely on the vectors of attribute class labels $\mathbf{b} = (b_1, \ldots, b_N)$, as detailed in Section 3.3.2. The classifier-discriminator aims to predict the class labels, by optimizing its probability distribution p_{dis}. On the other hand, the encoder aims to prevent the classifier-discriminator from predicting the class labels correctly. Hence, the targets of the encoder for the classifier-discriminator prediction is the complement vector of the class labels instead of a scalar in [8], defined as $\mathbf{1} - \mathbf{B} \in \mathbb{R}^{N \times K}$, where all the elements of $\mathbf{1}$ are equal to 1.

$$\mathbf{B} = \text{onehot}(\mathbf{b}) \quad (12)$$

$$\mathcal{L}_D(\theta_{dis} | \theta_{enc}) = -\,\mathbb{E}_{p_{enc}(\mathbf{z}|\mathbf{s})}[\log(p_{dis}(\mathbf{B}|\mathbf{z}))] \quad (13)$$

$$\mathcal{L}_D(\theta_{enc} | \theta_{dis}) = -\,\mathbb{E}_{p_{enc}(\mathbf{z}|\mathbf{s})}[\log(p_{dis}(\mathbf{1} - \mathbf{B}|\mathbf{z}))] \quad (14)$$

where θ_{dis} is parameters of the discriminator. Note that each element of the one-hot vectors $B_{n,k}$ is equal to $gt(n, k)$ in Eq (11).

3.4.2 Total Loss Function

The complete loss function of our method is defined as:

$$\mathcal{L}(\theta_{enc}, \theta_{dec} \mid \theta_{dis}) = \mathcal{L}_R + \alpha \mathcal{L}_{KL} + \beta \mathcal{L}_D(\theta_{enc} \mid \theta_{dis}) \quad (15)$$

$$\mathcal{L}(\theta_{dis} \mid \theta_{enc}) = \mathcal{L}_D(\theta_{dis} \mid \theta_{enc}) \quad (16)$$

where α and β are hyperparameters for controlling the impact of each loss in the optimization of the model.

4. EXPERIMENTS

4.1 Dataset

In our experiments, we rely on Nottingham dataset [32], a collection of monophonic British and American folk tunes, including both melodies and chords information. The examples are divided between 694 train, 170 test, and 173 validation instances. We filtered the dataset to retain only examples with 4/4 signature and used only the melody information. We split the data to obtain every sequence of four bars and performed pitch augmentation by shifting pitches from -5 to 6 for the training set. In the final dataset, the pitch ranges from 50 (D3) to 95 (B6).

4.2 Input Representation

In this work, we use a piano-roll-like input representation of monophonic music, which is a sequence of one-hot vectors, representing fixed-bar melodies as a matrix of dimension $time \times (pitch+2)$. The difference from a piano-roll is that we add two dimensions to represent *continue* (holding the previous note) and *rest* (no pitch is on) information. We choose this representation as it allows distinguishing short repeated notes more precisely. We compute the final matrix from the input $\mathbf{s} = (s_1, s_2, \ldots, s_T)$, where $T = 64$.

4.3 Baselines

We compare our proposal with two baselines: *naive VAE* and *GLSR-VAE* [6]. The implementation of the naive VAE is the same as ours without the attribute conditioning and the classifier-discriminator, and we calculate mean latent vectors with/without an attribute to decide an interpolation direction for a given latent vector following [16,17]. To apply binary attributes information, in this case, we split the data into two classes so that the instance is considered *with* an attribute if $a_n \geq 0$ and *without* the attribute if $a_n < 0$. We define the mean latent vector of the attribute presence as $\mathbf{z_w}$ and absence as $\mathbf{z_{wo}}$ and perform the interpolation by moving a latent vector as $\mathbf{z} + \delta \times (\mathbf{z_w} - \mathbf{z_{wo}})$, where δ ranges from -0.5 to 0.5 by steps of 0.1, which is within the interquartile range of $\mathcal{N}(0, \mathcal{I})$.

In the implementation of the GLSR-VAE, we simply add its regularization term to the naive VAE model. In this model, one dimension of the latent vector z_0 corresponds to the *number of notes* attribute. The interpolation is performed by computing $z_0 + \delta$, where δ is as defined previously. Note that this model is only able to interpolate the number of notes with existing implementation, while ours is applicable to any attributes.

4.4 Implementation Details

We train all of the models by using a batch size of 64, the learning rate is set to 1e-4, and we use the ADAM optimizer [33] for 50K iterations. The GRU layer has a hidden size of 1024 for both the encoder and the decoder, and the latent vector has 128 dimensions. In the VAE loss function (see Eq (15)), we use $\alpha = 0.1$ and $\beta = 0.1$.

attribute	Naive	GLSR	Ours
total number of notes	0.973	0.975	**0.981**
pitch variability	0.807	-	**0.938**
rhythmic value variability	0.830	-	**0.938**
pitch kurtosis	0.528	-	**0.723**
pitch skewness	0.366	-	**0.492**
most common rhythm val.	0.795	-	**0.851**
average note duration	0.968	-	**0.983**
note density variability	0.677	-	**0.855**
amount of arpeggiation	0.126	-	**0.386**
chromatic motion	0.284	-	**0.622**
direction of melodic motion	0.428	-	**0.702**
melodic arcs interval span	0.262	-	**0.523**
total number of notes	0.949	0.279	**0.950**
pitch variability	0.797	-	**0.918**
rhythmic value variability	0.809	-	**0.849**

Table 1. Correlation coefficient comparison of musical attributes. **Top**: Results of transformed outputs by changing δ. **Bottom**: Results of *cycle* transformation reverting original attributes as condition to already transformed outputs.

4.5 Evaluation Metrics

To evaluate our model performance, we need to assess both the reconstruction accuracy and the efficiency of the attributes control. Hence, the outputs generated by the model should adequately reflect the changes in the attributes. To evaluate this aspect, we rely on Spearman's rank-order correlation coefficient between conditioning attribute a_{in} and the resulting attribute a_{out} calculated from the output. The conditioning attribute a_{in} is calculated from the original attribute a_{org} depending on the change made in the control, so that $a_{in} = a_{org} + \delta$, where δ is the same as Section 4.3.

5. RESULTS

5.1 Results with Single Attribute

We compute the correlation coefficient between the target attributes and the corresponding jSymbolic features calculated on the interpolated outputs. All the interpolation results can be found in Table 1 (Top). Although some attributes such as the *total number of notes* and the *pitch variability* are easily handled by a naive VAE, some others such as *chromatic motion* and *amount of arpeggiation* are largely more difficult to detect. Our model allows us to obtain control over all the features, while outperforming others on the interpolated results.

Cycle consistency check. We also perform an experiment to check the *consistency* of attribute control. Ideally, based on a transformed output, if we revert the attribute to its original value, and use it as a condition to regenerate the interpolated results, the model should be expected to reproduce the original score. According to Table 1 (Bottom), our model is able to maintain high correlation coefficients even after successive interpolations. Oppositely, GLSR-VAE is unable to maintain this correlation, as the model cannot control the degree of attribute change.

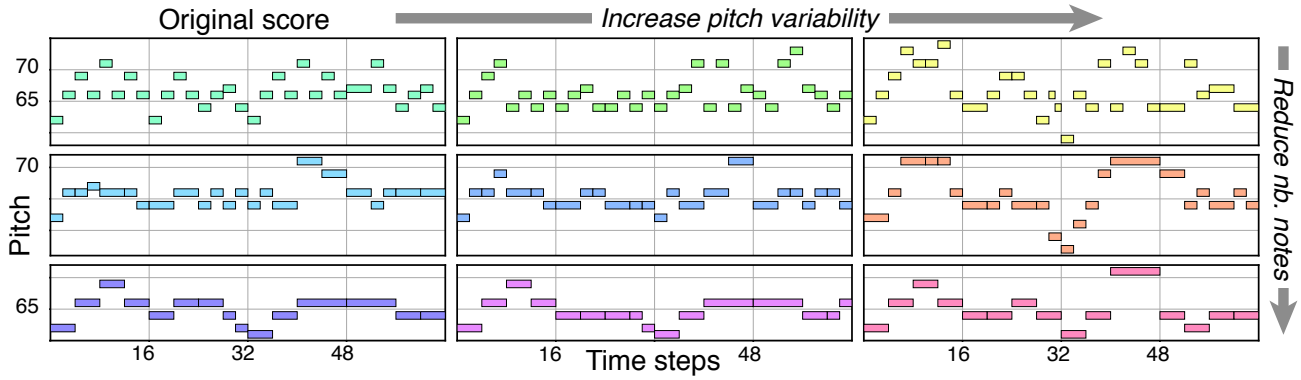

Figure 3. Generated samples by changing the *total number of notes* and *pitch variability* attributes. Top left shows the original score and the others are outputs from our model, which is able to transform multiple musical attributes simultaneously.

	Naive	GLSR	Ours
NLL	2.269	**1.002**	1.679
accuracy	0.759	**0.808**	0.790

Table 2. Negative log-likelihood (NLL) and accuracy comparison when models are trained on an attribute, the *total number of notes.*

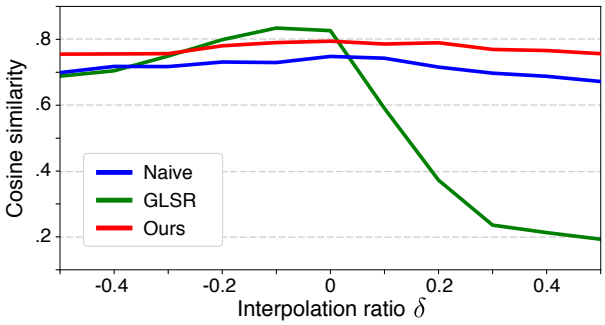

Figure 4. Cosine similarity of chroma features changing the *total number of notes* to compare chord consistency.

Reconstruction quality. We compute the negative log-likelihood (NLL) and reconstruction quality (accuracy) for unchanged attribute values and display results in Table 2. It seems that the GLSR-VAE approach provides slightly better results in the pure reconstruction scenario. However, our proposed model does not deteriorate the reconstruction quality as seen with the naive VAE.

Chord consistency check. Previous works in the field of music style transfer [30, 31] rely on chord consistency analysis to evaluate the capacity of the model to transfer the style in a given music piece while keeping its content. In [14], the authors use chroma feature as one of the criteria for the evaluation of music similarity. We follow this evaluation metrics to observe how different models can preserve the chord structures. This feature possesses 12 dimensions, each of them representing the intensity of a given pitch across octaves in one bar. To evaluate consistency, we compute the cosine similarity between chroma features of the original data and the interpolated one. The results for the *total number of notes* attribute are displayed in Figure 4 for varying values of δ. As a result, our method performs better than baselines for preserving chord consistency. Although GLSR-VAE provides better results for $\delta \approx 0$, its cosine similarity significantly drops for a positive δ, where its chord consistency quality rapidly degrades as $|\delta|$ becomes larger.

N	attribute	corr
2	total number of notes	0.983
	pitch variability	0.944
3	total number of notes	0.978
	pitch variability	0.936
	rhythmic value variability	0.922

Table 3. Correlation coefficient with multiple attributes used in experiments for our model.

5.2 Results with Multiple Attributes

We display in Figure 3 the samples generated by our model trained on multiple attributes simultaneously (here the *total number of notes* and *pitch variability*). It shows that we can obtain interesting transformations on a given original score, while using interpretable controls. This emphasizes the ability of our model to control multiple attributes.

To further analyze this behavior, we compute the correlation coefficient of multiple attributes by shifting only one attribute at the same time, while inputting the original values for the others, as displayed in Table 3. These results indicate that the correlation coefficient remains stable, even when successively adding new controls. This shows that our model is able to produce *independent* control of multiple musical attributes, which is of prime importance for precise and intuitive music creation.

6. CONCLUSION

In our work, we proposed a new model for deep music transformation. We relied on musical attributes and introduced a model able to learn how to generate music based on these attributes. This was done by quantizing the attributes and introducing an adversarial classifier-discriminator on latent features. The experimental results showed that our model leads to independent and robust controls of musical attributes for monophonic music.

7. ACKNOWLEDGEMENTS

This work was partially supported by JST AIP Acceleration Research Grant Number JPMJCR20U3, and partially supported by JSPS KAKENHI Grant Number JP19H01115. This work was also supported by the ANR:17-CE38-0015-01 MAKIMOno project, the SSHRC:895-2018-1023 ACTOR Partnership and Emergence(s) ACIDITEAM project from Ville de Paris and ACIMO project of Sorbonne Université.

8. REFERENCES

[1] H. H. Mao, T. Shin, and G. W. Cottrell, "Deepj: Style-specific music generation," in *Proc. of International Conference on Semantic Computing, ICSC*, 2018, pp. 377–382.

[2] D. Williams, A. Kirke, J. Eaton, E. Miranda, I. Daly, J. Hallowell, E. Roesch, F. Hwang, and S. J. Nasuto, "Dynamic game soundtrack generation in response to a continuously varying emotional trajectory," in *Audio Engineering Society Conference: Audio for Games*, 2015.

[3] F. Pachet, A. Papadopoulos, and P. Roy, "Sampling variations of sequences for structured music generation," in *Proc. of International Society for Music Information Retrieval Conference, ISMIR*, 2017, pp. 167–173.

[4] T. Akama, "Controlling symbolic music generation based on concept learning from domain knowledge," in *Proc. of International Society for Music Information Retrieval Conference, ISMIR*, 2019, pp. 816–823.

[5] L. Ferreira and J. Whitehead, "Learning to generate music with sentiment," in *Proc. of International Society for Music Information Retrieval Conference, ISMIR*, 2019, pp. 384–390.

[6] G. Hadjeres, F. Nielsen, and F. Pachet, "Glsr-vae: Geodesic latent space regularization for variational autoencoder architectures," in *IEEE Symp. Series on Computational Intelligence, SSCI*, 2017, pp. 1–7.

[7] C. McKay and I. Fujinaga, "jsymbolic: A feature extractor for midi files," in *Proc. of the International Computer Music Conference, ICMC*, 2006, pp. 302–305.

[8] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. A. Ranzato, "Fader networks:manipulating images by sliding attributes," in *Advances in Neural Information Processing Systems, NeurIPS*, 2017, pp. 5967–5976.

[9] 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, NeurIPS*, 2014, pp. 2672–2680.

[10] L. Yang, S. Chou, and Y. Yang, "Midinet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proc. of International Society for Music Information Retrieval Conference, ISMIR*, 2017, pp. 324–331.

[11] H.-M. Liu and Y.-H. Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," in *Proc. of Machine Learning and Applications, ICMLA*, 2018, pp. 722–727.

[12] H. Dong, W. Hsiao, L. Yang, and Y. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. of Conference on Artificial Intelligence, AAAI*, 2018, pp. 34–41.

[13] H. Dong and Y. Yang, "Convolutional generative adversarial networks with binary neurons for polyphonic music generation," in *Proc. of International Society for Music Information Retrieval Conference, ISMIR*, 2018, pp. 190–196.

[14] M. Bretan and L. P. Heck, "Self-supervised methods for learning semantic similarity in music," in *Proc. of International Society for Music Information Retrieval Conference, ISMIR*, 2019, pp. 446–453.

[15] G. Brunner, Y. Wang, R. Wattenhofer, and S. Zhao, "Symbolic music genre transfer with cyclegan," in *Proc. of Tools with Artificial Intelligence, ICTAI*, 2018, pp. 786–793.

[16] A. Roberts, J. H. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *Proc. of International Conference on Machine Learning, ICML*, 2018, pp. 4361–4370.

[17] I. Simon, A. Roberts, C. Raffel, J. Engel, C. Hawthorne, and D. Eck, "Learning a latent space of multitrack measures," *CoRR*, vol. abs/1806.00195, 2018.

[18] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. of International Conference on Learning Representations, ICLR*, 2014.

[19] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. of International Conference on Machine Learning, ICML*, 2014, pp. 1278–1286.

[20] N. Hadad, L. Wolf, and M. Shahar, "A two-step disentanglement method," in *Proc. of Computer Vision and Pattern Recognition, CVPR*, 2018, pp. 772–780.

[21] P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, and K. Weinberger, "Deep feature interpolation for image content changes," in *Proc. of Computer Vision and Pattern Recognition, CVPR*, 2017, pp. 6090–6099.

[22] M. Li, W. Zuo, and D. Zhang, "Deep identity-aware transfer of facial attributes," *CoRR*, vol. abs/1610.05586, 2016.

[23] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras, "Neural face editing with intrinsic image disentangling," in *Proc. of Computer Vision and Pattern Recognition, CVPR*, 2017, pp. 5444–5453.

[24] C. M. Payne, *MuseNet*, 2019, https://openai.com/blog/musenet/.

[25] R. Yang, D. Wang, Z. Wang, T. Chen, J. Jiang, and G. Xia, "Deep music analogy via latent representation disentanglement," in *Proc. of International Society for Music Information Retrieval Conference, ISMIR*, 2019, pp. 596–603.

[26] J. H. Engel, M. Hoffman, and A. Roberts, "Latent constraints: Learning to generate conditionally from unconditional generative models," in *Proc. of International Conference on Learning Representations, ICLR*, 2018.

[27] A. Pati and A. Lerch, "Latent space regularization for explicit control of musical attributes," in *ICML Workshop on Machine Learning for Music Discovery Workshop (ML4MD)*, 2019.

[28] A. Pati and A. Lerch, "Attribute-based regularization of vae latent spaces," *CoRR*, vol. abs/2004.05485, 2020.

[29] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer," in *Proc. of International Society for Music Information Retrieval Conference, ISMIR*, 2018, pp. 747–754.

[30] W. T. Lu and L. Su, "Transferring the style of homophonic music using recurrent neural networks and autoregressive model," in *Proc. of International Society for Music Information Retrieval Conference, ISMIR*, 2018, pp. 740–746.

[31] O. Cífka, U. Simsekli, and G. Richard, "Supervised symbolic music style translation using synthetic data," in *Proc. of International Society for Music Information Retrieval Conference, ISMIR*, 2019, pp. 588–595.

[32] E. Foxley, *Nottingham Database*, 2011, http://abc.sourceforge.net/NMD/.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of International Conference on Learning Representations, ICLR*, 2015.