CONTROLLING SYMBOLIC MUSIC GENERATION BASED ON CONCEPT LEARNING FROM DOMAIN KNOWLEDGE

Taketo Akama
Sony Computer Science Laboratories, Tokyo, Japan
taketo.akama@sony.com

ABSTRACT

Machine learning allows automatic construction of generative models for music. However, they are learned from only the succession of notes itself without explicitly employing domain knowledge of musical concepts such as rhythm, contour, and fragmentation & consolidation. We approximate such musical domain knowledge as a function, and feed it into our model. Then, two decoupled spaces are learned: the *extraction space* that captures the target concept, and the *residual space* that captures the remainder. For monophonic symbolic music, our model exhibits high decoupling/modeling performance. Controllability in generation is improved: (i) our *interpolation* enables concept-aware flexible control over blending two musical fragments, and (ii) our *variation generation* enables users to make concept-aware adjustable variations.

1. INTRODUCTION

Listeners not only perceive the succession of notes itself, but also respond to higher-level concepts in music. Two critical components in melodic perception and memory are scale and contour [5]. It is said that similarity between musical fragments is important in listeners' emotional arousal responses to music [18]. Listeners sense those similarities through perceiving patterns of music constructs or transformations such as rhythm, interval, and fragmentation & consolidation (F&C) [20].

Music data processing, especially music generation and analysis have attracted much attention. One of the major methods exploits models that learn the *latent space* [1, 6, 9, 16, 27–29]. These models learn compressed but informative feature vectors of data samples and distribute them in the multi-dimensional latent space. The spacial arrangement represents the relation of data samples. Also, numerous intermediate features—corresponding to samples hopefully not in the dataset—are yielded to fill in the "holes" in the latent space. Then generation and analysis are performed by bidirectional mapping of the latent space and the data space. The latent space is, however, learned from raw musical data without supervision. Therefore, the musical notions or concepts that are important for people are not sufficiently organized on the latent space. Hereinafter, we refer to such notions or concepts as *musical concepts*, examples of which include rhythm, contour, and F&C, as mentioned above.

How do we organize those musical concepts on the latent space? People possess domain knowledge about musical concepts, although even the major concepts are not necessarily defined clearly. In fact, various musical concepts can be approximated as a *function* of raw musical sequences. We input such domain knowledge to our model in the form of a function, and then our model learns latent spaces that capture the corresponding musical concepts.

Our model is called ExtRes (Extraction-Residual Latent Space Decoupling Model), and it aims at learning decoupled latent spaces, each of which is associated with a musical concept. In other words, each musical concept is learned as a *latent-space concept* that occupies a part of the dimensions in the multi-dimensional latent space. The concept-wise decoupled latent spaces allow us to measure similarity between musical fragments in terms of each concept. The similarity is then used for e.g., pattern discovery in a musical piece [17]. In generation, the control over concept-wise latent features helps us to create musical phrases as imagined or to compose patterns/structures in a musical piece [23].

Our ExtRes has the following characteristics. (I) **Knowledge based**: musical concepts can be incorporated as function approximations on the basis of domain knowledge. For monophonic musical sequences, various important musical concepts can be incorporated such as rhythm, chromatic/diatonic interval or pitch, step-leap/signed contour, and F&C. This is possible because given an input sequence, these concepts can be approximated as other sequences with rules or algorithms [2, 19] to construct the functions. (II) **Complex concept and active learning**: complex concepts are actively learned without obtaining attributes first [3, 4, 12, 14], where a set of attributes is a concept. (III) **Extraction-residual**: aiming at comprehending complex data by repeatedly analyzing "one concept versus the rest." The rest (residual) may capture the useful concepts (e.g., scale for contour and "pitch order" for rhythm). (IV) **Coexisting latent spaces**: depending on the input domain knowledge, corresponding concepts are captured in latent spaces. Users can exploit multiple models of ours—wherein latent spaces with different concepts *coexist*—for handling many concepts.

We apply our ExtRes to improving controllability in interactive music generation. (i) **Concept-axes interpolation**: this mechanism helps users to create musical phrases as imagined. Users first input two musical fragments into the system. Prior methods allow users to adjust the *blending* ratio of the two fragments uniformly regardless of concepts [27, 28], whose blending is more musically meaningful than the naive data space blending. Our concept-axes interpolation offers more flexibility: enabling users to blend only the factor of the desired concept in musical fragments and to also adjust the blending ratio for each target/residual concept. (ii) **Concept-aware variation generation**: given a musical fragment, this mechanism allows users to obtain variations, where the amount of variation for each concept is adjustable. When generating a long structured piece of music, this mechanism helps to faithfully realize either of the instructions of a song template [23] or the user intention.

2. METHODOLOGY

2.1 Outline

We propose ExtRes, a generative model that allows learning reusable representation (as latent features and embedding vectors) for a user-specified concept, given a function based on domain knowledge on the concept (Sec.2.2). We then instantiate our proposed ExtRes model for sequence datasets (Sec.2.4). In Sec.2.5, musical concepts are approximated as functions on the basis of domain knowledge. For applications of our ExtRes, we focus on meaning-level controllability in generation. We consider the following kinds of control: (i) altering in relation to other samples (e.g., interpolation), (ii) altering a sample to another sample with similar but not the same meaning (e.g., variation), and (iii) altering individual concepts. ExtRes not only puts (iii) into practice by free explorations in concept spaces, but also allows hybridizing the meaning-level controls (i-iii) for more intended controllability: Sec.2.3 Concept-Axes Interpolation is for bridging (i) and (iii), Sec.2.3 Concept-Aware Variation Generation is for bridging (ii) and (iii).

2.2 Extraction-Residual Latent Space Decoupling Model (ExtRes)

Let us consider a dataset $\mathcal{D} = \{x^{(n)}\}_{n=1}^{N}$, consisting of N i.i.d. samples of stochastic variable $x \in \mathcal{X}$. Given $f_{ext}\colon \mathcal{X} \to \mathcal{Y}$ that extracts information of a target concept as $y \in \mathcal{Y}$ from x, ExtRes is for learning latent spaces $\mathcal{Z}_e$ and $\mathcal{Z}_r$ that correspond to f_{ext}. $\mathcal{Z}_e$ is called an *extraction space*, which captures the extracted target concept, and $\mathcal{Z}_r$ is called a *residual space*, which is expected to be decoupled from $\mathcal{Z}_e$, aiming at capturing a concept corresponding to all the rest. Figure 1 summarizes our model.

f_{ext} can be obtained e.g., by constructing rules or algorithms on the basis of the data domain knowledge (for musical sequences, see [2, 19]). Specific examples based on musical domain knowledge are shown in Sec.2.5.

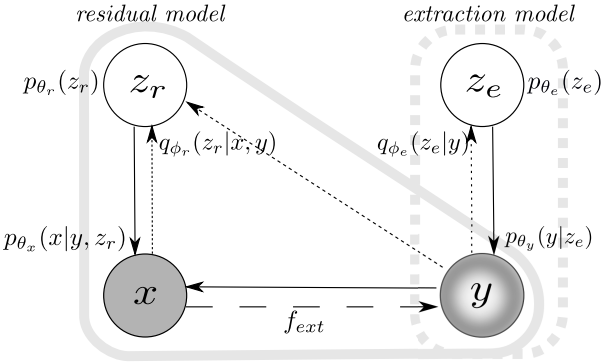


Figure 1: Graphical model of ExtRes.

First, we conduct *data derivation*: augmenting the dataset $\mathcal{D}$ to obtain $\mathcal{D}_{drv} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$, through the mapping $y = f_{ext}(x)$. Then, our approach is to learn a generative model involving two latent variables: $z_e \in \mathcal{Z}_e$ for capturing variability in y and $z_r \in \mathcal{Z}_r$ for variability in x given y. We assume the dataset is generated from the following process: (i) $z_e^{(n)} \sim p_{\theta_e^*}(z_e)$, $z_r^{(n)} \sim p_{\theta_r^*}(z_r)$, (ii) $y^{(n)} \sim p_{\theta_y^*}(y|z_e^{(n)})$, and (iii) $x^{(n)} \sim p_{\theta_x^*}(x|y^{(n)}, z_r^{(n)})$. Then we model this generative process by maximizing marginal log likelihood $\log p_\theta(\mathcal{D}_{drv}) = \sum_{n=1}^{N} \log p_\theta(x^{(n)}, y^{(n)})$ with each term rewritten as:

$$\log p_\theta(x,y) = \log \int p_{\theta_x}(x|y,z_r) p_{\theta_r}(z_r) dz_r \,, \\ + \log \int p_{\theta_y}(y|z_e) p_{\theta_e}(z_e) dz_e. \quad (1)$$

Since this is computationally intractable in general, we derive an evidence lower bound (ELBO) [16]:

$$\log p_\theta(x,y) \geq \mathcal{L}_{res} + \mathcal{L}_{ext}, \quad (2)$$

$$\text{where } \mathcal{L}_{res} = \mathbb{E}_{q_{\phi_r}(z_r|x,y)}\left[\log p_{\theta_x}(x|y,z_r)\right] \\ - D_{KL}\left(q_{\phi_r}(z_r|x,y)||p_{\theta_r}(z_r)\right), \quad (3)$$

$$\text{and } \mathcal{L}_{ext} = \mathbb{E}_{q_{\phi_e}(z_e|y)}\left[\log p_{\theta_y}(y|z_e)\right] \\ - D_{KL}\left(q_{\phi_e}(z_e|y)||p_{\theta_e}(z_e)\right). \quad (4)$$

Here, D_{KL} denotes the Kullback-Leibler (KL) divergence.

We maximize the data likelihood of right hand side of Eq.(2) for optimizing parameters $\{\theta_x, \theta_y, \theta_r, \theta_e, \phi_r, \phi_e\}$. We refer to the models corresponding to Eq.(3) and Eq.(4) as the *residual model* and *extraction model*, respectively.

2.3 Controlling Generation

Concept-Axes Interpolation. With our decoupled latent spaces, interpolation can be done for each latent space $\mathcal{Z}_e$ and $\mathcal{Z}_r$. In the simplest linear case, interpolation between two latent vectors $[z_e^{(i)}; z_r^{(i)}]$ and $[z_e^{(j)}; z_r^{(j)}]$ produces samples $x(\alpha_e, \alpha_r) \sim p_{\theta_x}(x|y(\alpha_e), z_r(\alpha_r))$ with $y(\alpha_e) \sim p_{\theta_y}(y|z_e(\alpha_e))$, where $z_r(\alpha_r) = z_r^{(i)} + \alpha_r(z_r^{(j)} - z_r^{(i)})$ and $z_e(\alpha_e) = z_e^{(i)} + \alpha_e(z_e^{(j)} - z_e^{(i)})$ with $(\alpha_r, \alpha_e) \in [0,1] \times [0,1]$.

Variation Generation Approach. Finding the *boundary*, in a latent space, between *variations* and *non-variations* is semi-automatically learned by defining that *variations* of a given data sample are the samples that contain enough information in terms of reconstruction error ϵ,

in expectation. For our ExtRes, the error ϵ can be adjusted by introducing trade-off parameters in Eq.(3) and Eq.(4) (see Sec.4.2). Intuitively, the latent vectors capture *high-level* features for the corresponding data samples, and given a data sample $\hat{x}$, the learned inference distributions $q_{\phi_r}(z_r|\hat{x}, f_{ext}(\hat{x}))$ and $q_{\phi_e}(z_e|f_{ext}(\hat{x}))$ specify feature-wise similar/dissimilar (i.e. *variations*/*non-variations*) *boundaries* of the given sample. Therefore, our approach is to generate variations $x^{(i)}$ of $\hat{x}$ in accordance with the *boundaries* as follows: $x^{(i)} \sim p_{\theta_x}(x|y^{(i)}, z_r^{(i)})$ with $y^{(i)} \sim p_{\theta_y}(y|z_e^{(i)})$ and $z_r^{(i)} \sim q_{\phi_r}(z_r|\hat{x}, f_{ext}(\hat{x}))$, where $z_e^{(i)} \sim q_{\phi_e}(z_e|f_{ext}(\hat{x}))$.

Concept-Aware Variation Generation. We also propose how to generate variations $x^{(i)}$ in a concept-wise manner when the inference models are normal distributions. The amount of variation is controlled for each concept in either of the following ways: simply changing the ratio of the covariance scales of inference distributions, or ordering the samples $z_e^{(i)}$ or $z_r^{(i)}$ on the basis of Mahalanobis distances:

$$D_M(z^{(i)}, \hat{z}) = \sqrt{(z^{(i)} - \hat{z})^{\mathrm{T}} \Sigma^{-1} (z^{(i)} - \hat{z})}, \quad (5)$$
$$\text{for } (z^{(i)}, \hat{z}, \Sigma) \in \{(z_e^{(i)}, \hat{z}_e, \Sigma_e), (z_r^{(i)}, \hat{z}_r, \Sigma_r)\},$$
$$\text{where } q_{\phi_e}(z_e|f_{ext}(\hat{x})) = \mathcal{N}(z_e|\hat{z}_e, \Sigma_e),$$
$$\text{and } q_{\phi_r}(z_r|\hat{x}, f_{ext}(\hat{x})) = \mathcal{N}(z_r|\hat{z}_r, \Sigma_r).$$

Here, $\mathcal{N}$ denotes normal distribution.

2.4 Instantiation of ExtRes for Sequences

Throughout the rest of this paper, we consider the case where x consists of a sequence of discrete variables s_t i.e. $x = (s_1, ..., s_T)$. Here, each s_t has a distribution over the elements of a finite alphabet set $\mathcal{A}$. Let f_{ext} be a function that maps a sequence of length T to that of length T, i.e. $f_{ext}\colon \mathcal{A}^T \to \mathcal{B}^T$, where $\mathcal{B}$ is another alphabet set. First, we conduct *data derivation* using $f_{ext}(x^{(n)}) = y^{(n)} = (a_1^{(n)}, ..., a_T^{(n)})$. The given dataset $\mathcal{D} = \{x^{(n)}\}_{n=1}^N$ can be augmented to become $\mathcal{D}_{drv} = \{(x^{(n)}, y^{(n)})\}_{n=1}^N$, where $x^{(n)} = (s_1^{(n)}, ..., s_T^{(n)})$ and $y^{(n)} = (a_1^{(n)}, ..., a_T^{(n)})$. We refer to $(a_1^{(n)}, ..., a_T^{(n)})$ as an *abstract sequence* of $(s_1^{(n)}, ..., s_T^{(n)})$. Now, the inference models are

$$(h_{z_e,t}, c_{z_e,t}) = \mathrm{LSTM}(\mathrm{E}(a_t), h_{z_e,t-1}, c_{z_e,t-1}), \quad (6)$$
$$q_{\phi_e}(z_e|a_{1:T}) = \mathcal{N}(z_e|\mathrm{MLP}(h_{z_e,T}), \mathrm{diag}(\exp(\mathrm{MLP}(h_{z_e,T})))), \quad (7)$$
$$(h_{z_r,t}, c_{z_r,t}) = \mathrm{LSTM}([\mathrm{E}(s_t); \mathrm{E}(a_t)], h_{z_r,t-1}, c_{z_r,t-1}), \quad (8)$$
$$q_{\phi_r}(z_r|s_{1:T}, a_{1:T}) = \mathcal{N}(z_r|\mathrm{MLP}(h_{z_r,T}), \mathrm{diag}(\exp(\mathrm{MLP}(h_{z_r,T})))), \quad (9)$$

where LSTM, E, and MLP denote a long short-term memory RNN (first, second, and third arguments of LSTM are input, hidden state, and cell state, respectively) [13], embedding layer, and multi-layer perceptron, respectively. The generative model for the *abstract sequence* is

$$p_{\theta_e}(z_e) = \mathcal{N}(z_e|\mathbf{0}, \mathcal{I}), \quad (10)$$
$$(h_{a,1}, c_{a,1}) = \mathrm{LSTM}([Ea_0; z_e], \mathrm{MLP}(z_e), c_{a,0}), \quad (11)$$
$$(h_{a,t}, c_{a,t}) = \mathrm{LSTM}([\mathrm{E}(a_{t-1}); z_e], h_{a,t-1}, c_{a,t-1}), \quad (12)$$
$$p_{\theta_y}(a_t|a_{1:t-1}, z_e) = \mathrm{Cat}(a_t|\sigma(\mathrm{MLP}(h_{a,t}))), \quad (13)$$
$$p_{\theta_y}(a_{1:T}|z_e) = \textstyle\prod_{t=1}^T p_{\theta_y}(a_t|a_{1:t-1}, z_e), \quad (14)$$

where Cat and σ denote the categorical distribution and softmax function, respectively. Note that we use notation $p_{\theta_y}(a_1|a_{1:0}, z_e) = p_{\theta_y}(a_1|z_e)$ for brevity. The generative model for the original sequence is

$$p_{\theta_r}(z_r) = \mathcal{N}(z_r|\mathbf{0}, \mathcal{I}), \quad (15)$$
$$(h_1, c_1) = \mathrm{LSTM}([Es_0; \mathrm{E}(a_1); z_r], \mathrm{MLP}(z_r), c_0), \quad (16)$$
$$(h_t, c_t) = \mathrm{LSTM}([\mathrm{E}(s_{t-1}); \mathrm{E}(a_t); z_r], h_{t-1}, c_{t-1}), \quad (17)$$
$$p_{\theta_x}(s_t|s_{1:t-1}, a_{1:t}, z_r) = \mathrm{Cat}(s_t|\sigma(\mathrm{MLP}(h_t))), \quad (18)$$
$$p_{\theta_x}(s_{1:T}|a_{1:T}, z_r) = \textstyle\prod_{t=1}^T p_{\theta_x}(s_t|s_{1:t-1}, a_{1:t}, z_r), \quad (19)$$

where we use the following notation for brevity: $p_{\theta_x}(s_1|s_{1:0}, a_{1:1}, z_r) = p_{\theta_x}(s_1|a_1, z_r)$.

2.5 Formulating Musical Domain Knowledge

On the basis of musical domain knowledge, we approximate musical concepts as $f_{ext}(x)$ for monophonic sequences. As mentioned in Sec.1, given an input musical sequence, many *abstract sequences* expressing important musical concepts can be derived [2, 19]. Among these, we demonstrate formulating two examples: *rhythm* and *contour*. Our finding is that a concept of musical transformations—fragmentation & consolidation, which is the main characteristic of similarity in musical sequences [20]—can also be approximated as a function, and we demonstrate it. We use the real name of musical notes as symbols ('C3', 'D#4', etc.) and use 'R' to represent a rest symbol. We add an extra symbol '__' representing that a note is held and not replayed [11]. Then the alphabet set $\mathcal{A}$ becomes a set of symbols listed above. Let $\mathcal{O}$ be a set of symbols that has no pitch, i.e., $\mathcal{O} = \{\text{'__'}, \text{'R'}\}$. We use the notation $\mathcal{S}_{1:t-1} = \{s_1^{(n)}, ..., s_{t-1}^{(n)}\}$.

Rhythm. Let 'N' denote a symbol that represents any note. Then the *rhythm* sequence for $(s_1^{(n)}, ..., s_T^{(n)})$ is defined as $(a_1^{(n)}, ..., a_T^{(n)})$, where

$$a_t^{(n)} = \begin{cases} s_t^{(n)} & (s_t^{(n)} \in \mathcal{O}) \\ \text{'N'} & (\text{otherwise}) \end{cases}$$

Thus, $\mathcal{B} = \mathcal{O} \cup \{\text{'N'}\}$. Now we define f_{ext} for *rhythm* as: $f_{ext}(x) = (a_1, ..., a_T)$.

Contour. In this paper, we refer to a chromatic signed contour with rhythm information as *contour*. Formally, the *contour* sequence for $(s_1^{(n)}, ..., s_T^{(n)})$ can be defined as $(a_1^{(n)}, ..., a_T^{(n)})$, where

$$a_t^{(n)} = \begin{cases} s_t^{(n)} & (s_t^{(n)} \in \mathcal{O}) \\ \text{'SP'} & (s_t^{(n)} \notin \mathcal{O} \wedge \mathcal{S}_{1:t-1} \subset \mathcal{O}) \\ \mathrm{sgn}(\mathrm{Interval}(s_t^{(n)})) & (\text{otherwise}) \end{cases}$$

Here 'SP' stands for Starting Pitch, sgn is a real sign function, and Interval is a function that calculates the chromatic pitch difference between the symbol of its argument and the previous symbol that has pitch. Then $\mathrm{sgn}(\mathrm{Interval}(s_t^{(n)}))$ outputs whether the pitch of $s_t^{(n)}$ is higher than ('1'), lower than ('-1'), or the same as ('0') the last symbol that has pitch. Thus, the alphabet set $\mathcal{B}$={'__', 'R', 'SP', '1', '-1', '0'}. Now we define f_{ext}

for *contour* as: $f_{ext}(x) = (a_1, ..., a_T)$.

Fragmentation and Consolidation (F&C). Fragmentation involves replacing one long note with several shorter notes, whereas consolidation conversely involves replacing several shorter notes with a single long note [20]. Formally, the *F&C*-invariant sequence for $(s_1^{(n)}, ..., s_T^{(n)})$ can be defined as $(a_1^{(n)}, ..., a_T^{(n)})$, where

$$a_t^{(n)} = \begin{cases} \text{'FC'} & (s_t^{(n)} \in \mathcal{O} \vee s_t^{(n)} = \text{LSP}(s_t^{(n)})) \\ s_t^{(n)} & (\text{otherwise}) \end{cases}$$

Here LSP stands for Last Symbol with Pitch, and $\text{LSP}(s_t^{(n)}) = s_{t_l}^{(n)}$, where $t_l = \max\{u \colon s_u^{(n)} \in \mathcal{S}_{1:t-1} \wedge s_u^{(n)} \notin \mathcal{O}\}$. Thus, the alphabet set $\mathcal{B} = \{\text{'FC'}\} \cup \mathcal{A} \setminus \mathcal{O}$. Now we define f_{ext} for *F&C* as: $f_{ext}(x) = (a_1, ..., a_T)$.

3. RELATED WORK

Conditional Models. Conditional deep generative models are widely studied especially in the image domain [15, 22]. Although our *residual model* is similar to this family of models, their methods are different from ours in that (i) the problem setting itself is different: data for conditioning variable y is given in their method, (ii) condition y is not as structured as ours (usually labels), and (iii) the latent space of condition $\mathcal{Z}_e$ is not learned (i.e., their method has no *extraction model* of ours). In the music domain, conditions of notes or chords are used for factoring out those information from latent variables [7, 28, 29].

Disentangled Latent Spaces. In the image domain, some approaches successfully disentangle latent space [3, 4, 14]. The popular approach is regularizing each latent dimension to be independent, hoping to obtain interpretable factors as attributes. Meanwhile, an approach that permits disentangling a latent space applicable to symbolic music has been proposed [10], although this method assumes an attribute has order.

Exploring Latent Space in Music. After learning the latent space, some methods attempt to discover a meaningful direction in a latent space [8, 27]. These methods are also useful for exploring within our individual concept spaces.

Variation Generation in Music. The differences between our *concept-aware variation generation* and the variation mechanism proposed by Pachet et al. [23] are (i) their method is based on a Markov model; (ii) their method controls variation generation in terms of edit operations, while our method controls it in terms of musical concepts; (iii) our ExtRes tries to learn the notion of *variation* (see Sec.2.3). Difference (iii) also differentiates our method from the previous VAE method for generating melody variation, wherein simply Gaussian noises are added to the latent vector to create perturbed latent vectors [29], although their method could produce more diverse variations. One can also use their method within our concept spaces.

Regularizing Latent Space in Music. Human dissimilarity ratings on timbre are utilized to regularize the latent space for bridging audio analysis, perception, and generation [9].

4. EXPERIMENTS

4.1 Dataset

We conduct experiments using a leadsheet dataset introduced by Pachet et al. [24] with more than 12,000 songs, by hundreds of famous songwriters, covering several genres of popular music: jazz, blues, pop, and rock. The dataset has been used in music generation studies [21, 23, 25, 26]. We extract all monophonic melody parts with time signature 4/4, which are transposed in all possible keys if the transposition remains within the midi pitch range of $[55, 84]$. We choose to discretize time with 24 symbols in a bar, where every beat has six symbols whose note-on timings in one beat are $\{0, 1/4, 1/3, 1/2, 2/3, 3/4\}$. We then extract all consecutive subsequences of length $T = 24$ (1 bar) or $T = 96$ (4 bars). The total dataset is split into proportion of $\{0.85, 0.1, 0.05\}$ for train, validation, and test data respectively.

4.2 Implementation Details

The numbers of dimensions for z_e or z_r are chosen to be $(16, 32)$ for the $T = (24, 96)$ model. 2-layer stacked LSTMs are employed. We introduce trade-off parameters β_1 for Eq.(3) and β_2 for Eq.(4) to weight the second terms [9, 27, 28]. Intuitively, the amount of information required for each latent variable depends on the target concepts: *rhythm*, *contour*, and *F&C*. Therefore, we conduct a hyper-parameter search using the validation dataset such that β_1 and β_2 are the maximum subject to reconstruction accuracies being sufficiently high. Then, β_1 and β_2 for *rhythm*, *contour*, and *F&C* are chosen to be $(\beta_1, \beta_2) = (0.7, 0.7)$, $(1.0, 0.7)$, and $(0.9, 0.7)$, respectively. The number of training epochs is set to 20, KL-annealing is used [1, 9], and teacher forcing is not used.

4.3 Model Performance

Decoupling Performance. We first sample sequences $\{x^{(i)} \colon i \in \{1, ..., I\}\}$ and $\{x^{(i,j)} \colon (i, j) \in \{1, ..., I\} \times \{1, ..., J\}\}$, following the sampling procedure:

$$x^{(i)} \sim p_{\theta_x}(x|y^{(i)}, z_r^{(i)}),\ x^{(i,j)} \sim p_{\theta_x}(x|y^{(i)}, z_r^{(j)}),$$
$$\text{with } z_r^{(i)} \sim p_{\theta_r}(z_r),\ z_r^{(j)} \sim p_{\theta_r}(z_r), \text{ and}$$
$$y^{(i)} \sim p_{\theta_y}(y|z_e^{(i)}), \text{ where } z_e^{(i)} \sim p_{\theta_e}(z_e).$$

For $x^{(i)}$ and $x^{(i,j)}$, we count

$$N_{i,j} = |\{t \in \{1, ..., T\} \colon f_{ext}(x^{(i)})_t = f_{ext}(x^{(i,j)})_t\}|, \quad (20)$$

which is how many elements of the same index are the identical symbol between *abstract sequences* of $x^{(i)}$ and $x^{(i,j)}$. Then, we define the accuracy for a pair $(x^{(i)}, x^{(i,j)})$ as $N_{i,j}/T$.

Figure 2 illustrates the results of cumulative decoupling accuracies for $I = 1000$ and $J = 100$. *F&C* accuracies are almost perfect. Even for *rhythm* and *contour*, $(T = 24, T = 96) = (99.6, 97.5)\%$ and $(85.0, 37.6)\%$ of the samples have perfect accuracies, respectively. Interestingly, for *rhythm* and *contour*, the cumulative percentage of samples grows sharply if one symbol mistake

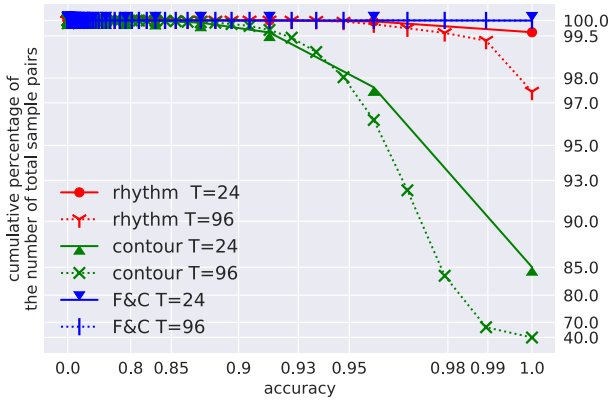


Figure 2: Decoupling accuracy.

	NLL		Accuracy	
	T=24	T=96	T=24	T=96
VAE (β=1.0)	0.714	0.577	0.880	0.880
VAE (β=0.8)	0.730	0.586	0.935	0.913
VAE (β=0.7)	0.753	0.606	0.952	0.923
VAE (β=0.5)	0.792	0.660	0.974	0.956
VAE (β=0.3)	0.856	0.748	0.988	0.960
Ours, *Rhythm*	0.412	0.322	0.967	0.969
	0.335	0.274	0.982	0.976
Ours, *Contour*	0.297	0.212	0.968	0.968
	0.468	0.405	0.973	0.938
Ours, *F&C*	0.141	0.130	0.978	0.975
	0.630	0.497	0.963	0.950

Table 1: Negative log-likelihood and reconstruction accuracy. For our models, upper and lower rows denote the *residual model* and *extraction model*, respectively.

is allowed. For instance, the percentage for *contour* grows from $(85.0, 37.6)\%$ to $(97.6, 66.3)\%$.

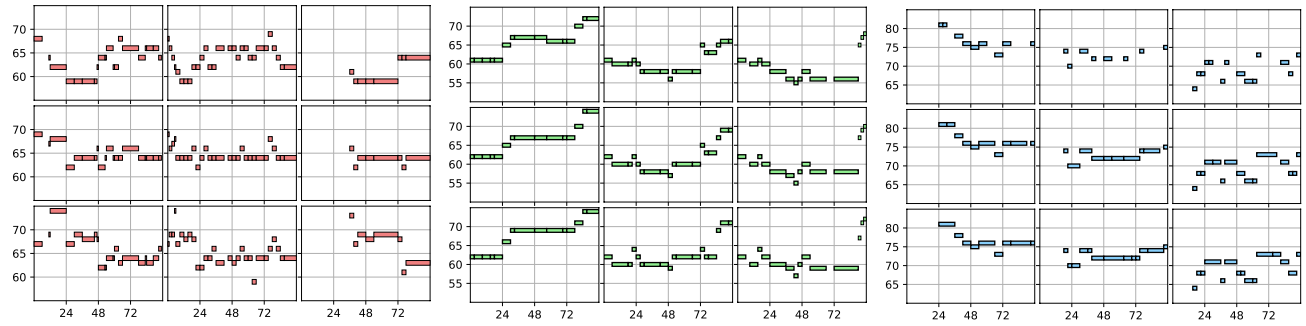
Modeling Performance. Table 1 shows negative log-likelihoods (NLLs; per symbol and lower bound) and reconstruction accuracies (accuracies) for the test dataset. We compare baseline variational auto-encoders (VAEs) [16] (its model implementation and the training algorithms are the same as our *residual model* without the condition y except that the number of dimensions for the latent variable is doubled for fair comparisons) with our three proposed models. Each of our models is divided into two models (*residual/extraction model* (see Sec.2.2)), whose individual NLLs and accuracies are shown in the table. The additions of two NLLs for the *residual/extraction model* are comparable to the NLLs for the baseline VAE. The multiplications of two accuracies for the *residual/extraction model* are also comparable to the accuracies for the baseline VAE.

4.4 Concept-Axes Interpolation

As depicted in Fig.4, given two musical fragments (top-left and bottom-right) in each subfigure with 5×5 fragments, the other 23 fragments "in between" are yielded using *concept-axes interpolation*, whereas the interpolation in a traditional latent space [27, 28] would produce only the three diagonal fragments. In these figures, horizontal axes are the *extraction space* $\mathcal{Z}_e$ axes, and moving towards the axes smoothly changes the extracted target concepts (i.e., *rhythm*, *contour*, and *F&C*-invariant), while generally not changing the *residual* concepts. On the other hand, vertical axes are the *residual space* $\mathcal{Z}_r$ axes, showing smooth change in *residual* concepts and little change in target concepts. Note that here we only show 5×5, but interpolation of the two fragments could be arbitrarily finer/coarser (at any positions of the subfigure) upon users' demand. In Fig.4a, *rhythm* direction preserves "pitch appearing order," showing that the $\mathcal{Z}_r$ successfully captures concepts that are important but might be difficult to learn in $\mathcal{Z}_e$. In Fig.4b, in non-*contour* direction, fragments tend to transpose to match the "pitch set" of the bottom-right fragment without changing *contour*, which captures the scale-like concept. In *contour* direction, the fragments gradually adopt the descending-like contour. In other words, only the descending-like feature is retrieved from the bottom-right fragment to generate fragments in the first row. For Fig.4c, in *F&C*-invariant direction, the gradual altering of fragmenting or consolidating notes is observed. Similar analyses can also be done in longer sequences: Figure 3 shows results for $T = 96$ in piano roll representation with each subfigure consisting of 3×3 musical fragments.

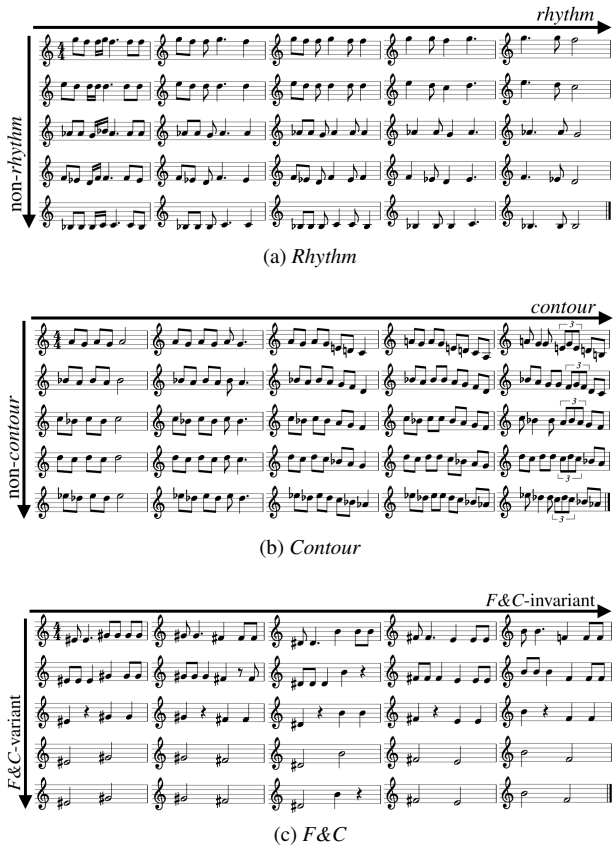
4.5 Concept-Aware Variation Generation

In Fig.5, our variation generation approach is applied to ExtRes/VAE, which are for concept-aware/-unaware variation generations, respectively. In each column of the figure, the generated variations are sorted from top to bottom in the ascending order of learned Mahalanobis distance (see Sec.2.3). Figure 5a depicts the variations for *rhythm*. The second column shows the *extraction space* $\mathcal{Z}_e$ variations, where various rhythms are produced without changing the other factors. In contrast, *residual space* $\mathcal{Z}_r$ variations (the third column) all keep the rhythm unchanged, whereas the other factors such as the "order of used pitches" change. Variations by VAE (the fourth column) mix factors of rhythm/non-rhythm, without drastic change in rhythm. Figure 5b depicts the variations for *contour*. $\mathcal{Z}_e$ variations have various contours without changing other factors. In contrast, $\mathcal{Z}_r$ variations all keeps the contour unchanged, whereas the scale-like concept "set of used pitches" changes. VAE yields relatively conservative variations with mixed contour/scale-like factors. Lastly, variations for *F&C* are in Fig.5c. For the $\mathcal{Z}_e$ variations, melodies with different pitches are generated, while fixing the concept of "the consecutive notes with the same pitch" except the third row from the bottom. As for the $\mathcal{Z}_r$ variations, F&C of notes are observed, which is not the case in the fourth column except the second row from the bottom, indicating that our ExtRes successfully captures the notion of F&C. Note that the variations by ExtRes are generated with simply $(0, 1)$ or $(1, 0)$ variance scales for two spaces to clearly explain the capabilities of ExtRes, but one could interactively change the scale ratio to obtain



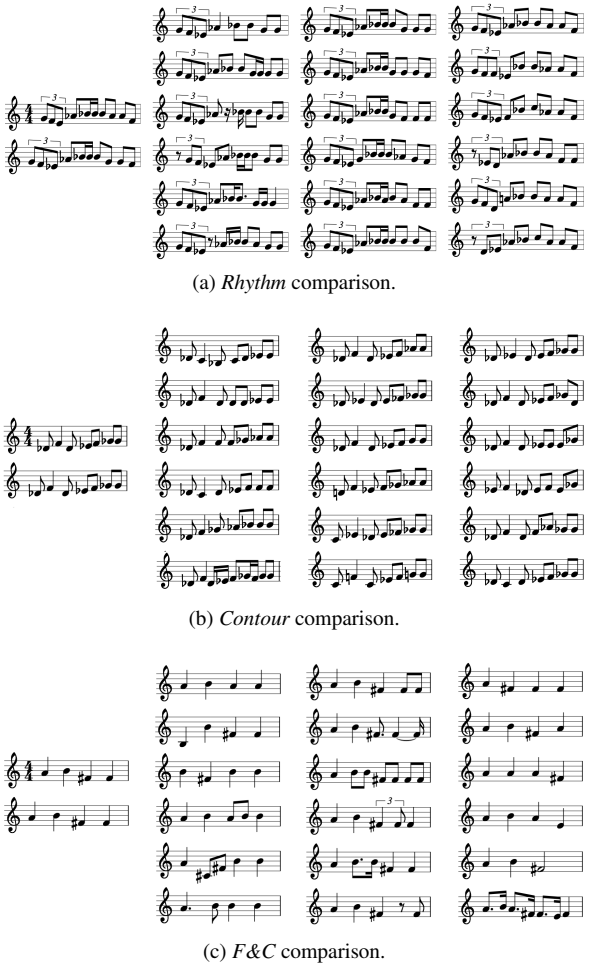
(a) *Rhythm* (b) *Contour* (c) *F&C*

Figure 3: *Concept-axes interpolation (T=96, 0.5 stride).* Vertical axis denotes MIDI note number, and horizontal axis denotes $t \in \{1, ..., T\}$.



(a) *Rhythm* (b) *Contour* (c) *F&C*

Figure 4: *Concept-axes interpolation (T=24, 0.25 stride).*



(a) *Rhythm* comparison. (b) *Contour* comparison. (c) *F&C* comparison.

Figure 5: Our variation generation approach is applied to ExtRes and VAE. For each subfigure (a,b,c), left (first column): top is an original fragment and bottom is its reconstruction; center left (second column): ExtRes *extraction space* $\mathcal{Z}_e$ variations; center right (third column): ExtRes *residual space* $\mathcal{Z}_r$ variations; right (fourth column): VAE variations.

variations with desired mixing proportions of the concepts.

5. CONCLUSION AND FUTURE WORK

We presented a latent space decoupling model for learning concept spaces using domain knowledge. For monophonic symbolic music, we experimented on three musical concepts. Controllability in generation was improved by *concept-axes interpolation* and *concept-aware variation generation*. In future, other musical concepts mentioned in Sec.1 should also be tested on ExtRes. We believe that this paper opens up possibilities for learning models with concept-aware inference/generative processes to be used for different information retrieval tasks or more controlled and flexible generations.

6. ACKNOWLEDGMENTS

I would like to thank Gaëtan Hadjeres for data encoding codes and reviewing my manuscript. I also greatly appreciate Frank Nielsen for helping me with writing the manuscript.

7. REFERENCES

[1] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proc. of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016.

[2] Emilios Cambouropoulos, Tim Crawford, and Costas S. Iliopoulos. Pattern processing in melodic sequences: challenges, caveats and prospects. *Computers and the Humanities*, 35(1):9–21, 2001.

[3] Tian Qi Chen, Xuechen Li, Roger B. Grosse, and David K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Annual Conference on Neural Information Processing Systems*, 2018.

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Annual Conference on Neural Information Processing Systems*, 2016.

[5] Walter Dowling. Scale and contour: Two components of a theory of memory for melodies. *Psychological Review*, 85:341–354, 07 1978.

[6] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martín Arjovsky, Olivier Mastropietro, and Aaron C. Courville. Adversarially learned inference. In *International Conference on Learning Representations*, 2017.

[7] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. GANSynth: Adversarial neural audio synthesis. In *International Conference on Learning Representations*, 2019.

[8] Jesse Engel, Matthew Hoffman, and Adam Roberts. Latent constraints: Learning to generate conditionally from unconditional generative models. In *International Conference on Learning Representations*, 2018.

[9] Philippe Esling, Axel Chemla-Romeu-Santos, and Adrien Bitton. Bridging audio analysis, perception and synthesis with perceptually-regularized variational timbre spaces. In *Proc. of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*.

[10] Gaëtan Hadjeres, Frank Nielsen, and François Pachet. GLSR-VAE: Geodesic latent space regularization for variational autoencoder architectures. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI 2017)*, pages 1–7. IEEE, 2017.

[11] Gaëtan Hadjeres, François Pachet, and Frank Nielsen. DeepBach: a steerable model for Bach chorales generation. In *Proc. of the 34th International Conference on Machine Learning*, 2017.

[12] Irina Higgins, Nicolas Sonnerat, Loic Matthey, Arka Pal, Christopher P Burgess, Matko Bošnjak, Murray Shanahan, Matthew Botvinick, Demis Hassabis, and Alexander Lerchner. SCAN: Learning hierarchical compositional visual concepts. In *International Conference on Learning Representations*, 2018.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[14] Arka Pal Christopher Burgess Xavier Glorot Matthew Botvinick Shakir Mohamed Alexander Lerchner Irina Higgins, Loic Matthey. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[15] Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Annual Conference on Neural Information Processing Systems*, 2014.

[16] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[17] Stefan Lattner, Maarten Grachten, and Gerhard Widmer. Learning transposition-invariant interval features from symbolic music and audio. In *Proc. of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*.

[18] Steven Livingstone, Caroline Palmer, and Emery Schubert. Emotional response to musical repetition. *Emotion (Washington, D.C.)*, 12:552–67, 06 2011.

[19] David Meredith. The ps13 pitch spelling algorithm. *Journal of New Music Research*, 35, 06 2006.

[20] Marcel Mongeau and David Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175, 1990.

[21] Simon Moulieras and François Pachet. Maximum entropy models for generation of expressive music. *CoRR*, abs/1610.03606, 2016.

[22] Siddharth Narayanaswamy, T. Brooks Paige, Jan-Willem van de Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. Learning disentangled representations with semi-supervised deep generative models. In *Annual Conference on Neural Information Processing Systems*, 2017.

[23] François Pachet, Alexandre Papadopoulos, and Pierre Roy. Sampling variations of sequences for structured music generation. In *Proc. of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*.

[24] François Pachet, Jeff Suzda, and Dani Martínez. A comprehensive online database of machine-readable lead-sheets for jazz standards. In *In Proc. of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013*.

[25] François Pachet. A joyful ode to automatic orchestration. *ACM TIST*, 8:18:1–18:13, 2016.

[26] Alexandre Papadopoulos, Pierre Roy, and François Pachet. Assisted lead sheet composition using flowcomposer. In *CP*, 2016.

[27] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In *Proc. of the 35th International Conference on Machine Learning*, 2018.

[28] Ian Simon, Adam Roberts, Colin Raffel, Jesse Engel, Curtis Hawthorne, and Douglas Eck. Learning a latent space of multitrack measures. *CoRR*, abs/1806.00195, 2018.

[29] Yifei Teng, Anny Zhao, and Camille Goudeseune. Generating nontrivial melodies for music as a service. In *Proc. of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*.