MUSIC FADERNETS: CONTROLLABLE MUSIC GENERATION BASED ON HIGH-LEVEL FEATURES VIA LOW-LEVEL FEATURE MODELLING

Hao Hao Tan[1] **Dorien Herremans**[1]
[1] Singapore University of Technology and Design
{haohao_tan, dorien_herremans}@sutd.edu.sg

ABSTRACT

High-level musical qualities (such as emotion) are often abstract, subjective, and hard to quantify. Given these difficulties, it is not easy to learn good feature representations with supervised learning techniques, either because of the insufficiency of labels, or the subjectiveness (and hence large variance) in human-annotated labels. In this paper, we present a framework that can learn high-level feature representations with a limited amount of data, by first modelling their corresponding quantifiable *low-level* attributes. We refer to our proposed framework as Music FaderNets, which is inspired by the fact that low-level attributes can be continuously manipulated by separate "sliding faders" through feature disentanglement and latent regularization techniques. High-level features are then inferred from the low-level representations through semi-supervised clustering using Gaussian Mixture Variational Autoencoders (GM-VAEs). Using arousal as an example of a high-level feature, we show that the "faders" of our model are disentangled and change linearly w.r.t. the modelled low-level attributes of the generated output music. Furthermore, we demonstrate that the model successfully learns the intrinsic relationship between arousal and its corresponding low-level attributes (rhythm and note density), with only 1% of the training set being labelled. Finally, using the learnt high-level feature representations, we explore the application of our framework in style transfer tasks across different arousal states. The effectiveness of this approach is verified through a subjective listening test.

1. INTRODUCTION

We consider *low-level* musical attributes as attributes that are relatively straightforward to quantify, extract and calculate from music, such as rhythm, pitch, harmony, etc. On the other hand, *high-level* musical attributes refer to semantic descriptors or qualities of music that are relatively abstract, such as emotion, style, genre, etc. Due to the nature of abstractness and subjectivity in these high-level musical qualities, obtaining labels for these qualities typically requires human annotation. However, training conditional models on top of these human-annotated labels using supervised learning might result in sub-par performance because firstly, obtaining such labels can be costly, hence the amount of labels collected might be insufficient to train a model that can generalize well [1]; Secondly, the annotated labels could have high variance among raters due to the subjectivity of these musical qualities [2, 3].

Instead of inferring high-level features directly from the music sample, we propose to use low-level features as a "bridge" between the music and the high level features. This is because the relationship between the sample and its low-level features can be learnt relatively easier, as the labels are easier to obtain. In addition, we learn the relationship between the low-level features and the high-level features in a data-driven manner. In this paper, we show that the latter works well even with a limited amount of labelled data. Our work relies heavily on the concept that each high-level feature is intrinsically related to a set of low-level attributes. By tweaking the levels of each low-level attribute in a constrained manner, we can achieve a desired change on the high-level feature. This idea is heavily exploited in rule-based systems [4–6], however rule-based systems are often not robust enough as their capabilities are constrained by the fixed set of predefined rules handcrafted by the authors. Hence, we propose an alternative path which is to *learn* these implicit relationships with semi-supervised learning techniques.

To achieve the goals stated above, we intend to build a framework which can fulfill these two objectives:

- Firstly, the model should be able to control multiple low-level attributes of the music sample in a continuous manner, as if it is controlled by sliding knobs on a console (or also known as *faders*). Each knob should be independent from the others, and only controls one single feature that it is assigned to.
- Secondly, the model should be able to learn the relationship between the levels of the sliding knobs controlling the low-level features, and the selected high-level feature. This is analogous to learning a *preset* of the sliding knobs on a console.

We named our model "Music FaderNets", with reference to musical "faders" and "presets" as described above. Achieving the first objective requires representation learning and feature disentanglement techniques. This motivates us to use *latent variable models* [7] as we can learn

Attribution: Hao Hao Tan, Dorien Herremans, "Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling", in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.,* Montréal, Canada, 2020.

separate latent spaces for each low-level feature to obtain disentangled controllability. Achieving the second objective requires the latent space to have a hierarchical structure, such that high-level information can be inferred from low-level representations. This is achieved by incorporating Gaussian Mixture VAEs [8] in our model.

2. RELATED WORK

2.1 Controllable Music Generation

The application of deep learning techniques for music generation has been rapidly advancing [9–13], however, embedding *control* and *interactivity* in these systems still remains a critical challenge [10]. Variants of conditional generative models (such as CGAN [14] and CVAE [15]) are used to allow control during generation, which have attained much success mainly in the image domain. Fader Networks [16] is one of the main inspirations of this work (hence the name Music FaderNets), in which users can modify different visual features of an image using "sliding faders". However, their approach is built upon a CVAE with an additional adversarial component, which is very different from our approach. Recently, controllable music generation has gained much research interest, both on modelling low-level [17–20] and high-level features [21, 22]. Specifically, [18] and [19] each proposed a novel latent regularization method to encode attributes along specific latent dimensions, which inspired the "sliding knob" application in this work.

2.2 Disentangled Representation Learning for Music

Disentangled representation learning has been widely used across both the visual [23–26] and speech domain [1, 27, 28] to learn disjoint subsets of attributes. Such techniques have also been applied to music in several recent works, both in the audio [29–31] and symbolic domain [32–34]. The discriminator component in our model draws inspiration from both the explicit conditioning component in the EC^2-VAE model [33], and the *extraction* component in the Ext-Res model [34]. We find that most of the work on disentanglement in symbolic music focuses on low-level features, and is done on monophonic music.

This research distinguishes itself from other related work through the following novel contributions:

- We combine latent regularization techniques with disentangled representation learning to build a framework that can control various continuous low-level musical attribute values using "faders", and apply the framework on *polyphonic* music modelling.
- We show that it is possible to infer high-level features from low-level latent feature representations, even under weakly supervised scenario. This opens up possibilities to learn good representations for abstract, high-level musical qualities even under data scarcity conditions. We further demonstrate that the learnt representations can be used for controllable generation based on high-level features.

3. PROPOSED FRAMEWORK

3.1 Gaussian Mixture Variational Autoencoders

VAEs [35] combine the power of both latent variable models and deep generative models, hence they provide both representation learning and generation capabilities. Given observations $\mathbf{X}$ and latent variables $\mathbf{z}$, the VAE learns a graphical model $\mathbf{z} \rightarrow \mathbf{X}$ by maximizing the evidence lower bound (ELBO) of the marginal likelihood $p(\mathbf{X})$ as below:

$$\mathcal{L}(p, q; \mathbf{X}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{X})}[\log p(\mathbf{X}|\mathbf{z})] - \mathcal{D}_{KL}(q(\mathbf{z}|\mathbf{X})||p(\mathbf{z}))$$

where $q(\mathbf{z}|\mathbf{X})$ and $p(\mathbf{z})$ represent the learnt posterior and prior distribution respectively. In vanilla VAEs, $p(\mathbf{z})$ is an isotropic, unimodal Gaussian. Gaussian Mixture VAEs (GM-VAE) [8] extend the prior to a mixture of K Gaussian components, which corresponds to learning a graphical model with an extra hierarchy of dependency $c \rightarrow \mathbf{z} \rightarrow \mathbf{X}$. The newly introduced categorical variable $c \in \mathcal{C}$, whereby $|\mathcal{C}| = K$, is a discrete representation of the observations. Hence, a new distribution $q(c|\mathbf{X})$ is introduced to infer the class of each observation, which enables semi-supervised and unsupervised clustering applications.

Following [8], the ELBO of a GM-VAE is derived as:

$$\begin{aligned}\mathcal{L}(p, q; \mathbf{X}) = {} & \mathbb{E}_{q(\mathbf{z}|\mathbf{X})}[\log p(\mathbf{X}|\mathbf{z})] \\ & - \sum_{k=1}^{K} q(c_k|\mathbf{X}) \mathcal{D}_{KL}(q(\mathbf{z}|\mathbf{X})||p(\mathbf{z}|c_k)) \\ & - \mathcal{D}_{KL}(q(c|\mathbf{X})||p(c))\end{aligned}$$

The original KL loss term from the vanilla VAE is modified into two new terms: (i) the KL divergence between the approximate posterior $q(\mathbf{z}|\mathbf{X})$ and the conditional prior $p(\mathbf{z}|c_k)$, marginalized over all Gaussian components; (ii) the KL divergence between the cluster inferring distribution $q(c|\mathbf{X})$, and the categorical prior $p(c)$.

3.2 Model Formulation

Figure 1 shows the model formulation of our proposed Music FaderNets. Input $\mathbf{X}$ is a sequence of performance tokens converted from MIDI following [12, 13]. Assume that we want to model a high-level feature with K discrete states, which is related to a set of N low-level features. We denote the latent variables learnt for each low-level feature as $\mathbf{z}_{1...N}$; the labels for each low-level feature as $\mathbf{y}_{1...N}$; and the class inferred from each latent variable as $c_{1...N}$.

The joint probability of $\mathbf{X}, \mathbf{z}_{1...N}, c_{1...N}$ is written as:

$$p(\mathbf{X}, \mathbf{z}_{1...N}, c_{1...N}) = p(\mathbf{X}|\mathbf{z}_{1...N}) \prod_{i=1}^{N} p(\mathbf{z}_i|c_i) \prod_{i=1}^{N} p(c_i)$$

We assume that each categorical prior $p(c_i)$, $i \in [1, N]$ is uniformly distributed, and the conditional distributions $p(\mathbf{z}_i|c_i) = \mathcal{N}(\mu_{c_i}, \text{diag}(\sigma_{c_i}))$ are diagonal-covariance Gaussians with learnable means and constant variances. For each low-level attribute, we learn an approximate posterior $q(\mathbf{z}_i|\mathbf{X})$, parameterized by an encoder neural network, that samples latent code $\mathbf{z}_i$ which represents the i-th low-level feature.

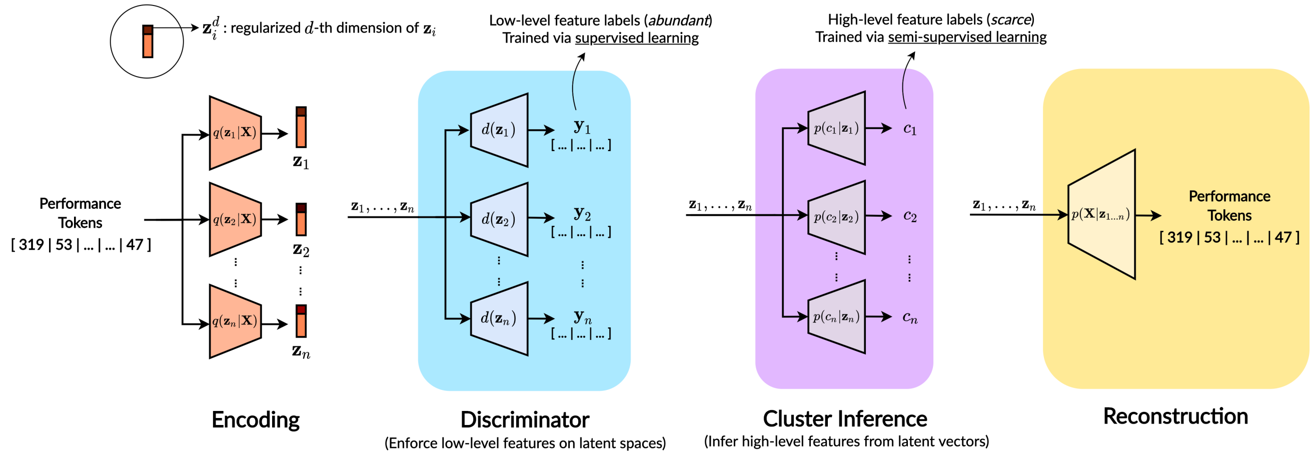
$\mathbf{z}_i^d$: regularized d-th dimension of $\mathbf{z}_i$
Low-level feature labels (*abundant*)
Trained via supervised learning
High-level feature labels (*scarce*)
Trained via semi-supervised learning
Performance Tokens
[319 | 53 | ... | ... | 47]
$q(\mathbf{z}_1|\mathbf{X})$
$q(\mathbf{z}_2|\mathbf{X})$
$q(\mathbf{z}_n|\mathbf{X})$
$\mathbf{z}_1$
$\mathbf{z}_2$
$\mathbf{z}_n$
$\mathbf{z}_1, \ldots, \mathbf{z}_n$
$d(\mathbf{z}_1)$
$d(\mathbf{z}_2)$
$d(\mathbf{z}_n)$
$\mathbf{y}_1$ [... | ... | ...]
$\mathbf{y}_2$ [... | ... | ...]
$\mathbf{y}_n$ [... | ... | ...]
$\mathbf{z}_1, \ldots, \mathbf{z}_n$
$p(c_1|\mathbf{z}_1)$
$p(c_2|\mathbf{z}_2)$
$p(c_n|\mathbf{z}_n)$
c_1
c_2
c_n
$\mathbf{z}_1, \ldots, \mathbf{z}_n$
$p(\mathbf{X}|\mathbf{z}_{1\ldots n})$
Performance Tokens
[319 | 53 | ... | ... | 47]
Encoding
Discriminator
(Enforce low-level features on latent spaces)
Cluster Inference
(Infer high-level features from latent vectors)
Reconstruction

Figure 1. Music FaderNets model architecture.

The latent codes $\mathbf{z}_{1\ldots N}$ are then passed through the remaining three components: (1) **Discriminator**: To ensure that $\mathbf{z}_i$ incorporates information of the assigned low-level feature, it is passed through a discriminator represented by a function $d(\mathbf{z}_i)$ to reconstruct the low-level feature label $\mathbf{y}_i$; (2) **Reconstruction**: All latent codes are fed into a global decoder network which parameterizes the conditional probability $p(\mathbf{X}|\mathbf{z}_{1\ldots n})$ to reconstruct the input $\mathbf{X}$; (3) **Cluster Inference**: This component parameterizes the cluster inference probability $q(c|\mathbf{X})$, with c representing the selected high-level feature. It can be approximated by $q(c|\mathbf{X}) \approx \mathbb{E}_{q(\mathbf{z}|\mathbf{X})} p(c|\mathbf{z})$ [36], where the cluster state is predicted from each latent code $\mathbf{z}_i$ instead of $\mathbf{X}$.

To incorporate the "sliding knob" concept, we need to map the change of value of an arbitrary dimension on $\mathbf{z}_i$ (denoted as $\mathbf{z}_i^d$, shown on Figure 1 as the darkened dimension) linearly to the change of value of the low-level feature label $\mathbf{y}_i$. After comparing across previous methods on conditioning and regularization [15, 16, 18, 19], we choose to adopt [19] which applies a latent regularization loss term written as $\mathcal{L}_{\text{reg}}(\mathbf{z}_i^d, \mathbf{y}_i) = \text{MSE}(\tanh(\mathcal{D}_{\mathbf{z}_i^d}), \text{sign}(\mathcal{D}_{\mathbf{y}_i}))$, where $\mathcal{D}_{\mathbf{z}_i^d}$ and $\mathcal{D}_{\mathbf{y}_i}$ denotes the *distance matrix* of values $\mathbf{z}_i^d$ and $\mathbf{y}_i$ within a training batch respectively. We provide a detailed comparison study across each proposed method in Section 4.2. Hence, if we define:

$$\mathcal{L}^i_\phi(p, q; \mathbf{X}) = \begin{cases} \sum_{k=1}^{K} q(c_{i,k}|\mathbf{X}) \mathcal{D}_{KL}(q(\mathbf{z}_i|\mathbf{X}) || p(\mathbf{z}_i|c_{i,k})) \\ + \mathcal{D}_{KL}(q(c_i|\mathbf{X}) || p(c_i)), \text{if unsupervised} \\ \\ \mathcal{D}_{KL}(q(\mathbf{z}_i|\mathbf{X}) || p(\mathbf{z}_i|c_i)), \text{if supervised} \end{cases} \tag{1}$$

then the entire training objective can be derived as:

$$\begin{aligned} \mathcal{L}(p, q; \mathbf{X}) = {} & \mathbb{E}_{q(\mathbf{z}_1|\mathbf{X})\ldots q(\mathbf{z}_N|\mathbf{X})}[\log p(\mathbf{X}|\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N)] \\ & - \beta \cdot \sum_{i=1}^{N} \mathcal{L}^i_\phi(p, q; \mathbf{X}) + \sum_{i=1}^{N} \mathcal{L}_{\text{reg}}(\mathbf{z}_i^d, \mathbf{y}_i) \\ & + \mathbb{E}_{q(\mathbf{z}_1|\mathbf{X})\ldots q(\mathbf{z}_n|\mathbf{X})}[\log p(\mathbf{y}_1|\mathbf{z}_1)\ldots p(\mathbf{y}_N|\mathbf{z}_N)] \end{aligned} \tag{2}$$

where β is the KL weight hyperparameter [24]. The first term in Eq. 2 represents the reconstruction loss. The second KL loss term (derived from the ELBO function of GM-VAE) correspond to the cluster inference component, which allows both *supervised* and *unsupervised* training setting, depending on the availability of label c. If we omit the cluster inference component, it could conform to a vanilla VAE by replacing this term with the KL loss term of VAE. The third term is the latent regularization loss applied during the encoding process. The last term is the reconstruction loss of the low-level feature labels, which corresponds to the discriminator component. All encoders and decoders are implemented with gated recurrent units (GRUs), and teacher-forcing is used to train all decoders.

4. EXPERIMENTAL SETUP

In this work, we chose *arousal* (which refers to the energy level conveyed by the song [37]) as the high-level feature to be modelled. In order to select relevant low-level features, we refer to musicology papers such as [6, 38, 39], which suggest that arousal is related to features including rhythm density, note density, key, dynamic, tempo, etc. Among these low-level features, we focus on modelling the score-level features in this work (i.e. rhythm, note and key).

4.1 Data Representation and Hyperparameters

We use two polyphonic piano music datasets for training: the **Yamaha Piano-e-Competition dataset** [12], and the **VGMIDI dataset** [3], which contains piano arrangements of 95 video game soundtracks in MIDI, annotated with valence and arousal values in the range of -1 to 1. The arousal labels are used to guide the cluster inference component in our GM-VAE model using semi-supervised learning. We extract every 4-beat segment from each music sample, with a beat resolution of 4 (quarter-note granularity). Each segment is encoded into event-based tokens following [12] with a maximum sequence length of 100. This results in a total of 103,934 and 1,013 sequences from the Piano e-Competition and VGMIDI dataset respectively, which are split into train/validation/test sets with a ratio of 80/10/10.

Inspired by [33], we represent each rhythm label, $\mathbf{y}_{\text{rhythm}}$, as a sequence of 16 one-hot vectors with 3 dimensions, denoting an onset for any pitch, a holding state, or a rest. The *rhythm density* value is calculated as the number of onsets in the sequence divided by the total sequence length. Each note label, $\mathbf{y}_{\text{note}}$, is represented by a sequence of 16 one-hot vectors with 16 dimensions, each dimension denoting the number of notes being played or held at that time step (we assume a minimum polyphony of 0 and a maximum of 15). The *note density* value is the average number of notes being played or held for per time step. For key, we use the key analysis tool from music21 [40] to extract the estimated global key of each 4-beat segment. The key is represented using a 24-dimension one-hot vector, accounting for major and minor modes. In this work, we directly concatenate the key vector as a conditioning signal with $\mathbf{z}_{\text{rhythm}}$ and $\mathbf{z}_{\text{note}}$ as an input to the global decoder for reconstruction. For representing arousal, we split the arousal ratings into two clusters ($K = 2$): *high* arousal cluster for positive labels, and *low* arousal cluster for negative labels. We remove labels annotated within the range [-0.1, 0.1] so as to reduce ambiguity in the annotations.

The hyperparameters are tuned according to the results on the validation set using grid search. The mean vectors of $p(c|\mathbf{z})$ are all randomly initialized with Xavier initialization, whereas the variance vectors are kept fixed with value e^{-2}. We observe that the following annealing strategy for β leads to the best balance between reconstruction and controllability: β is set to 0 in the first 1,000 training steps, and is slowly annealed up to 0.2 in the next 10,000 training steps. We set the batch size to 128, all hidden sizes to 512, and the encoded $\mathbf{z}$ dimensions to 128. The Adam optimizer is used with a learning rate of 10^{-3}.

4.2 Measuring the Controllability of Latent Features

The proposed Music FaderNets model should meet two requirements: (i) Each "fader" independently controls one low-level musical feature without affecting other features (disentanglement), and (ii) the "faders" should change linearly with the controlled attribute of the generated output (linearity). For disentanglement, we follow the definition proposed in [41] which decomposes the concept of disentanglement into *generator consistency* and *generator restrictiveness*. Using rhythm density as an example:

- *Consistency* on rhythm density means that for the same value of $\mathbf{z}^d_{\text{rhythm}}$, the value of the output's rhythm density should be consistent.
- *Restrictiveness* on rhythm density means that changing the value of $\mathbf{z}^d_{\text{rhythm}}$ does not affect the attributes other than rhythm density (in our case, note density).
- *Linearity* on rhythm density means that the value of rhythm density is directly proportional to the value of $\mathbf{z}^d_{\text{rhythm}}$, which is analogous to a sliding fader.

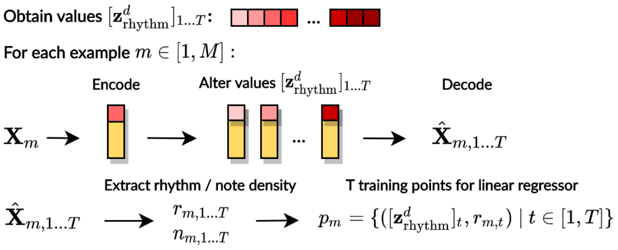
Obtain values $[\mathbf{z}^d_{\text{rhythm}}]_{1...T}$:
...
For each example $m \in [1, M]$:
Encode
Alter values $[\mathbf{z}^d_{\text{rhythm}}]_{1...T}$
Decode
$\mathbf{X}_m$
...
$\hat{\mathbf{X}}_{m,1...T}$
Extract rhythm / note density
T training points for linear regressor
$\hat{\mathbf{X}}_{m,1...T}$
$r_{m,1...T}$
$n_{m,1...T}$
$p_m = \{([\mathbf{z}^d_{\text{rhythm}}]_t, r_{m,t}) \mid t \in [1, T]\}$

Figure 2. Workflow of obtaining evaluation metrics for "faders" controlling rhythm density.

We will be evaluating all three of these points in our experiment. For evaluating linearity, [19] proposed a slightly modified version of the interpretability metric by [42], which includes the following steps: (1) encode each sample in the test set, obtain the rhythm latent code and the dimension $\mathbf{z}^d$ which has the maximum mutual information with regards to the attribute; (2) learn a linear regressor to predict the input attribute values based on $\mathbf{z}^d$. The linearity score is hence the coefficient of determination (R^2) score of the linear regressor. However, this method evaluates only the encoder and not the decoder. As we want the sliding knobs to directly impact the output, we argue that the relationship between $\mathbf{z}^d$ and the output attributes should be more important. Hence, we propose to "slide" the values of the regularized dimension $\mathbf{z}^d$ within a given range and decode them into reconstructed outputs. Then, instead of predicting the *input* attributes given the encoded $\mathbf{z}^d$, the linear regressor learns to predict the corresponding *output* attributes given the "slid" values of $\mathbf{z}^d$.

We demonstrate a single workflow to calculate the consistency, restrictiveness and linearity scores of a given model based on the low-level features (we use rhythm density as an example low-level feature for the discussion below), as depicted in Figure 2. After obtaining the rhythm density latent code for all samples in the training set and finding the minimum and maximum value of $\mathbf{z}^d_{\text{rhythm}}$, we "slide" for $T = 8$ steps by calculating $\min(\mathbf{z}^d_{\text{rhythm}}) + \frac{t}{T}(\max(\mathbf{z}^d_{\text{rhythm}}) - \min(\mathbf{z}^d_{\text{rhythm}}))$, with $t \in [1, T]$. This results in a list of values denoted as $[\mathbf{z}^d_{\text{rhythm}}]_{1...T}$. Then, we conduct the following steps:

1. Randomly select $M = 100$ samples from the test set, and encode each sample into $\mathbf{z}_{\text{rhythm}}$ and $\mathbf{z}_{\text{note}}$;
2. Alter the d-th element in $\mathbf{z}_{\text{rhythm}}$ using the values in the range $[\mathbf{z}^d_{\text{rhythm}}]_{1...T}$, to obtain $[\hat{\mathbf{z}}_{\text{rhythm}}]_{m,1...T}$ for each sample m;
3. Decode each new rhythm density latent code together with the unchanged note density latent code $\mathbf{z}_{\text{note}}$ to get $\hat{\mathbf{X}}_{m,1...T}$;
4. Calculate rhythm density $r_{m,1...T}$ and note density $n_{m,1...T}$ for each reconstructed output;
5. Pair up the new rhythm density latent code with the resulting rhythm density of the output as T training data points $p_m = \{([\mathbf{z}^d_{\text{rhythm}}]_t, r_{m,t}) \mid t \in [1, T]\}$ for a linear regressor.

The final evaluation scores are then calculated as follows:

$$\text{Consistency score} = 1 - \frac{1}{T}\sum_{t=1}^{T} \underset{t}{\sigma}(r_{1...M,t}) \tag{3}$$

	Consistency		Restrictiveness		Linearity	
	Rhythm Density	Note Density	Rhythm Density	Note Density	Rhythm Density	Note Density
Proposed (Vanilla VAE)	0.4367 ± 0.0258	0.3490 ± 0.0360	0.6645 ± 0.0169	0.6481 ± 0.0154	**0.7805 ± 0.0142**	**0.8255 ± 0.0107**
Proposed (GM-VAE)	**0.5096 ± 0.0248**	0.4207 ± 0.0309	0.6603 ± 0.0164	0.6457 ± 0.0132	0.7580 ± 0.0124	0.7792 ± 0.0177
Pati et al. [19]	0.4625 ± 0.0264	**0.5100 ± 0.0150**	0.6417 ± 0.0171	0.5497 ± 0.0206	0.7613 ± 0.0171	0.8220 ± 0.0143
CVAE [15]	0.2613 ± 0.0376	0.4997 ± 0.0355	**0.6863 ± 0.0221**	0.7140 ± 0.0130	0.4969 ± 0.0166	0.3997 ± 0.0411
Fader Networks [16]	0.2730 ± 0.0366	0.4983 ± 0.0425	0.6861 ± 0.0163	**0.7379 ± 0.0149**	0.5482 ± 0.0283	0.4647 ± 0.0292
GLSR [18]	0.1891 ± 0.0346	0.1969 ± 0.0831	0.6365 ± 0.0276	0.7136 ± 0.0185	0.2465 ± 0.0197	0.1799 ± 0.0209

Table 1. Experimental results (conducted on the Yamaha dataset test split) on the controllability of low-level features (rhythm density and note density) using disentangled latent variables. Bold marks the best performing model.

$$\text{Restrictiveness score} = 1 - \frac{1}{M}\sum_{m=1}^{M} \sigma_{m}(n_{m,1..T}) \quad (4)$$

$$\text{Linearity score} = R^2(\mathcal{M}(p_{1...M})) \quad (5)$$

where $\sigma(\cdot)$ denotes the standard deviation, and $\mathcal{M}$ denotes the linear regressor model. In other words, consistency calculates the average standard deviation across all output rhythm density values given the same $\mathbf{z}^d_{\text{rhythm}}$, whereas restrictiveness calculates the average standard deviation across all output note density values given the changing $\mathbf{z}^d_{\text{rhythm}}$. In a perfectly disentangled and linear model, the consistency, restrictiveness and linearity scores should be equal to 1, and higher scores indicate better performance.

5. EXPERIMENTS AND RESULTS

We compare the evaluation scores of our proposed model, using both a vanilla VAE (omitting the cluster inference component) and GM-VAE, with several models proposed in related work on controllable synthesis: CVAE [15], Fader Networks [16], GLSR [18] and Pati et al. [19]. We repeat the above steps for 10 runs for each model and report the mean and standard deviation of each score. Table 1 shows the evaluation results. Overall, our proposed models achieve a good all-rounded performance on every metric as compared to other models, especially in terms of linearity, models that use [19]'s regularization method largely outperform other models. Our model shares similar results with [19], however as compared to their work, we encode a multi-dimensional, regularized latent space instead of a single dimension value for each low-level feature, thus allowing more flexibility. Our model can also be used for "generation via analogy" as mentioned in EC^2-VAE [33], by mix-matching $\mathbf{z}_{\text{rhythm}}$ from one sample with $\mathbf{z}_{\text{note}}$ from another. Moreover, the feature latent vectors can be used to infer interpretable and semantically meaningful clusters.

5.1 Inferring High-Level Features from Latent Low-Level Representations

Figure 3 visualizes the rhythm and note density latent space learnt by GM-VAE using t-SNE dimensionality reduction. We observe that both spaces successfully learn a Gaussian-mixture space with two well-separated components, which correspond to high and low arousal clusters, even though it was trained with only around 1% of labelled data. We also find that the regularized $\mathbf{z}^d$ values capture the overall trend of the actual rhythm and note density values. Interestingly, the model learns the implicit relationship between high/low arousal and the corresponding levels of rhythm/note density. From Figure 3, we observe that the high arousal cluster corresponds to higher rhythm density and lower note density, whereas the low arousal cluster corresponds to lower rhythm density and higher note density. This is reasonable as music segments with high arousal often consist of fast running notes and arpeggios, being played one note at a time, whereas music segments with low arousal often exhibit a chordal texture with more sustaining notes and relatively less melodic activity.

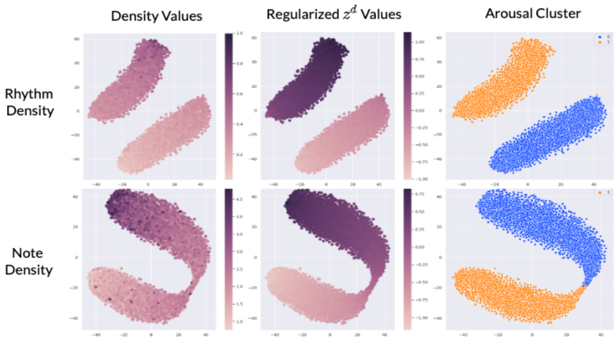

Figure 3. Visualization of rhythm (top) and note (bottom) density latent space in the GM-VAE. Each column is colored in terms of: (left) original density values, (middle) regularized $\mathbf{z}^d$ values, (right) arousal cluster labels (0 refers to low arousal and 1 refers to high arousal).

To further inspect the importance of using low-level features, we train a separate GM-VAE model with only one encoder (without discriminator component), which encodes only a single latent vector for each segment. The model is trained to infer the arousal label with the single latent vector similarly in a semi-supervised manner, and the hyperparameters are kept the same. From Figure 4, we can observe that the latent space learnt without using low-level features is not well-segregated into two separate components, suggesting that the right choice of low-level features helps the learning of a more discriminative and disentangled feature latent space.

The major advantage demonstrated from the results above is that by carefully choosing low-level features supported by domain knowledge, semi-supervised (or weakly supervised) training can be leveraged to learn interpretable representations that can capture implicit relationships between high-level and low-level features, overcoming the

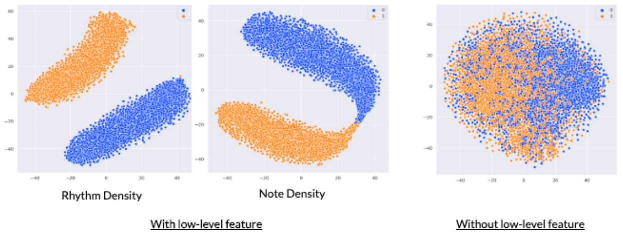

Figure 4. Arousal cluster visualization of GM-VAE with (left), and without (right) using low-level features.

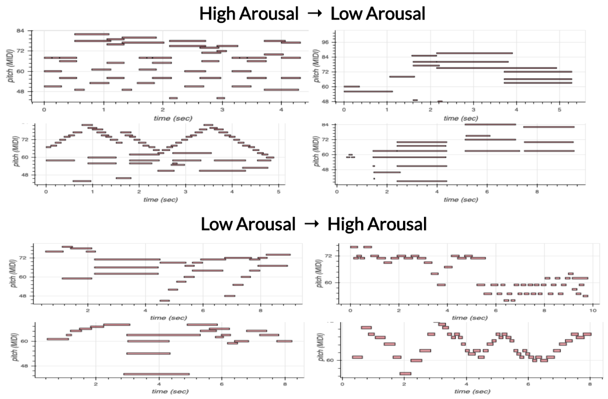

Figure 5. Examples of arousal transfer on music samples.

difficulties mentioned in the introduction section. This is an important insight for learning representations of abstract musical qualities under label scarcity conditions in future.

5.2 Style Transfer on High Level Features

Utilizing the learnt high-level feature representations enables the application of feature style transfer. Following [29], given the means of each Gaussian component, $\mu_{\text{arousal=0}}$ and $\mu_{\text{arousal=1}}$, the “shifting vector” from high arousal to low arousal is $s_{\text{low\_shift}} = \mu_{\text{arousal=0}} - \mu_{\text{arousal=1}}$, and vice versa. To shift a music segment from high to low arousal, we modify the latent codes by $\mathbf{z}'_{\text{rhythm}} = \mathbf{z}_{\text{rhythm}} + s_{\text{low\_shift}}$, $\mathbf{z}'_{\text{note}} = \mathbf{z}_{\text{note}} + s_{\text{low\_shift}}$. Both new latent codes $\mathbf{z}'_{\text{rhythm}}$ and $\mathbf{z}'_{\text{note}}$ are fed into the global decoder for reconstruction. For cases where $c_{\text{rhythm}} \neq c_{\text{note}}$, we choose to perform shifting only on the latent codes which are not lying within the target arousal cluster. Figure 5 shows several examples of arousal shift performed on given music segments. We can observe that the shift is clearly accompanied with the desired changes in rhythm density and note density, as mentioned in Section 5.1. More examples are available online. [1] We also conducted a subjective listening test to evaluate the quality of arousal shift performed by Music FaderNets. We randomly chose 20 music segments from our dataset, and performed a low-to-high arousal shift on 10 segments and a high-to-low arousal shift on the other 10. Each subject listened to the original sample and then the transformed sample, and was asked whether (1) the arousal level changes after the transformation, and; (2) how well the transformed sample sounds in terms of rhythm, melody, harmony and naturalness, on a Likert scale of 1 to 5 each.

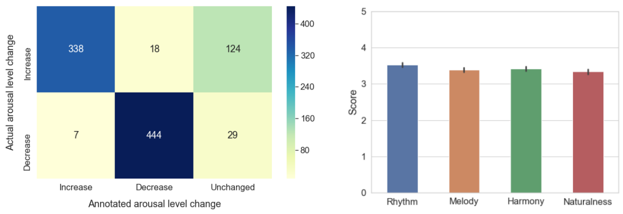

Figure 6. Subjective listening test results. Left: Heat map of annotated arousal level change against actual arousal level change. Right: Bar plot of opinion scores for each musical quality, with 95% confidence interval.

A total of 48 subjects participated in the survey. We found that 81.45% of the responses agreed with the actual direction of level change in arousal, shifted by the model. This showed that our model is capable of shifting the arousal level of a piece to a desired state. From the heat map shown in Figure 6, we observe that shifting from high to low arousal has a higher rate of agreement (92.5%) than shifting from low to high arousal (70.41%). Meanwhile, the mean opinion score of rhythm, melody, harmony and naturalness were reported at 3.53, 3.39, 3.41 and 3.33 respectively, showing that the quality of the generated samples are generally above moderate level.

6. CONCLUSION AND FUTURE WORK

We propose a novel framework called Music FaderNets [2] , which can generate new variations of music samples by controlling levels (“sliding knobs”) of low-level attributes, trained with latent regularization and feature disentanglement techniques. We also show that the framework is capable of inferring high-level feature representations (“presets”, e.g. arousal) on top of latent low-level representations by utilizing the GM-VAE framework. Finally, we demonstrate the application of using learnt high-level feature representations to perform arousal transfer, which was confirmed in a user experiment. The key advantage of this framework is that it can learn interpretable mixture components that reveal the intrinsic relationship between low-level and high-level features using semi-supervised learning, so that abstract musical qualities can be quantified in a more concrete manner with limited amount of labels.

While the strength of arousal transfer is gradually increased, we find that the identity of the original piece is also gradually shifted. A recent work on text generation using VAEs [43] observed this similar trait and attributed its cause to the “latent vacancy" problem by topological analysis. A possible solution is to adopt the Constrained-Posterior VAE [43], in which we aim to explore in future work. Future work will also focus on applying the framework on other sets of abstract musical qualities (such as valence [37], tension [44], etc.), and extending the framework to model multi-track music with longer duration to produce more complete music.

[1] `https://music-fadernets.github.io/`

[2] Source code available at: `https://github.com/gudgud96/music-fader-nets`

7. ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their constructive reviews. We also thank Yin-Jyun Luo for the insightful discussions on GM-VAEs. This work is supported by MOE Tier 2 grant no. MOE2018-T2-2-161 and SRG ISTD 2017 129. The subjective listening test is approved by the Institutional Review Board under SUTD-IRB 20-315. We would also like to thank the volunteers for taking the subjective listening test.

8. REFERENCES

[1] R. Habib, S. Mariooryad, M. Shannon, E. Battenberg, R. Skerry-Ryan, D. Stanton, D. Kao, and T. Bagby, "Semi-supervised generative modeling for controllable speech synthesis," in *International Conference of Learning Representations*, 2020.

[2] A. Aljanaki, Y.-H. Yang, and M. Soleymani, "Emotion in music task at mediaeval 2015." in *MediaEval*, 2015.

[3] L. N. Ferreira and J. Whitehead, "Learning to generate music with sentiment," in *Proc. of the International Society for Music Information Retrieval Conference*, 2019.

[4] R. Bresin and A. Friberg, "Emotional coloring of computer-controlled music performances," *Computer Music Journal*, vol. 24, no. 4, pp. 44–63, 2000.

[5] S. R. Livingstone, R. Muhlberger, A. R. Brown, and W. F. Thompson, "Changing musical emotion: A computational rule system for modifying score and performance," *Computer Music Journal*, vol. 34, no. 1, pp. 41–64, 2010.

[6] S. K. Ehrlich, K. R. Agres, C. Guan, and G. Cheng, "A closed-loop, music-based brain-computer interface for emotion mediation," *PloS one*, vol. 14, no. 3, 2019.

[7] Y. Kim, S. Wiseman, and A. M. Rush, "A tutorial on deep latent variable models of natural language," *arXiv preprint arXiv:1812.06834*, 2018.

[8] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," *arXiv preprint arXiv:1611.05148*, 2016.

[9] D. Herremans and C.-H. Chuan, "The emergence of deep learning: new opportunities for music and audio technologies," *Neural Computing and Applications*, vol. 32], p. 913–914, 2020.

[10] J.-P. Briot, G. Hadjeres, and F. Pachet, *Deep learning techniques for music generation*. Springer, 2019, vol. 10.

[11] D. Herremans, C.-H. Chuan, and E. Chew, "A functional taxonomy of music generation systems," *ACM Computing Surveys (CSUR)*, vol. 50, no. 5, pp. 1–30, 2017.

[12] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Computing and Applications*, pp. 1–13, 2018.

[13] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long term structure," in *International Conference of Learning Representations*, 2019.

[14] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[15] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, 2015, pp. 3483–3491.

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

[17] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *International Conference of Machine Learning*, 2018.

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

[19] A. Pati and A. Lerch, "Latent space regularization for explicit control of musical attributes," in *ICML Machine Learning for Music Discovery Workshop (ML4MD), Extended Abstract, Long Beach, CA, USA*, 2019.

[20] J. Engel, M. Hoffman, and A. Roberts, "Latent constraints: Learning to generate conditionally from unconditional generative models," in *International Conference of Learning Representations*, 2017.

[21] S. Dai, Z. Zhang, and G. G. Xia, "Music style transfer: A position paper," in *Proc. of International Workshop on Musical Metacreation*, 2018.

[22] K. Choi, C. Hawthorne, I. Simon, M. Dinculescu, and J. Engel, "Encoding musical style with transformer autoencoders," in *International Conference of Machine Learning*, 2020.

[23] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2016, pp. 2172–2180.

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

[25] H. Kim and A. Mnih, "Disentangling by factorising," in *International Conference of Machine Learning*, 2018.

[26] L. Yingzhen and S. Mandt, "Disentangled sequential autoencoder," in *International Conference on Machine Learning*, 2018, pp. 5670–5679.

[27] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *Advances in neural information processing systems*, 2017, pp. 1878–1889.

[28] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference of Machine Learning*, 2018.

[29] Y.-J. Luo, K. Agres, and D. Herremans, "Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders," in *Proc. of the International Society for Music Information Retrieval Conference*, 2019.

[30] Y.-N. Hung, Y.-A. Chen, and Y.-H. Yang, "Learning disentangled representations for timber and pitch in music audio," *arXiv preprint arXiv:1811.03271*, 2018.

[31] Y.-N. Hung, I. Chiang, Y.-A. Chen, Y.-H. Yang *et al.*, "Musical composition style transfer via disentangled timbre representations," in *International Joint Conference of Artificial Intelligence*, 2019.

[32] 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 the International Society for Music Information Retrieval Conference*, 2018.

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

[34] T. Akama, "Controlling symbolic music generation based on concept learning from domain knowledge," in *Proc. of the International Society for Music Information Retrieval Conference*, 2019.

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

[36] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen *et al.*, "Hierarchical generative modeling for controllable speech synthesis," in *International Conference of Learning Representations*, 2019.

[37] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[38] A. Gabrielsson and E. Lindström, "The influence of musical structure on emotional expression." 2001.

[39] P. Gomez and B. Danuser, "Relationships between musical structure and psychophysiological measures of emotion." *Emotion*, vol. 7, no. 2, p. 377, 2007.

[40] M. S. Cuthbert and C. Ariza, "music21: A toolkit for computer-aided musicology and symbolic music data," in *Proc. of the International Society for Music Information Retrieval Conference*, 2010.

[41] R. Shu, Y. Chen, A. Kumar, S. Ermon, and B. Poole, "Weakly supervised disentanglement with guarantees," in *International Conference of Learning Representations*, 2020.

[42] T. Adel, Z. Ghahramani, and A. Weller, "Discovering interpretable representations for both deep generative and discriminative models," in *International Conference on Machine Learning*, 2018, pp. 50–59.

[43] P. Xu, J. C. K. Cheung, and Y. Cao, "On variational learning of controllable representations for text without supervision," in *International Conference on Machine Learning*, 2020.

[44] D. Herremans and E. Chew, "Morpheus: generating structured music with constrained patterns and tension," *IEEE Transactions on Affective Computing*, 2017.