COLLAGENET: FUSING ARBITRARY MELODY AND ACCOMPANIMENT INTO A COHERENT SONG

Abudukelimu Wuerkaixi[1] **Christodoulos Benetatos**[2] **Zhiyao Duan**[2] **Changshui Zhang**[1]
[1] Institute for Artificial Intelligence, Tsinghua University (THUAI),
State Key Lab of Intelligent Technologies and Systems,
Beijing National Research Center for Information Science and Technology (BNRist),
Department of Automation, Tsinghua University Beijing, P.R.China
[2] Department of Electrical and Computer Engineering, University of Rochester
wekxabdk21@mails.tsinghua.edu.cn, c.benetatos@rochester.edu
zhiyao.duan@rochester.edu, zcs@mail.tsinghua.edu.cn

ABSTRACT

When writing pop or hip-hop music, musicians sometimes sample from other songs and fuse the samples into their own music. We propose a new task in the symbolic music domain that is similar to the music sampling practice and a neural network model named CollageNet to fulfill this task. Specifically, given a piece of melody and an irrelevant accompaniment with the same length, we fuse them into harmonic two-track music after some necessary changes to the inputs. Besides, users are involved in the fusion process by providing controls to the amount of changes along several disentangled musical aspects: *rhythm* and *pitch* of the melody, and *chord* and *texture* of the accompaniment. We conduct objective and subjective experiments to demonstrate the validity of our model. Experimental results confirm that our model achieves significantly higher level of harmony than rule-based and data-driven baseline methods. Furthermore, the musicality of each of the tracks does not deteriorate after the transformation applied by CollageNet, which is also superior to the two baselines. [1]

1. INTRODUCTION

Recent years witnessed growing interest in symbolic multi-track music generation with the development of deep neural networks [1–3]. In particular, generating an accompaniment for a given melody has been a topic of interest [4]. Current deep learning models for accompaniment generation and music arrangement focus on the generation quality. Only a few methods incorporate user control into the generation process [5, 6].

In this work, we present a new task in the scope of multi-track music generation, specifically, *music fusion*. Taking multiple unrelated music tracks as input, the task is to fuse them into a harmonic multi-track music piece, with some necessary changes to the input tracks; To involve users into the fusion process, users can control how much and on what aspects each input track can be changed. This task is similar to the music sampling practice, which started from hip-hop, and has been influencing pop and electronic music writing as well [7, 8]: Musicians sample melodies, rhythmic patterns, or other musical elements from other songs and fuse them into a new composition after certain changes [9, 10]. Our proposed task can be viewed as the first step towards the automation of the sampling practice. This task opens new possibilities in music arrangement and style fusion, and may lead to many creative applications involving user interaction into the music generation process.

In this paper, we concentrate on the fusion of a monophonic melody and an irrelevant polyphonic accompaniment with the same length. Specifically, we propose a neural network model named *CollageNet* to fuse the two tracks. We use two pretrained VAEs [11], one for the melody and the other for the accompaniment. The melody VAE computes a latent representation that disentangles *pitch* and *rhythm*, while the accompaniment VAE computes a latent representation that disentangles *chord* and *texture* [12, 13]. We then use adversarial training [14, 15] to train an actor model G to apply necessary transformations to the latent representations and decode them back to musical notes, to achieve a harmonic fusion of the two tracks while preserving a similarity to their original content. Because the latent representations are disentangled, the G model allows users to control the amount of changes along the disentangled musical aspects relatively independently by manipulating their corresponding latent vectors. An example of the input and output of the fusion process is displayed in Figure 1.

As this is a new task, there is no existing method to compare with. We therefore design two baselines, a rule-based method and a data-driven method. Objective and subjective experiments show that CollageNet significantly improves the harmony between the two tracks while

[1] Code is available at https://github.com/urkax/CollageNet

maintaining the similarities with the original input along user specified aspects. The achieved level of harmony is close to that of human-composed songs and is significantly higher than that of the two baselines. Besides, results also show that the musicality of each individual track does not deteriorate after the fusion.

The key contributions of this paper are as follows:

- We put forward a new task on symbolic multi-track music fusion, which is similar to the music sampling practice in music writing of modern genres.
- We propose a neural method, which allows users to control the degree of changes along several disentangled aspects of the input tracks in the fusion process.
- Objective and subjective experiments show that our proposed method outperforms two baseline methods in terms of harmony and musical quality.

2. RELATED WORK

2.1 Multi-track Music Generation

Multi-track music generation aims at generating music containing several tracks (parts) with different musical characters but constituting a pleasing whole. Some research focuses on harmonizing or accompanying a music track in an offline fashion [16] or an online fashion [17,18], while others focus on learning the representation of multi-track music [19–23]. DeepBach was proposed to generate Bach chorales using a graphical model [20]. Yan et al. proposed a part-invariant neural model to learn a representation of multi-part music [21]. Dong et al. proposed three models in different scenarios for multi-track generation using the GAN framework [22]. Simon et al. used a hierarchical VAE to model multi-track music [23].

2.2 Controllable Music Generation

There has been much attention to controllable generation in the image domain, such as CVAE [24, 25] and CGAN [26]. In recent years, there are also growing research interest in controllable symbolic music generation. Researchers proposed models to control quantifiable low-level musical attributes like note density, etc. Hadjeres et al. proposed a constrained method to train a VAE model with a regularized latent space [27]. Similarly, Pati et al. used a regularization loss within a mini-batch to train a controllable VAE model [28]. As for high-level musical features like musical arousal, Tan et al. proposed Music FaderNets to control them by sliding the corresponding low-level attributes. Music FaderNets are trained by first modelling the low-level attributes and learn the high-level features through semi-supervised clustering. [29].

2.3 Latent Space Transformations

There have been some studies in the image and text domain that learn transformations in the latent space. Engel et al. proposed to impose attributes on generated images through transformations in the VAE latent space [15]. Similar idea was applied in music domain for connective fusion [30]. Shen et al. proposed to disentangle textual content from style by learning a shared content latent space for texts in different style [31]. Mueller et al. proposed to improve the input sequence by optimizing its latent vector of VAE [32].

3. PROPOSED METHOD

In this paper, we propose a new user-guided method that can transform and combine a two-measure-long melody and an unrelated accompaniment into harmonic two-track music while maintaining a similarity to their original content. Specifically, we encode the melody and the accompaniment with the encoders of two disentangled VAEs respectively. Then an actor model applies necessary transformations to the pairs of latent representations, and the decoders of the VAEs decode them back to musical notes. The actor model G is trained against the critic model D with adversarial training.

3.1 Model Architecture

Our model is based on the disentangled VAE framework [12, 13], where the encoder takes an input x and outputs a posterior $q(z|x)$ for the latent vector z to sample from, and the decoder $p(x|z)$ reconstructs the input. The latent vector z disentangles different musical aspects, each of which is encoded by a certain part of the vector. Given a pair of melody and accompaniment, we use the encoders of two VAEs to encode each to a latent vector. Specifically, we use EC^2-VAE [12] to encode the melody input. The disentangled latent vector z_{mel} is a concatenation of a vector for *pitch* z_p and a vector for *rhythm* z_r, i.e., $z_{mel} = z_p \oplus z_r$ [2]. For the polyphonic accompaniment, we use the disentangled VAE in [13] to compute the latent vector z_{acc}, which is a concatenation of a *chord* vector z_c and a *texture* vector z_t, i.e., $z_{acc} = z_c \oplus z_t$..

After encoding the pair of melody and accompaniment into latent vectors z_{mel} and z_{acc}, we feed them to the actor model G which transforms them into another latent vector pair $\hat{z}_{mel}$ and $\hat{z}_{acc}$. The actor model applies changes to the latent vectors to achieve transformations on the music content and the pair is supposed to be more harmonic. By applying different amount of changes to different parts of the latent vectors, the degree of transformations is controlled along the different music aspects. The inference process is displayed in Figure 2 (b).

While the encoders and decoders are pre-trained, the actor model G is trained under an adversarial framework together with a critic model D. The critic model is a binary classifier to distinguish positive samples and negative samples of the latent vectors. The definition of the positive and negative samples is described in Section 3.2. It is noted that the D model is not used in the inference process.

3.2 Training

Firstly, we pre-train the two VAEs for melodies and accompaniments. They are specially designed to learn a seman-

[2] $\oplus$ denotes for concatenation

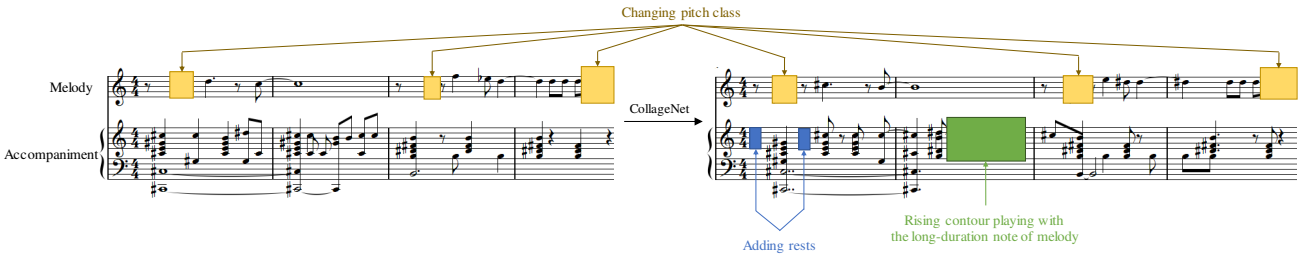


Figure 1. Example fusion result of CollageNet. An irrelevant pair of melody and accompaniment (left) is fused into a more harmonic pair while similarities to the input tracks are maintained. The control parameters are set to $c_{mp} = 0, c_{mr} = 1, c_{ac} = 0.8, c_{at} = 0$, so that *rhythm* of melody and *chord* of accompaniment are more preserved, while *pitch* of melody and *texture* of accompaniment are more altered. The bar lines are for clear visualization, not musically meaningful.

tically disentangled latent space, but fundamentally, they are both trained to maximize the evidence lower bound (ELBO) [12, 13]. The posterior $q(z|x)$ of VAE is trained to be close to the prior $p(z)$, which is the standard normal distribution. We obtain prior samples utilized for adversarial training by sampling latent vectors from $p(z)$.

Afterwards, we adversarially train the G and D models in the latent space. The training diagram is given in Figure 2 (a). Our dataset consists of two-track music segments, each of which has a monophonic melody track and a polyphonic accompaniment track. Suppose there are N music segments $\{x^{(i)}_{mel}, x^{(i)}_{acc}\}_{i=1}^{N}$, with the i-th melody segment indicated as $x^{(i)}_{mel}$ and the i-th accompaniment segment indicated as $x^{(i)}_{acc}$. We define a pair of melody and accompaniment with the same data index as a *harmonic pair* $\{x^{(i)}_{mel}, x^{(i)}_{acc}\}$, and define the set of harmonic pairs as the *harmonic pair set* Ω_h. To create disharmonic pairs accordingly, we randomly pick a melody $x^{(i)}_{mel}$ and an accompaniment $x^{(j)}_{acc}$ from the dataset with different data indexes ($i \neq j$). The *disharmonic pair* is indicated as $\{x^{(i)}_{mel}, x^{(j)}_{acc}\}$, and the *disharmonic pair set* is denoted as Ω_{dh}.

As discussed in Section 3.1, the D model is trained to distinguish between positive samples and negative samples. Positive samples are latent vectors of the harmonic pairs, indicated as $\{z_{mel}, z_{acc}\} \sim \Omega^z_h$. Negative samples include: (1) latent vectors of the disharmonic pairs $\{z_{mel}, z_{acc}\} \sim \Omega^z_{dh}$, (2) latent vectors sampled from prior $\{z_{mel}, z_{acc}\} \sim p(z)$, and (3) latent vectors produced by the actor model $G(z_{mel}, z_{acc})$. Following [15], we introduce the shorthand:

$$\begin{aligned} \mathcal{L}_{c=1}(z_{mel}, z_{acc}) &\triangleq -\log(D(z_{mel}, z_{acc})), \\ \mathcal{L}_{c=0}(z_{mel}, z_{acc}) &\triangleq -(1 - \log(D(z_{mel}, z_{acc}))). \end{aligned} \tag{1}$$

The training loss of the critic model D is as follows:

$$\begin{aligned} \mathcal{L}_D = &\mathop{\mathbb{E}}_{\{z_{mel}, z_{acc}\} \sim \Omega^z_h} [\mathcal{L}_{c=1}(z_{mel}, z_{acc})] \\ &+ \mathop{\mathbb{E}}_{\{z_{mel}, z_{acc}\} \sim N^z} [\mathcal{L}_{c=0}(z_{mel}, z_{acc})] \\ &+ \mathop{\mathbb{E}}_{\{z_{mel}, z_{acc}\} \sim N^z} [\mathcal{L}_{c=0}(G(z_{mel}, z_{acc}))], \\ &\text{where } N^z = \Omega^z_{dh} \cup p(z). \end{aligned} \tag{2}$$

The actor model G is trained to transform disharmonic latent vector pairs into a more harmonic pair. In other words, it is trained to fool the D model. For simplicity, we omit user control for this subsection, and we have: $\{\hat{z}_{mel}, \hat{z}_{acc}\} = G(z_{mel}, z_{acc})$. The outputs $\{\hat{z}_{mel}, \hat{z}_{acc}\}$ are expected to be close to the inputs $\{z_{mel}, z_{acc}\}$ to preserve a similarity on musical content. Therefore, the training loss of the actor model G consists of both an adversarial loss $\mathcal{L}_{Ga}$ and a distance loss $\mathcal{L}_{Gd}$. The adversarial loss is:

$$\mathcal{L}_{Ga} = \mathop{\mathbb{E}}_{\{z_{mel}, z_{acc}\} \sim N^z} [\mathcal{L}_{c=1}(G(z_{mel}, z_{acc}))]. \tag{3}$$

The distance loss is to constrain the distance between the output and the input of the G model. For clarity, we define a distance function $\rho(\hat{z}, z) \triangleq \frac{1}{d_z} \| \frac{1}{\bar{\sigma}^2_z} \log(1 + (\hat{z} - z)^2) \|_1$ for two latent vectors z and $\hat{z}$ in $\mathbb{R}^{d_z}$. The $\bar{\sigma}_z$ is the averaged scale of distribution $q(z|x)$ over the training set: $\bar{\sigma}_z = \frac{1}{N} \sum_n \sigma_z(x_n)$. We scale the distance penalty by the reciprocal of $\bar{\sigma}^2_z$ because latent vector dimensions with a smaller average scale contribute more to the identity of decoded data samples x [15]. Ignoring user control here, the distance loss is defined as follows:

$$\begin{aligned} &\mathcal{L}_{Gd} = \rho(\hat{z}_{mel}, z_{mel}) + \rho(\hat{z}_{acc}, z_{acc}), \\ &\text{where } \{\hat{z}_{mel}, \hat{z}_{acc}\} = G(z_{mel}, z_{acc}). \end{aligned} \tag{4}$$

The loss of the G model is the sum of these two parts, with distance penalty scaled by λ:

$$\mathcal{L}_G = \mathcal{L}_{Ga} + \lambda \mathcal{L}_{Gd}. \tag{5}$$

3.3 User Control

As discussed in Section 3.1, the latent vectors of melodies and accompaniments are disentangled into shorter vectors related to particular musical aspects. Specifically, $z_{mel} = z_p \oplus z_r$, and $z_{acc} = z_c \oplus z_t$. Users can control the amount of changes along these four musical aspects during the fusion process. To achieve that, aside from $\{z_{mel}, z_{acc}\}$, the input of the G model is extended with four scalars $c_{mp}, c_{mr}, c_{ac}, c_{at} \in [0, 1]$. These scalars respectively control *pitch* and *rhythm* of melodies, *chord* and *texture* of accompaniments.

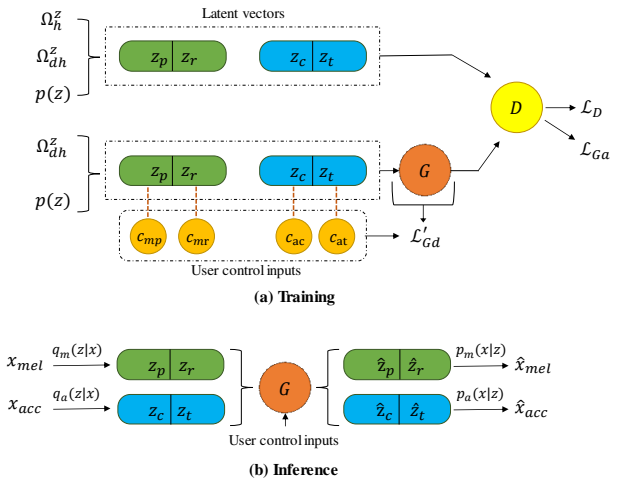


Figure 2. The training diagram of the G and D model and the inference diagram. The $q_m(z|x)$ and $p_m(x|z)$ are the encoder and decoder of the VAE for melodies. The $q_a(z|x)$ and $p_a(x|z)$ are from the VAE for accompaniments.

To impose such constraints on the G model, we randomly sample the scalars from the standard uniform distribution $U(0,1)$ for each data sample during training. The distance penalty of latent vectors is scaled by the corresponding scalars. Therefore, the distance loss in Eqn (4) becomes:

$$\begin{aligned}\mathcal{L}'_{Gd} &= c_{mp}\cdot\rho(\hat{z}_p, z_p) + c_{mr}\cdot\rho(\hat{z}_r, z_r)\\ &\quad + c_{ac}\cdot\rho(\hat{z}_c, z_c) + c_{at}\cdot\rho(\hat{z}_t, z_t),\\ &\text{where } \{\hat{z}_p\oplus\hat{z}_r, \hat{z}_c\oplus\hat{z}_t\} =\\ &\quad G(z_p\oplus z_r, z_c\oplus z_t, c_{mp}, c_{mr}, c_{ac}, c_{at}).\end{aligned} \tag{6}$$

After being trained with distance loss $\mathcal{L}'_{Gd}$, the actor model G can respond differently to the control input. For instance, if the user adjusts c_{mp} to a large value, then the G model will produce $\hat{z}_p$ close to the input z_p. Thus the *pitch* feature of the melody will hardly change after the transformation.

3.4 Implementation Details

For the disentangled VAEs, we use the same settings as original papers [12, 13], except that for EC^2-VAE we do not use conditional information. The data representation for melodies and accompaniments also follow the VAE papers. Both the G and D model take in latent vectors z_{mel} and z_{acc}. Different from [30], we sample from $q(z|x)$ to get latent vectors. Before concatenation, each of z_{mel} and z_{acc} are passed through linear layers and ReLU activation. Then the concatenated vector is passed through 8-layer blocks made up of linear layers with 1024 outputs, ReLU activation, and dropout layers with rate of 0.5. For the output of the G model, we use the gate mechanism following [15]. The G and D models are trained using the Adam optimizer [33], with learning rate of 3e-5, β_1 of 0, and β_2 of 0.9.

4. EXPERIMENTS

4.1 Dataset

We use the POP909 dataset [34], which contains melodies of 909 popular songs. Professional musicians composed piano accompaniments for them. We choose the songs with the time signature of 4/4 and randomly split them into 80%:10%:10% for training, validation, and test sets. Then we extract 8-beat long segments from them with a stride of 1 beat. We randomly select 40k segments for the training set, 5k segments for the validation set and 5k for the test set. We quantize time to 16th notes, so each segment is 32 steps long and we augment the training data by transposing them to all 12 keys.

4.2 Baseline Methods

As this is a new task, there is no existing methods to compare with. Therefore, we design a data-driven method and a rule-based method as baselines.

The data-driven baseline is derived from the proposed method. It also trains a critic model D to distinguish between harmonic pairs and disharmonic pairs. However, different from the proposed method, the D model in the data-driven baseline is pre-trained without the terms involving G. During the inference process, we use the pre-trained D model to implement gradient optimization $GradientDescent(z_{mel}, z_{acc}; \mathcal{L}_{c=1}(z_{mel}, z_{acc}))$. In other words, we optimize the inputs z_{mel} and z_{acc} to maximize the output of the pre-trained D model. We use the Adam optimizer [33] and the learning rate of 0.005 for both z_{mel} and z_{acc}.

The rule-based baseline applies revisions to the pitches and onsets of the melodies. According to music theory, to create a harmonic accompaniment for a melody, their notes should be on the same scale [35]. Besides, they need to be composed of matched rhythm. To fuse a pair of unrelated melody and accompaniment, the rule-based baseline tries to make their pitch class histogram similar and put their notes on the same onsets. At the same time, the revisions should be minor to preserve the identity of the inputs. The rule-based baseline only changes the input melody. For every note of the melody, we find the closest pitch class of the accompaniment notes. If the pitch distance is below the threshold of one semitone, we change the melody pitch to that pitch class. For example, if the pitch classes of the accompaniment notes are { C, E, F}, and a note of the melody is C#4. The closest pitch class is C, and the distance is below the pitch threshold of one semitone, then we change the C#4 to C4. As for rhythm, we move the notes of the melody to the same onsets of the accompaniment if the time distance is under the onset threshold of two steps. Besides, there is a 20% chance that a note retains even if it is changeable.

4.3 Evaluation of Harmony

We aim to fuse disharmonic pairs of melodies and accompaniments into harmonic pairs. In this subsection, we evaluate the level of harmony of the outputs of CollageNet and

	harmony rate	$\rho(\hat{z}_{mel}, z_{mel})$	$\rho(\hat{z}_{acc}, z_{acc})$
Ω_{dh}	10.10%	-	-
Data-driven baseline	61.70%	1.12	1.36
Rule-based baseline	67.27%	1.28	-
CollageNet-vanilla	92.59%	0.98	1.75
CollageNet ($\lambda = 0.1$)	**92.71%**	0.92	1.89
CollageNet ($\lambda = 0.5$)	89.58%	0.69	1.56
CollageNet ($\lambda = 1.0$)	89.56%	0.66	1.35
CollageNet ($\lambda = 2.0$)	84.58%	**0.52**	**1.27**

Table 1. Harmony rates of the disharmonic test set Ω_{dh}, and output from CollageNet (with different distance penalty λ) and two baseline methods, which take data from Ω_{dh} as inputs. Besides, the average latent space distances of melodies and accompaniments between the outputs $\hat{z}$ and inputs z of the methods are also reported.

baseline methods. It is hard to design exhaustive metrics to evaluate the level of harmony. We use both the deep-learning model and musical statistics to evaluate the level of harmony.

We train a deep-learning evaluation model to discriminate between harmonic pairs and disharmonic pairs. The evaluation model uses a PianoTree encoder [36] to encode the polyphonic accompaniments, and a bidirectional GRU to encode the monophonic melodies. Then the encoded vectors are concatenated and passed through a multilayer perceptron (MLP) to produce a score between 0 and 1 for each pair of samples. The binary cross-entropy loss is used to train the evaluation model with the data from Ω_h and Ω_{dh}. After training, the accuracy is about 90% in the test set. We define *harmony rate* as the proportion of samples identified as positive by the evaluation model.

The harmony rates of the four methods are displayed in Table 1. The latent space distances $\rho(\hat{z}, z)$ between outputs $\hat{z}$ and inputs z of the methods are also displayed. We report the results of CollageNet with different distance penalty λ. In addition to CollageNet and two baselines, we also evaluate Collagenet-vanilla, which utilizes vanilla VAEs [36] instead of disentangled VAEs. According to the results, CollageNet can produce music with higher harmony rates while making fewer changes to the inputs. Besides, with a higher distance penalty, the performance of CollageNet degrades slightly, and the latent space distances reduce. It is noted that CollageNet and CollageNet-vanilla achieve comparable harmony rates, but the disentangled VAEs in CollageNet provides user control in the fusion process as described in Section 3.3 and validated in Section 4.5.

Although the deep-learning model evaluates more comprehensively, it is agnostic. Inspired by [37, 38], we adopt several musical statistics to evaluate the level of harmony of each pair of melody and accompaniment. Firstly, we extract several features from both melodies and accompaniments. **PCH** is the pitch class histogram with 12 bins. **OH** is the onset histogram with 32 bins corresponding to 32 time steps. **RE** is the rhythm pattern, a 32-dimensional vector denoting states of every time step, including onsets, holding states of any pitch, and rests. The **PCH** feature reveals the pitch pattern of melodies and accompaniments, while **OH** and **RE** reveal the rhythm pattern. For **PCH** and **OH**, we calculate the Kullback-Leibler Divergence (KLD) and Overlapping Area (OA) between the melody and accompaniment of each pair. For **RE**, we calculate the ratio of the same pattern between the melody and accompaniment. The average values of these statistics are in Table 2. According to the results, the outputs of the three methods get closer to the harmonic test set Ω_h than inputs from Ω_{dh}. CollageNet is significantly better in most metrics. The rule-based baseline is better than the data-driven baseline because it directly optimizes these metrics.

	PCH		**OH**		**RE**↑
	KLD↓	OA↑	KLD↓	OA↑	
Ω_h	0.96	0.578	2.171	0.471	0.643
Ω_{dh}	5.02	0.292	4.522	0.299	0.457
Data-driven baseline	2.69	0.397	3.421	0.377	0.531
Rule-based baseline	1.91	0.459	**1.831**	0.464	**0.662**
CollageNet	**1.38**	**0.588**	2.228	**0.476**	0.612

Table 2. The musical statistics averaged over datasets for harmony evaluation. The Ω_h is the harmonic test set. Two baseline methods and CollageNet take data from the disharmonic test set Ω_{dh} as inputs. The arrows indicate a better direction.

4.4 Evaluation of Music Quality

To fuse a pair of unrelated melody and accompaniment, CollageNet and baseline methods change the inputs. The fusion process may destroy the musicality of each of the input tracks. Thus, we compare several musical statistics between created datasets and the original dataset to evaluate the quality of transformed melodies and accompaniments respectively. The datasets whose statistics are closer to the test set are more similar to human-made music [37]. Different from Section 4.3 where the comparison is between each pair of melody and accompaniment, the comparison happens between two datasets in this subsection.

We calculate **PC** (pitch count), **PI** (pitch interval), and **IOI** (inter-onset-interval) as in [37] from melodies and accompaniments respectively. Table 3 shows the results. Except for CollageNet and baseline methods, we also calculate the statistics of music generated by VAEs. The VAEs generate music by sampling latent vectors from the prior $p(z)$ and decoding them to musical notes. According to the results, although the rule-based baseline outperforms the data-driven baseline in Table 1 and Table 2, its outputs are very different from the real data. As for CollageNet, the musicality of melodies and accompaniments does not deteriorate after the transformation. The musical statistics of CollageNet are closest to the test set.

4.5 Subjective Experiment

We implement subjective experiments to evaluate CollageNet and the rule-based baseline methods. Before the test, we ask the subjects three questions following [18]:

Do you master any musical instruments? Have you received vocal training before? Have you learned music theory systematically before?

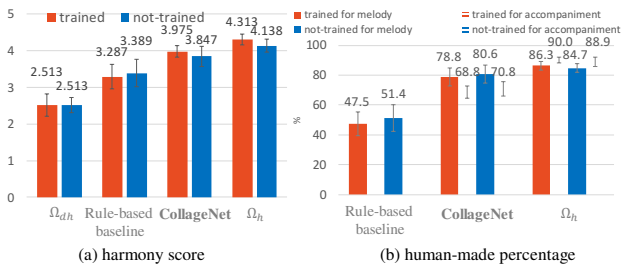
	melody			accompaniment	
	PC	**PI**	**IOI**	**PC**	**IOI**
test set	10.7	0.024	3.1	31.6	2.02
VAE	-0.30	+0.054	+0.05	-3.7	**+0.00**
Data-driven baseline	-0.99	+0.099	+0.33	-2.3	+0.14
Rule-based baseline	-1.33	+0.102	+0.33	-	-
CollageNet	**-0.10**	**+0.010**	**+0.03**	**-1.2**	+0.01

Table 3. The average musical statistics of melodies and accompaniments in the test set Ω_h. For four methods, we report the difference of their outputs from the test set. The rule-based baseline does not alter accompaniments.

We denote the subjects who answer *yes* to any of these questions as *trained*, other subjects as *not-trained*. We invite 38 people to complete the survey. Among them, 20 people are trained and 18 people are not-trained.

Each subject listens to 16 randomly shuffled and anonymous music pieces, comprising of 4 pieces from the disharmonic pair set Ω_{dh}, 4 pieces from outputs of the rule-based baseline, 4 pieces from outputs of CollageNet, and 4 pieces from the harmonic pair set Ω_h. Therefore, 142 pieces of each kind of data are evaluated in total. The music pieces are rendered using violin for melodies and piano for accompaniments. The subjects rate the harmony of these pieces on a 5-point scale where "1" indicates "disharmonic" and "5" indicates "harmonic". They are told to concentrate on the coherence of melodies and accompaniments. Figure 3 (a) illustrates the average harmony score. The outputs of CollageNet are rated as more harmonic than the rule-based baseline.

Then each subject is asked to listen to four pieces of melodies from outputs of the rule-based baseline, outputs of CollageNet, and Ω_h each; four pieces of accompaniments from outputs of CollageNet and Ω_h each (the rule-based baseline does not revise accompaniments). They judge whether each piece is composed by humans or generated by machine. Figure 3 (b) illustrates the average percentage of pieces rated as human-made by the subjects. Although the rule-based baseline can fuse the disharmonic inputs, nearly half of the output melodies of the rule-based baseline are regarded as machine-made. CollageNet produces both harmonic and high-quality music. Such observation is consistent with the conclusions of Section 4.4.



(a) harmony score (b) human-made percentage

Figure 3. Average harmony score for two-track segments from four datasets are displayed. And human-made percentage are calculated for melodies and accompaniments respectively, which are from three datasets. The scores of trained subjects and not-trained subjects are displayed.

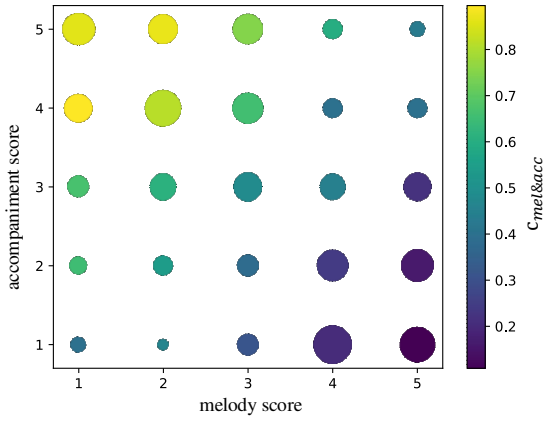


Figure 4. All similarity scores on a 5-point scale given by subjects for melodies and accompaniments. The melodies and accompaniments are produced by CollageNet with different user control $c_{mel\&acc}$. Each circle represents all the songs with the specific melody score and accompaniment score. The size of the circles indicates the number of the songs, and the color of the circles indicates the average $c_{mel\&acc}$ of the songs.

To demonstrate the validity of CollageNet's user control, the subjects listen to the output melodies and accompaniments respectively of CollageNet with sliding user control inputs. And the subjects rate the similarity of each output melody and accompaniment to the inputs. We define the term $c_{mel\&acc} = c_{mp} = c_{mr} = 1 - c_{ac} = 1 - c_{at}$. Each subject rates two groups of songs, with each group consists of six melodies and accompaniments produced with different $c_{mel\&acc}$ sliding from 0 to 1. The scores are on a 5-point scale where "1" indicates "similar" and "5" indicates "different". Figure 4 displays all the scores. As the $c_{mel\&acc}$ increases, the output melodies are considered more similar to the input, while the output accompaniments are the opposite.

5. CONCLUSION

In this paper, we presented a new task and a neural approach on multi-track music fusion, which is similar to the music sampling practice. Specifically, given an unrelated pair of melody and accompaniment of the same length, the proposed approach fuses them to produce harmonic two-track music while maintaining their musical identity. Besides, users can control the magnitude of changes along disentangled musical aspects. We conducted objective and subjective experiments and compared the proposed approach with rule-based and data-driven baseline methods. Experimental results showed that the proposed method achieved significantly higher level of harmony than that of baselines, with musically high-quality outputs.

For future work, CollageNet can be extended to arbitrary tracks and longer music pieces. Besides, similar ideas and systems can be explored in the audio domain.

6. ACKNOWLEDGEMENT

This work is funded by the National Key Research and Development Program of China (No. 2018AAA0100701) and the National Science Foundation grant No. 1846184 of the USA.

7. REFERENCES

[1] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. C. Courville, and D. Eck, "Counterpoint by convolution." in *Proceedings of the 18th International Society for Music Information Retrieval Conference*. Suzhou, China: ISMIR, Oct. 2017, pp. 211–218.

[2] C. Donahue, H. H. Mao, Y. E. Li, G. Cottrell, and J. McAuley, "LakhNES: Improving multi-instrumental music generation with cross-domain pre-training," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*. Delft, The Netherlands: ISMIR, Nov. 2019, pp. 685–692.

[3] N. Jiang, S. Jin, Z. Duan, and C. Zhang, "Rl-duet: Online music accompaniment generation using deep reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 710–718.

[4] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020.

[5] H. Zhu, Q. Liu, N. J. Yuan, C. Qin, J. Li, K. Zhang, G. Zhou, F. Wei, Y. Xu, and E. Chen, "Xiaoice band: A melody and arrangement generation framework for pop music," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2837–2846.

[6] H. Chu, R. Urtasun, and S. Fidler, "Song from pi: A musically plausible network for pop music generation," *arXiv e-prints*, pp. arXiv–1611, 2016.

[7] B. J. McCann, *The mark of criminality: Rhetoric, race, and gangsta rap in the war-on-crime era*. University of Alabama Press, 2017.

[8] N. Patrin, *Bring That Beat Back: How Sampling Built Hip-Hop*. U of Minnesota Press, 2020.

[9] S. Howell, "The lost art of sampling: Part 4," *Sound on Sound Magazine*, 2005.

[10] Wikipedia contributors, "Sampling (music) — Wikipedia, the free encyclopedia," 2021, [Online; accessed 26-April-2021]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Sampling_(music)&oldid=1013854871

[11] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[12] R. Yang, D. Wang, Z. Wang, T. Chen, J. Jiang, and G. Xia, "Deep music analogy via latent representation disentanglement," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*. Delft, The Netherlands: ISMIR, Nov. 2019, pp. 596–603.

[13] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020.

[14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, p. 2672–2680.

[15] J. Engel, M. Hoffman, and A. Roberts, "Latent constraints: Learning to generate conditionally from unconditional generative models," *arXiv preprint arXiv:1711.05772*, 2017.

[16] M. Allan and C. K. Williams, "Harmonising chorales by probabilistic inference," *Advances in Neural Information Processing Systems*, vol. 17, pp. 25–32, 2005.

[17] C. Benetatos, J. VanderStel, and Z. Duan, "Bachduet: A deep learning system for human-machine counterpoint improvisation," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2020, pp. 635–640.

[18] N. Jiang, S. Jin, Z. Duan, and C. Zhang, "When counterpoint meets chinese folk melodies," in *NeurIPS*, 2020.

[19] F. T. Liang, M. Gotham, M. Johnson, and J. Shotton, "Automatic stylistic composition of bach chorales with deep lstm." in *ISMIR*, 2017, pp. 449–456.

[20] G. Hadjeres, F. Pachet, and F. Nielsen, "Deepbach: a steerable model for bach chorales generation," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1362–1371.

[21] Y. Yan, E. Lustig, J. VanderStel, and Z. Duan, "Part-invariant model for music generation and harmonization," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*. Paris, France: ISMIR, Sep. 2018, pp. 204–210.

[22] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[23] I. Simon, A. Roberts, C. Raffel, J. Engel, C. Hawthorne, and D. Eck, "Learning a latent space of multitrack measures," *arXiv preprint arXiv:1806.00195*, 2018.

[24] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, pp. 3483–3491, 2015.

[25] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *arXiv preprint arXiv:1906.02691*, 2019.

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

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

[28] 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.

[29] H. H. Tan and D. Herremans, "Music fadernets: Controllable music generation based on high-level features via low-level feature modelling," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020.

[30] T. Akama, "Connective fusion: Learning transformational joining of sequences with application to melody creation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020.

[31] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola, "Style transfer from non-parallel text by cross-alignment," *arXiv preprint arXiv:1705.09655*, 2017.

[32] J. Mueller, D. Gifford, and T. Jaakkola, "Sequence to better sequence: continuous revision of combinatorial structures," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2536–2544.

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

[34] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, X. Gu, and G. Xia, "Pop909: A pop-song dataset for music arrangement generation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020.

[35] S. Porter, *The Harmonization of the Chorale: A Comprehensive Workbook Course in Harmony and Counterpoint*. Taylor & Francis, 1987.

[36] Z. Wang, Y. Zhang, Y. Zhang, J. Jiang, R. Yang, J. Zhao, and G. Xia, "Pianotree vae: Structured representation learning for polyphonic music," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020.

[37] L.-C. Yang and A. Lerch, "On the evaluation of generative models in music," *Neural Computing and Applications*, vol. 32, no. 9, pp. 4773–4784, 2020.

[38] Y.-C. Yeh, W.-Y. Hsiao, S. Fukayama, T. Kitahara, B. Genchel, H.-M. Liu, H.-W. Dong, Y. Chen, T. Leong, and Y.-H. Yang, "Automatic melody harmonization with triad chords: A comparative study," *Journal of New Music Research*, vol. 50, no. 1, pp. 37–51, 2021.