

SELF-SUPERVISED DISENTANGLEMENT OF HARMONIC AND RHYTHMIC FEATURES IN MUSIC AUDIO SIGNALS

Yiming Wu

AlphaTheta Corporation
Yokohama, Japan

yiming.wu@alphatheta.com

ABSTRACT

The aim of latent variable disentanglement is to infer the multiple informative latent representations that lie behind a data generation process and is a key factor in controllable data generation. In this paper, we propose a deep neural network-based self-supervised learning method to infer the disentangled rhythmic and harmonic representations behind music audio generation. We train a variational autoencoder that generates an audio mel-spectrogram from two latent features representing the rhythmic and harmonic content. In the training phase, the variational autoencoder is trained to reconstruct the input mel-spectrogram given its pitch-shifted version. At each forward computation in the training phase, a *vector rotation* operation is applied to one of the latent features, assuming that the dimensions of the feature vectors are related to pitch intervals. Therefore, in the trained variational autoencoder, the rotated latent feature represents the pitch-related information of the mel-spectrogram, and the unrotated latent feature represents the pitch-invariant information, *i.e.*, the rhythmic content. The proposed method was evaluated using a predictor-based disentanglement metric on the learned features. Furthermore, we demonstrate its application to the automatic generation of music remixes.

1. INTRODUCTION

Deep neural network (DNN)-based data generation techniques are increasingly used in creative fields. In the audio domain, exciting new methods have been proposed for speech generation, music composition, and sound design. The main advantage of DNNs is their high expressiveness in approximating the real-world data distributions, which can provide consistent generation results that are convincing to human creators. However, because of their highly complicated architecture, the interpretability and controllability of the generative process have become the two main problems with DNN-based data generation. A DNN contains a huge number of stochastically optimized parameters, and hence it is impossible to explain how each parameter or each internal output influences the final output. In addition, a DNN-based generative method often introduces a stochastic process, which improves the diversity of the generation results, but also makes it more difficult for the users to control the output and obtain results that reflect their intentions.

Disentanglement learning is a key approach to solving the problems of interpretability and controllability. Disentanglement learning aims to model the generative process conditioned by multiple *disentangled* latent variables, *i.e.*, a set of independent vari-

Copyright: © 2023 Yiming Wu. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, adaptation, and reproduction in any medium, provided the original author and source are credited.

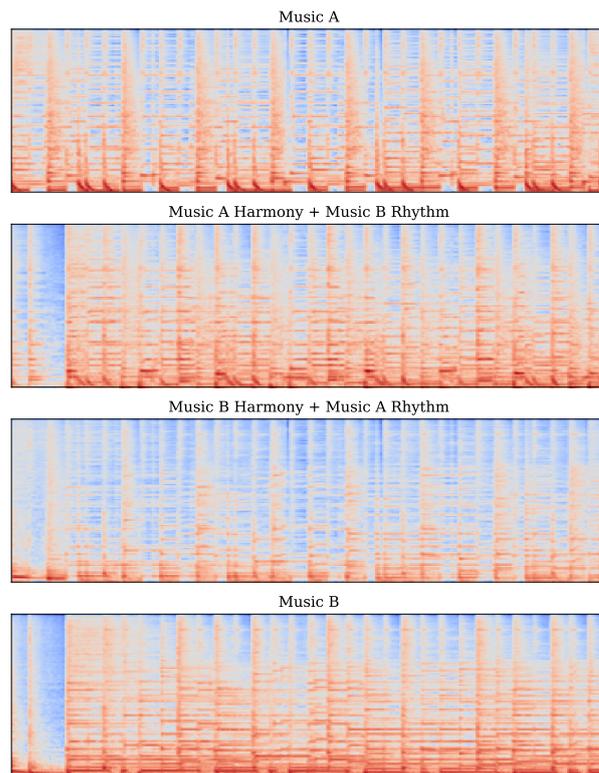


Figure 1: An example of music remix generation by the proposed harmony-rhythm disentanglement method. The middle two spectrograms were generated by combining the harmony and rhythm contents of different music.

ables that are sensitive only to certain factors of the observed data. For example, studies in the speech domain focus on the representations of speaker identity, gender, speed of speech, and emotions [1, 2]. Generative models with properly disentangled latent variables make it easier to explicitly reflect the intentions of human users in the generation results.

In this paper, we focus on disentanglement learning for generative models of musical audio. More specifically, our goal is to learn the disentangled latent features of the rhythmic and harmonic content in musical audio. For human listeners, the rhythmic content of a piece of music is derived from the onset timings of the musical audio, and the harmonic content is derived from the different pitches of the musical audio. Therefore, these two types of content are considered to be independent of each other. In the

time-frequency representations (such as spectrograms) of musical audio, the rhythmic and harmonic content can be observed in the temporal progressions along the time and frequency axes, respectively, and this can be used to implement a harmonic-percussive source separation algorithm [3]. We assume that harmonic and rhythmic content can also be separated in latent space.

We propose a simple training method to obtain the disentangled latent features by introducing several constraints during the training process. It involves training a generative model for music audio spectrogram using a variational autoencoder (VAE), in which the encoder network maps the input spectrogram to the latent features while the decoder network maps the latent features back to the audio spectrogram. The key idea behind our approach is to let the VAE not only reconstruct the input spectrogram, but also reverse the transformation applied to the input spectrogram. In the proposed method, the transformation is audio pitch-shifting. We assume that pitch-shifting on the musical audio only changes its harmonic content and not its rhythmic content. By introducing a vector rotation on the harmonic latent feature to reverse the pitch shift operation, the rotated and unrotated latent features can be trained without supervision to represent the pitch-related and pitch-invariant information in musical audio, respectively.

The main contribution of this work is to propose an effective disentanglement learning method that is suitable for DNN-based music audio generation models. In the evaluation section, we show the quality of disentanglement quantitatively using a predictor-based metric. We also explore the application of the proposed method to the automatic generation of music remixes, by replacing the rhythmic (or harmonic) feature of one musical audio clip with that of another musical audio clip. The quantitative evaluations and concrete audio examples demonstrate that the proposed method can generate realistic music remixes that possesses the characteristics of both sources of music.

2. RELATED WORK

This section reviews related work on DNN-based generation and disentanglement learning for musical audio.

2.1. DNN-based Musical Audio Generation

Several different approaches have been proposed for DNN-based music audio generation. One popular approach is based on differential digital signal processing (DDSP) [4], in which the generative model is concatenated with audio DSP modules such as filters and oscillators. DNNs are then trained to estimate the parameters of these DSP modules. Because DDSP-based generative models utilize strong inductive biases, they are generally more interpretable, and require fewer audio examples to achieve reasonable generalized performance. Therefore, DDSP has been applied in several existing synthesizer algorithms, such as wavetable synthesizer [5], waveshaping synthesizer [6], FM synthesizer [7], and the WORLD vocoder [8].

Another approach is the autoencoding approach, which trains a DNN-based generative model and its latent feature space using an autoencoder network. Once the autoencoder has been trained, musical audio can be generated by manipulating the latent feature and reconstructing the audio using the decoder network. More specifically, one can interpolate over the latent feature space like RAVE [9], or train the language model of the latent feature to generate musical audio from scratch, as in Jukebox [10], Musika [11],

and MusicLM [12].

2.2. Disentanglement Learning for Audio

The main goal of disentanglement learning for audio is to implement audio transformation systems that change certain aspects of the musical content, such as timbre or musical styles. For example, Noam et al. proposed a music translation method that transforms the domain (musical instruments and styles) of musical audio [13]. The method is based on a multi-domain autoencoder based on WaveNet [14], where the encoder WaveNet transforms the audio waveform into a domain-independent latent representation, and the domain-specific WaveNet decoders reconstruct the audio waveform from the latent representation. To make the encoder extract the domain-independent representation from audio waveforms, the encoder is trained to fool a domain classifier network that tries to correctly recognize the domain type from the latent representation. This approach is not a fully unsupervised method because a domain label should be given for each musical audio clip used to train the neural networks.

Studies on disentanglement learning for audio have proposed several learning schemes to automatically separate the pitch-related and pitch-invariant information in the musical audio in the latent space of an audio generative model. Luo et al. proposed a learning method to encode the pitch and timbre of musical instrument sounds using Gaussian mixture VAE [15], where the latent representations were learnt in a supervised and semi-supervised manner using pitch and instrument annotations. GANStrument proposed by Narita et al. introduces an adversarial training scheme to extract pitch-invariant features from musical instrument sound [16]. Using the trained feature encoder, GANStrument can generate pitched instrument sounds given a one-shot sound as input. Luo et al. also proposed an unsupervised learning method to encode the pitch and timbre of musical instrument sounds, in which the pitch is represented as a discrete label and the timbre is represented as a continuous feature vector [17]. Similar to our proposed method, they assume that a moderate pitch shift operation does not change the timbre of the original musical instrument sound. Based on this assumption, they treat the original sound and its pitch-shifted version as a pair, and swap the encoded pitch variables before reconstructing the musical sound using the decoder. Because the pitch is represented as a single discrete variable, this method is suitable for monophonic musical sound. Our proposed method formulates a VAE in a similar way; however, we formulate the pitch-related feature as continuous value vectors, so that these vectors can represent the polyphonic pitch information found in any kind of musical recording.

3. PROPOSED METHOD

This section describes the proposed self-supervised disentanglement learning method. An overview of the proposed method is shown in Fig. 2. We formulate a probabilistic generative model representing the generative process of an audio mel-spectrogram from two latent features representing harmony and rhythm in the form of a VAE (Section 3.1). In the training phase, we use an audio pitch-shifting algorithm to enable the model to learn the two latent features that represent the pitch-related and pitch-invariant information of the input audio (Section 3.2). In addition, we train the decoder as a generative adversarial network (GAN) to improve the generation quality (Section 3.3).

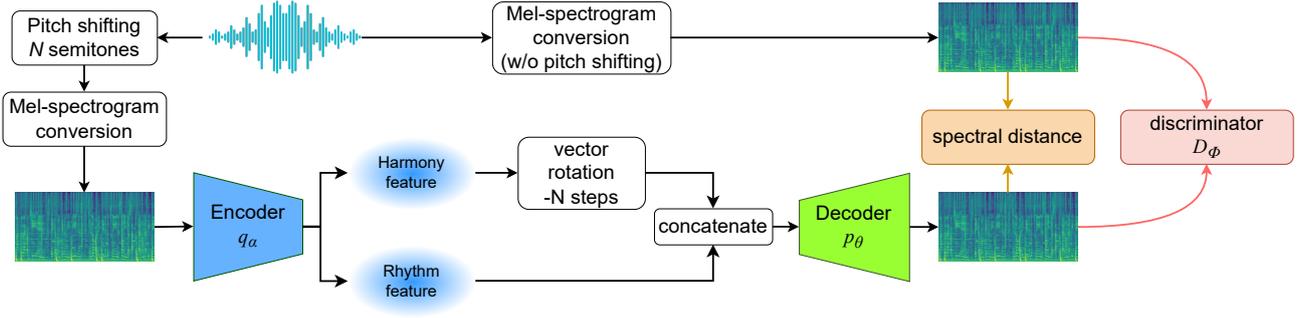


Figure 2: Proposed VAE architecture and its forward computation procedure.

3.1. VAE Formulation

Let $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ be a log-scaled mel-spectrogram of a musical audio, represented as a sequence of D-bins spectrum $\mathbf{x}_n \in \mathcal{R}^D$. Let $\mathbf{Z}^h = \{\mathbf{z}_n^h\}_{n=1}^N$ and $\mathbf{Z}^r = \{\mathbf{z}_n^r\}_{n=1}^N$ be sequences of latent features, where $\mathbf{z}_n^h, \mathbf{z}_n^r \in \mathcal{R}^L$ are L -dimensional continuous-valued vectors ($L = 128$) that abstractly represent the harmonic and rhythmic content at the n th audio frame, respectively. We formulate a generative model with \mathbf{X} as the observed variable and $\mathbf{Z}^h, \mathbf{Z}^r$ as the latent features as follows:

$$p(\mathbf{X}) = p_\theta(\mathbf{X}|\mathbf{Z}^h, \mathbf{Z}^r)p(\mathbf{Z}^h)p(\mathbf{Z}^r) \quad (1)$$

where $p_\theta(\mathbf{X}|\mathbf{Z}^h, \mathbf{Z}^r)$ is a conditional generative model with parameters θ . We define p_θ as a decoder neural network parametrized by θ . The decoder network models the generative process of mel-spectrogram from the two latent features \mathbf{Z}^h and \mathbf{Z}^r . In our work, we evaluate $p_\theta(\mathbf{X}|\mathbf{Z}^h, \mathbf{Z}^r)$ using the spectral distance between \mathbf{X} and the output of the decoder network $\omega_\theta(\mathbf{Z}^h, \mathbf{Z}^r)$:

$$p_\theta(\mathbf{X}|\mathbf{Z}^h, \mathbf{Z}^r) \sim S_\theta(\mathbf{X}, \mathbf{Z}^h, \mathbf{Z}^r) \stackrel{\text{def}}{=} \|\mathbf{X} - \omega_\theta(\mathbf{Z}^h, \mathbf{Z}^r)\|_1 \quad (2)$$

where $\|\cdot\|_1$ is the $L1$ norm.

Since the inference model of the latent features $p(\mathbf{Z}^h, \mathbf{Z}^r|\mathbf{X})$ is intractable, we use a neural encoder network q_α that approximates the distributions of the latent features given an observed mel-spectrogram as follows:

$$q_\alpha(\mathbf{Z}^h|\mathbf{X}) = \prod_{n=1}^N \mathcal{N}(\mathbf{z}_n^h | \mu_\alpha(\mathbf{X})_n^h, \sigma_\alpha(\mathbf{X})_n^h) \quad (3)$$

$$q_\alpha(\mathbf{Z}^r|\mathbf{X}) = \prod_{n=1}^N \mathcal{N}(\mathbf{z}_n^r | \mu_\alpha(\mathbf{X})_n^r, \sigma_\alpha(\mathbf{X})_n^r) \quad (4)$$

where $\mu_\alpha(\mathbf{X})^h, \sigma_\alpha(\mathbf{X})^h, \mu_\alpha(\mathbf{X})^r$, and $\sigma_\alpha(\mathbf{X})^r$ are the four parts of the encoder network output.

The priors $p(\mathbf{Z}^h)$ and $p(\mathbf{Z}^r)$ are set to a standard Gaussian distribution as follows:

$$p(\mathbf{Z}^r) = \prod_{n=1}^N \mathcal{N}(\mathbf{z}_n^r | \mathbf{0}_L, \mathbf{I}_L), \quad (5)$$

$$p(\mathbf{Z}^h) = \prod_{n=1}^N \mathcal{N}(\mathbf{z}_n^h | \mathbf{0}_L, \mathbf{I}_L), \quad (6)$$

As shown in Fig.3, the encoder neural network is composed of stacked residual convolution layers and downsampling layers. Two independent bottleneck modules are appended to the bottom layer

of the encoder to compute the parameters of the two latent distributions. Each downsampling layer is implemented with a strided convolution layer that reduces the dimension of the frequency axis of the mel-spectrogram by a factor of four while keeping the dimension of the time axis unchanged. Therefore, the encoder reduces the frequency axis of the input spectrogram by a factor of 64, and outputs the latent features with two dimensions on the frequency axis. Similarly, the decoder neural network is composed of stacked residual convolution layers and upsampling layers that are implemented with strided transposed convolution layers, each of which expands the frequency-axis by a factor of four.

3.2. Self-Supervised Disentanglement Learning

In a normal VAE setting [18], the generative model is trained within the framework of variational inference, which jointly optimizes the encoder and decoder network to maximize the evidence lower bound (ELBO) of the observed data likelihood $p(x)$ as:

$$\begin{aligned} \mathcal{L}_{VAE_{normal}} = & \mathbb{E}_{q_\alpha(\mathbf{z}^r, \mathbf{z}^h|\mathbf{X})} [\log p_\theta(\mathbf{X}|\mathbf{Z}^r, \mathbf{Z}^h)] \\ & - \beta \mathcal{D}_{KL}(q_\alpha(\mathbf{Z}^h|\mathbf{X})||p(\mathbf{Z}^h)) - \beta \mathcal{D}_{KL}(q_\alpha(\mathbf{Z}^r|\mathbf{X})||p(\mathbf{Z}^r)) \end{aligned} \quad (7)$$

where $\mathcal{D}_{KL}(q||p)$ is the KL divergence from distribution q to p , and β is a weighting factor that controls the trade-off between the reconstruction accuracy and level of disentanglement within the latent features [19]. The latent variable regularization term in ELBO encourages disentanglement between each dimension of the latent variable [19]. However, without explicit conditioning, there is no guarantee that the latent variables learn to explicitly represent the harmonic (or rhythmic) aspects of the mel-spectrogram.

To distinguish the harmonic and rhythmic content of musical audio, we make the assumption that rhythmic content is invariant to audio pitch shifting, whereas harmonic content is not. Assuming that the musical audio share the same tuning (e.g., tuned to 440Hz), we add a definition of the dimensions of the latent vector \mathbf{z}_n^h : the i -th dimension $z_{n_i}^h$ represents the pitch information of a certain pitch height, and the pitch intervals between the pitches corresponding to the i -th and j -th dimension is $j-i$ times of a small pitch interval unit (we use *semitone* in the following statements). In this way, we can relate audio pitch-shifting to a *vector rotation* operation on \mathbf{z}_n^h , i.e., when \mathbf{Z}^h is the harmony feature of \mathbf{X} , the n -step vector rotation of \mathbf{Z}^h is the harmony feature of the n -semitone pitch shift version of \mathbf{X} .

Based on our definition of \mathbf{Z}^h , we designed a training procedure to facilitate the harmony-rhythm disentanglement. Con-

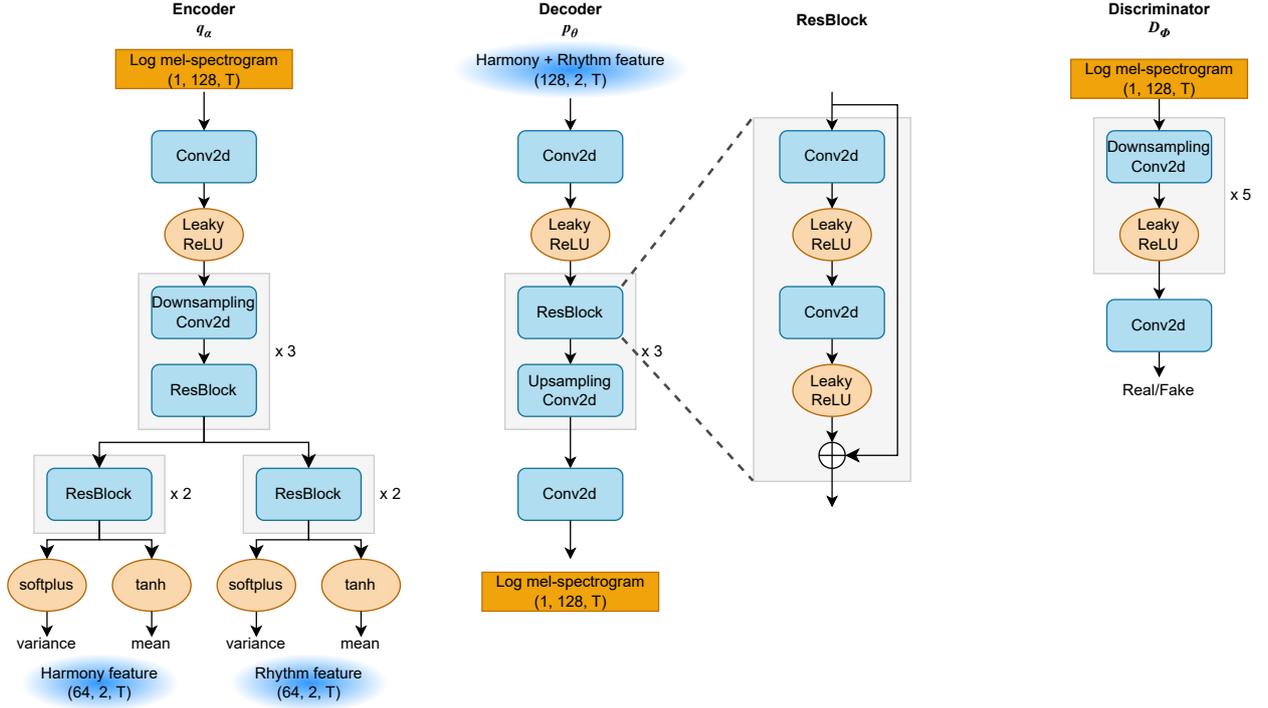


Figure 3: Proposed VAE architecture and its forward computation procedure.

cretely, each forward computation in a training iteration proceeds follows:

1. Shift the pitch of the input audio segment by a random number of semitones $n \in [-8, 8]$. Let \mathbf{X}' be the mel-spectrogram of the pitch-shifted audio,
2. Calculate the latent feature distribution $q_\alpha(\mathbf{Z}^{ih}, \mathbf{Z}^{ir} | \mathbf{X}')$ using the encoder network,
3. Sample the latent features $\mathbf{Z}^{ih}, \mathbf{Z}^{ir}$ from $q_\alpha(\mathbf{Z}^{ih}, \mathbf{Z}^{ir} | \mathbf{X}')$ using the reparameterization trick [18],
4. Apply $(-n)$ -step vector rotation to the channel dimension of \mathbf{Z}^{ih} . Let \mathbf{Z}^h be the rotated latent feature.
5. Reconstruct the mel-spectrogram from $p_\theta(\mathbf{X} | \mathbf{Z}^h, \mathbf{Z}^{ir})$ using the decoder network.

Combining Equation 7 with Equation 2, the training objective of the VAE is:

$$\begin{aligned} \mathcal{L}_{VAE} = & \mathbb{E}_{q_\alpha(\mathbf{Z}^{ih}, \mathbf{Z}^{ir} | \mathbf{X}')} [S_\theta(\mathbf{X}, \mathbf{Z}^{ih}, \mathbf{Z}^{ir})] \\ & - \beta \mathcal{D}_{KL}(q_\alpha(\mathbf{Z}^{ih} | \mathbf{X}') || p(\mathbf{Z}^h)) - \beta \mathcal{D}_{KL}(q_\alpha(\mathbf{Z}^{ir} | \mathbf{X}) || p(\mathbf{Z}^{ir})) \end{aligned} \quad (8)$$

We set $\beta = 0.1$ in our experiment, which places more weight on the reconstruction accuracy. The expectation term is approximated by the Monte Carlo method using the reparameterization trick. Intuitively, the VAE decoder is trained to reconstruct the original mel-spectrogram \mathbf{X} given the latent variables encoded from the pitch-shifted mel-spectrogram \mathbf{X}' . Because \mathbf{Z}^{ir} is not altered during the forward computation, it should represent the pitch-invariant elements in \mathbf{X}' and \mathbf{X} . By contrast, because the vector rotation on \mathbf{Z}^{ih} reverts the pitch shift on \mathbf{X} , the rotated variable \mathbf{Z}^h is able to

represent the pitch-specific elements of the original \mathbf{X} . Therefore, unlike \mathbf{Z}^r , \mathbf{Z}^h should represent the pitch-related elements in \mathbf{X} during the optimization.

3.3. GAN Learning

To improve the quality of the generated mel-spectrogram, the VAE networks are also trained as a GAN [20]. We additionally define a discriminator network D_ϕ that learns to distinguish the generated mel-spectrogram from the original mel-spectrogram. The GAN training objective is defined as follows:

$$\mathcal{L}_{dis} = (1 - D_\phi(\mathbf{X}))^2 + D_\phi(\hat{\mathbf{X}})^2 \quad (9)$$

$$\mathcal{L}_{gen} = -D_\phi(\hat{\mathbf{X}})^2 \quad (10)$$

where \mathbf{X} is the original spectrogram and $\hat{\mathbf{X}}$ is the spectrogram reconstructed by the VAE. To stabilize the adversarial training process, a feature matching loss \mathcal{L}_{FM} [21] is further added to the training objective. Altogether, the objective function for the VAE network optimization is

$$\mathcal{L}_{total} = \mathcal{L}_{VAE} + \mathcal{L}_{gen} + \mathcal{L}_{FM}$$

Following the ordinary GAN training procedure, the discriminator network is trained to minimize \mathcal{L}_{dis} , and the VAE network is trained to minimize \mathcal{L}_{total} . As illustrated in Fig. 3, the discriminator network is composed of five convolutional layers with leaky ReLU activation.

Combination of VAE and GAN objectives is also used to train the RAVE [9] and Musika! [11] audio synthesizer. Unlike RAVE, our method does not optimize the VAE and GAN objectives separately. We also do not fix the parameters of the encoder network.

In our experiments, the objective \mathcal{L}_{total} jointly optimizes the encoder and decoder network.

4. EVALUATION

This section reports the comparative experiment conducted to evaluate the effectiveness of the proposed disentanglement learning method. The experiments were implemented using PyTorch [22], and the source code is available on GitHub.¹

4.1. Datasets

We use the *fma-large* subset of the Free Music Archive (FMA) dataset [23] to train the VAE. The dataset contains 30-second musical audio snippets from 106,574 Creative Commons-licensed music tracks. To measure the quality of the rhythm–harmony disentanglement of the proposed method, we use the RWC-Popular dataset [24] as the test set. The RWC-Popular dataset contains 100 pieces of popular song audio with chord progression annotations. Following the common automatic chord estimation setting, the annotated chord labels are reduced to the *major* and *minor* triads.

The mel-spectrogram was computed from the audio signal using a sample rate of 22,050Hz. The FFT size, window length, and hop size of the short-time Fourier transform were set to 2048, 2048, and 512 samples, respectively, and the number of mel frequency bins was set to 128 (thus $D = 128$). Hann window was used for FFT computation.

A general-purpose audio pitch-shifting algorithm was used to obtain the pitch-shifted versions of the musical audio. In our experiments, we used the pitch-shifting function implemented in the *Pedalboard* audio processing library,² which wraps the *Rubber Band* audio stretching library.³ The *Rubber Band* audio stretching algorithm is based on the phase-vocoder method that uses phase resets on the percussive transients, an adaptive stretch ratio between phase reset points, and a "lamination" method to improve vertical phase coherence. In contrast to the naive phase-vocoder time stretching algorithm implemented in *librosa* [25] and *torchaudio*, *Rubber Band*'s algorithm can preserve percussive sounds without noticeable distortion.

4.2. Evaluation Metrics

We use a predictor-based evaluation metric similar to that used in [17] to measure the disentanglement between the inferred rhythm and harmony features. Specifically, a sequence classification model based on a two-layer bidirectional gated recurrent unit (GRU) network was trained to predict the chord labels and onset states from the audio features \mathbf{Z}^h , \mathbf{Z}^r , or the original audio mel-spectrogram \mathbf{X} . The accuracy of chord label prediction was measured by the frame-wise label overlap rate, and the accuracy of onset prediction was measured by the binary F-1 score over the onset positions.

The accuracy of chord prediction and onset prediction measures how well the latent features reflect the pitch-related and pitch-invariant information of the audio, respectively. If \mathbf{Z}^r and \mathbf{Z}^h are well disentangled, the classifiers on \mathbf{Z}^r should yield high accuracy for onset prediction and low accuracy for chord label predic-

tion. Similarly, the classifiers on \mathbf{Z}^h should yield high accuracy for chord label prediction and low accuracy for onset prediction.

The RWC-Popular dataset is divided into a training set (90%) and an evaluation set (10%). The data pairs of musical audio and chord label annotations in the RWC-Popular dataset were used to train and evaluate the chord label classifier. Similarly, the musical audio and onset label data pairs were used to train and evaluate the onset label classifier, where the onset label was inferred from the raw music audio using the onset detection algorithm implemented in the *librosa* library.

We further explore the application of the proposed method to the automatic generation of music remixes. To generate music remixes, we used the trained VAE to generate audio spectrograms that simultaneously contain the musical elements of two different music tracks. Given two pieces of beat-synchronized music A and B, a remix was created by the following process:

1. Infer the latent representations \mathbf{Z}_A^h , \mathbf{Z}_A^r , \mathbf{Z}_B^h , and \mathbf{Z}_B^r of the mel-spectrograms \mathbf{X}_A and \mathbf{X}_B using the encoder network,
2. Generate the mel-spectrogram from \mathbf{Z}_A^h , \mathbf{Z}_B^r using the decoder network.

We used the Fréchet Inception Distance (FID) [26] to quantitatively measure the quality of the generated spectrograms. The FID measure is given by:

$$F(\mathcal{N}_b, \mathcal{N}_e) = \|\mu_b - \mu_e\|^2 + \text{tr}(\Sigma_b + \Sigma_e - 2\sqrt{\Sigma_b \Sigma_e}) \quad (11)$$

where $\mathcal{N}_b(\mu_b, \Sigma_b)$ is the multivariate normal distribution estimated from the Inception V3 [27] features calculated from a set of spectrograms of the real musical audio, and $\mathcal{N}_e(\mu_e, \Sigma_e)$ is the distribution calculated from the generated spectrograms. The generated spectrograms are considered to be more musically realistic if the computed FID is low. The feature extractor is a pre-trained music genre classifier that was trained using the genre-annotated musical audio in the FMA dataset.

We used the following music remixing methods as the baselines:

- **HPSS.** We apply the harmonic-percussive source separation (HPSS) algorithm [28] in the *librosa* library to music A and music B, and mix the harmonic part of A and percussive part of B to create the remix version. The HPSS algorithm infers the spectral masks for harmonic and percussive parts using median-filtering along the time and frequency axis.
- **ASAP.** We use the *Spectral Morphing* audio effect implemented in the *ASAP* plug-in suite developed by IRCAM.⁴ The *Spectral Morphing* plugin combines the spectral characteristics of two audio signals using the source-filter technique where the audio signal of music B is used as a filter of the audio signal of music A. More specifically, the frequency-domain amplitude of the two audio signals are multiplied, while preserving the phase of the source audio. The spectral envelope of music B is further applied to the filtered signal. We set music A as the main input, music B as the sidechain input, and set the *Global Mix* parameter to 100% to generate the remixed version.

We randomly chose 20 songs from the RWC-Popular dataset to create 10 pairs of audio clips. Each audio clip was time-stretched

¹<https://github.com/WuYiming6526/HARD-DAFx2023>

²<https://spotify.github.io/pedalboard/reference/pedalboard.html>

³<https://breakfastquay.com/rubberband/>

⁴<https://forum.ircam.fr/projects/detail/asap/>

Table 1: *Harmony-rhythm disentanglement Metrics*

Feature	chord accuracy	onset F1
harmony feature	69.61%	60.09%
rhythm feature	24.65%	66.04%
mel-spectrogram	51.95%	65.19%

Table 2: *FIDs of the generated spectrogram*

Model	FID
HPSS	12.84
ASAP	13.18
Disentangled VAE (proposed)	12.46

to 120 BPM and was 8s long. Therefore, the FID for each compared method was computed on 10 audio clips generated by the corresponding method. The remixes created by the proposed and the baseline methods can be found on the online project page.⁵ Hifi-GAN [29] was used to convert the mel-spectrograms generated by the proposed method into an audio signal.

4.3. Results

Table 1 compares the accuracy of chord classification and onset detection for different audio features. The overall chord classification accuracy for the harmony feature was much higher than for the rhythm feature. The chord labels were almost unpredictable from the rhythm features because these features were trained to be pitch-invariant. By contrast, the beat detection score was higher for the rhythm features than for the harmony features by a much smaller margin. Although the rhythm features were better at representing onset information, the harmony features were not completely onset-invariant. This is somewhat inevitable, since onsets can be inferred in part from pitch transitions. Interestingly, both harmony and rhythm features scored higher than the mel-spectrogram representation in the chord classification and onset detection tasks, respectively. Since the latent features enhance the pitch-related and pitch-invariant elements in the musical audio, it is reasonable that the latent features were found to be more suitable for the pitch-related or rhythm-related music information retrieval tasks. This result indicates that the proposed method can also be used as a self-supervised pre-training method to provide better feature representations for other music information retrieval tasks.

As a qualitative evaluation, we visualized the latent harmony and rhythm representations. Fig. 4 compares the visualized latent features of a song from the RWC-Popular dataset with the ground-truth MIDI pianorolls. It can be seen that the harmony feature had similar pitch progressions to the ground-truth pianoroll. The rhythm features were relatively sparse, and there was no obvious correlation with the pitches or onsets of the ground-truth pianoroll.

As shown in Table 2, the remix generated by the proposed method achieved a better FID score than the baseline methods, suggesting that the proposed method generated spectrograms that are closer to real audio spectrograms than the baseline methods. This is a promising result as it indicates that the proposed method has the potential to generate high-quality remixes. The **HPSS** method simply replaced the percussive part of music A with music B, so the harmonic part does not change. The **ASAP** method

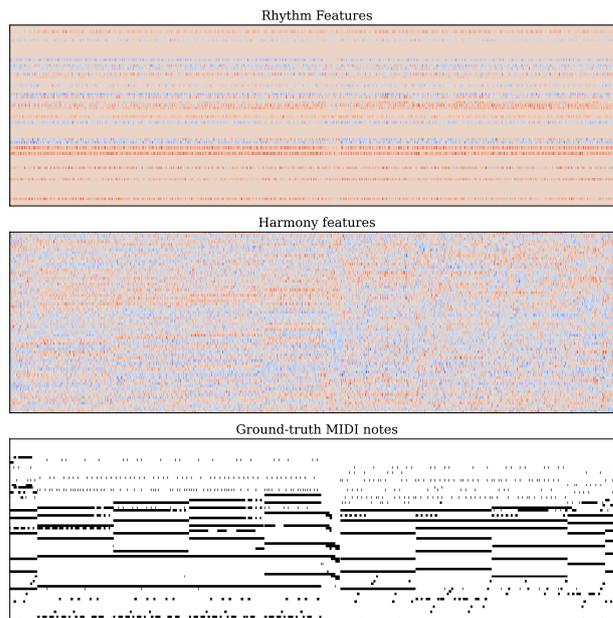


Figure 4: *Visualizations of the rhythm and harmony features of a song from the RWC-Popular dataset. The bottom figure visualizes the MIDI notes from the ground-truth MIDI file.*

added some rhythmic elements of music B to the audio of music A through the dynamic filtering effect, but the rhythmic sounds of music A were still present. In contrast to these baseline methods, the proposed method reflected the rhythmic elements of music B more clearly. Unlike the results of the **HPSS** method, all of the generated audio, including the harmonic part of music A, reflect the rhythm of music B. Unlike the results generated by the **ASAP** method, the rhythm of music A was removed and only the rhythm of music B was present.

5. CONCLUSION

We proposed a simple self-supervised learning method for inferring the disentangled rhythm and harmony features of musical audio. Through quantitative metrics and qualitative observations, we showed that the rhythm and harmony features obtained using the proposed method achieved a high degree of disentanglement. We also demonstrated its potential use for the automatic generation of music remixes.

The generative models that can be used in the proposed method are not limited to spectrogram-based models. In principle, the disentanglement learning strategy can be applied to any kind of autoencoder-based audio generation model, including time domain-based generative models such as RAVE and SoundStream [30]. However, the relationship between the time-domain audio signal and the audio pitch shift is less clear than it is in the time-frequency audio representations. Therefore, disentanglement learning using time domain audio signals may be practically more challenging. In our initial experiments, disentanglement learning on the time-domain generation models did not perform as well as it did with the mel-frequency domain model. The solution to this problem is left for future research.

⁵<https://wuyiming6526.github.io/HARD-demo/>

We also believe that the application of the proposed generative model is not limited to music audio generation. The proposed method could potentially be a pre-training method for downstream music information retrieval tasks. For example, the disentangled acoustic representation of harmony and rhythm may be suitable for musical notes, chords, or beat transcription tasks. Combining the encoder of the proposed VAE with the music transcription model would be worth exploring to push the boundaries of the automatic music transcription research.

6. ACKNOWLEDGMENT

This work has been supported by AlphaTheta Corporation. We thank Kimberly Moravec, PhD, from Edanz (<https://jp.edanz.com/ac>) for editing a draft of this manuscript .

7. REFERENCES

- [1] Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, Patrick Nguyen, and Ruoming Pang, “Hierarchical generative modeling for controllable speech synthesis,” in *International Conference on Learning Representations (ICLR)*, 2019, pp. 1–27.
- [2] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A. Saurous, “Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis,” in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 5180–5189.
- [3] Derry FitzGerald, “Harmonic/percussive separation using median filtering,” in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFX10)*, 2010, pp. 1–4.
- [4] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts, “DDSP: Differentiable digital signal processing,” in *International Conference on Learning Representations (ICLR)*, 2020, pp. 1–19.
- [5] Siyuan Shan, Lamtharn Hantrakul, Jitong Chen, Matt Avent, and David Trevelyan, “Differentiable wavetable synthesis,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 4598–4602, arXiv, version: 2.
- [6] Ben Hayes, Charalampos Saitis, and George Fazekas, “Neural waveshaping synthesis,” in *Proceedings of the 22rd International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 254–261.
- [7] Franco Caspe, Andrew McPherson, and Mark Sandler, “DDX7: Differentiable FM synthesis of musical instrument sounds,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 608–616.
- [8] Shahan Nercessian, “Differentiable WORLD synthesizer-based neural vocoder with application to end-to-end audio style transfer,” *arXiv preprint arXiv:2208.07282*, 2022.
- [9] Antoine Caillon and Philippe Esling, “RAVE: A variational autoencoder for fast and high-quality neural audio synthesis,” *arXiv preprint arXiv:2111.05011*, 2021.
- [10] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.
- [11] Marco Pasini and Jan Schlüter, “Musika! fast infinite waveform music generation,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 543–550.
- [12] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank, “MusicLM: Generating music from text,” *arXiv preprint arXiv:2301.11325*, 2023.
- [13] Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman, “A universal music translation network,” in *International Conference on Learning Representations (ICLR)*, 2019, pp. 1–13.
- [14] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukchoglu, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.
- [15] Yin-Jyun Luo, Kat Agres, and Dorien Herremans, “Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 746–753.
- [16] Gaku Narita, Junichi Shimizu, and Taketo Akama, “GANStrument: Adversarial instrument sound synthesis with pitch-invariant instance conditioning,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [17] Yin-Jyun Luo, Kin Wai Cheuk, Tomoyasu Nakano, Masataka Goto, and Dorien Herremans, “Unsupervised disentanglement of pitch and timbre for isolated musical instrument sounds,” in *Proceedings of the 21th International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 700–707.
- [18] Diederik P. Kingma and Max Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2014.
- [19] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner, “ β -VAE: Learning basic visual concepts with a constrained variational framework,” in *International Conference on Learning Representations (ICLR)*, 2017, pp. 1–22.
- [20] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, Cambridge, MA, USA, 2014, NIPS’14, p. 2672–2680.
- [21] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestein, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville, “Melgan: Generative adversarial networks for conditional waveform synthesis,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 1–12.

- [22] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, “Automatic differentiation in PyTorch,” in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*, 2017.
- [23] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson, “FMA: A dataset for music analysis,” *arXiv preprint arXiv:1612.01840*, 2017.
- [24] Masataka Goto, “RWC music database: Popular, classical, and jazz music databases,” in *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR)*, 2002, pp. 287–288.
- [25] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, “librosa: Audio and music signal analysis in python,” in *Python in Science Conference*, 2015, pp. 18–24.
- [26] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, “GANs trained by a two time-scale update rule converge to a local nash equilibrium,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 6629–6640.
- [27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna, “Rethinking the inception architecture for computer vision,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.
- [28] Jonathan Driedger, Meinard Müller, and Sascha Disch, “Extending harmonic-percussive separation of audio signals,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 611–616.
- [29] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, “HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020, pp. 17022–17033.
- [30] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi, “SoundStream: An end-to-end neural audio codec,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2022.