Q&A: Query-Based Representation Learning for Multi-Track Symbolic Music re-Arrangement

Jingwei Zhao[2,3] , **Gus Xia**[4,5] and **Ye Wang**[1,2,3]
[1]School of Computing, NUS
[2]Institute of Data Science, NUS
[3]Integrative Sciences and Engineering Programme, NUS Graduate School
[4]Music X Lab, NYU Shanghai
[5]MBZUAI
jzhao@u.nus.edu, gxia@nyu.edu, wangye@comp.nus.edu.sg

Abstract

Music rearrangement is a common music practice of reconstructing and reconceptualizing a piece using new composition or instrumentation styles, which is also an important task of automatic music generation. Existing studies typically model the mapping from a source piece to a target piece via supervised learning. In this paper, we tackle rearrangement problems via self-supervised learning, in which the mapping styles can be regarded as conditions and controlled in a flexible way. Specifically, we are inspired by the representation disentanglement idea and propose Q&A, a query-based algorithm for multi-track music rearrangement under an encoder-decoder framework. Q&A learns both a *content* representation from the mixture and function (*style*) representations from each individual track, while the latter queries the former in order to rearrange a new piece. Our current model focuses on popular music and provides a controllable pathway to four scenarios: 1) re-instrumentation, 2) piano cover generation, 3) orchestration, and 4) voice separation. Experiments show that our query system achieves high-quality rearrangement results with delicate multi-track structures, significantly outperforming the baselines.

1 Introduction

It is sometimes easy to craft an idea of the melody but usually hard to frame a good arrangement. Formally, *arrangement* refers to the form of a musical piece, typically with textures and voicing carefully designed for multiple instruments as a unique style. On top of that, a piece can also be rearranged to convey new feelings. Such *rearrangement* scenarios include piano cover generation from multi-track music, multi-track orchestration from piano, and re-instrumentation using varied instruments, which are all common tasks in music practice.

While much progress has been witnessed in automatic music generation [Huang *et al.*, 2019; Huang and Yang, 2020; Hsiao *et al.*, 2021], rearrangement remains a challenging problem. Among various ways to rearrange a piece, most studies have focused on the reduction from complex forms to simpler ones, such as generating piano covers from multi-track music. The reduction is typically done by masking least significant notes, either identified by rule-based criteria [Nakamura and Yoshii, 2018] or learned by supervision [Terao *et al.*, 2022]. While rearrangement in this way is generally faithful, it tends to produce sparse or repetitive textures that fall short of creativity. More recent works have also taken on simple-to-complex rearrangement, such as orchestration [Crestel and Esling, 2017; Dong *et al.*, 2021]. While these works are still fully supervised, the scarce of paired piano and multi-track data remains a major problem in this direction.

Another considerable challenge for music rearrangement lies in *multi-track* modelling. Previous works have typically interpreted “track” as “instrument” and merge individual tracks of the same instrument class to simplify the problem [Dong *et al.*, 2018; Ren *et al.*, 2020]. However, instrument alone is not necessarily a good representative of a multi-track system in symbolic music. For example, pop music often has two guitar tracks of quite different functions – a *melodic* one and a *harmonic* one. When merged together, the distinctive texture structures of each track become less transparent, which may add extra burden to the model.

In this paper, we aim to approach multi-track music rearrangement while balancing faithfulness with creativity. We render the content of a source piece using the style from a reference piece that is free to choose. In terms of the “style” of a multi-track piece, apart from instruments, we believe the *function* of each component track is also important. Specifically, we consider the function of a track as its texture density distribution along the time- and pitch-axes, respectively, which can describe both the distinctive intra-track structures (*e.g.*, melodic *v.s.* harmonic) and the inter-track dependencies (*e.g.*, pitch range and voicing). We use track functions as queries in a *query-based track separation process* to reconstruct individual tracks from a track-wise condensed mixture. Under the VAE framework [Wang *et al.*, 2020c], we devise a pipeline consisting of four components: 1) an encoder that maps a mixture to the latent space; 2) a query-net [Lee *et al.*, 2019] that encodes function features of each track; 3) a Transformer-based [Vaswani *et al.*, 2017] query system that separates each track from the mixture at the latent represen-

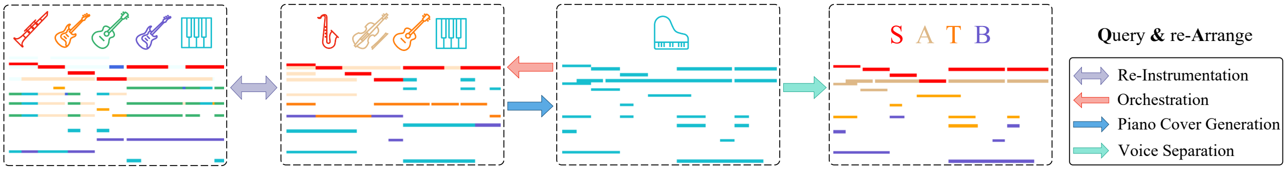

Figure 1: Q&A is a unified framework for re-instrumentation, orchestration, piano cover generation, and voice separation.

tation level; and 4) a decoder that reconstructs each separated track. At inference time, our model can query a piece and rearrange it with diverse track functions.

We name our model after *Q&A* (Query & re-Arrange), which provides a unified solution to a range of multi-track music rearrangement tasks, including: 1) *re-instrumentation* – to rearrange a multi-track piece with a new track system; 2) *piano cover generation* – to rearrange a multi-track piece into piano solo; and 3) *orchestration* – to rearrange a piano piece with a variable types of instruments in a variable number of tracks. By inferring track functions as voice hints, our model can additionally handle 4) *voice separation* – to separate distinctive voicing tracks (assuming a preset total number of voices) from an ensemble mixture by generating each track. Figure 1 shows the relations among the four tasks.

Our current model focuses on pop music rearrangement. We also test our model's voice separation performance on string quartets and Bach chorales. Experimental results show that our model not only generates high-quality arrangements, but also maintains fine-grained symbolic track structures with musically intuitive and playable textures for each track. In summary, our contributions in this paper are as follows:

- **A versatile rearrangement model**: We present *Q&A*[1], the first unified framework for re-instrumentation, piano cover generation, orchestration, and voice separation. The rearrangement results demonstrate state-of-the-art quality over existing models for similar purposes.
- **Function-aware multi-track music modelling**: We design instrument-agnostic track functions for multi-track modelling, which can better describe the distinctions of parallel tracks and their dependencies. This method is applicable to a wider range of music generation tasks.
- **Query-based representation learning**: We introduce a self-supervised query system separating parts from the mixture at latent representation level. Experiments show that our model learns style representations of each part disentangled from the mixture content, demonstrating interpretable and controllable generative modelling.

2 Related Work

2.1 Symbolic Music Rearrangement

Existing studies on music rearrangement commonly rely on supervised learning to map a source piece to a target one. For example, Crestel and Esling [2017] project piano solo to orchestra by training a seq2seq model on a classical repertoire of paired data [Crestel *et al.*, 2017]. Dong *et al.* [2021] approach automatic instrumentation by predicting the instrument attribute of each note in a track-wise condensed mixture. Models for piano reduction can have more rule-based designs [Nakamura and Sagayama, 2015; Takamori *et al.*, 2017], but are still generally under supervised frameworks. Except for several works that consider difficulty level as condition [Nakamura and Yoshii, 2018; Terao *et al.*, 2022], most models cannot steer the rearrangement process or change the composition style.

In this paper, instead of supervised mapping, we render the content of a source piece using the style from a reference piece. In terms of content, we preserve the general melodic and harmonic structures. As for style, we introduce a new track system, *i.e.*, textural functions of each track along with the instruments to play them, to reconceptualize the source piece. Our methodology can be formalized as composition style transfer [Dai *et al.*, 2018] while existing research most relevant to us is [Hung *et al.*, 2019], which approaches re-instrumentation by transferring instrument timbres from different references. While Hung *et al.* [2019] still require supervision from audio-symbolic pairs, our model is fully symbolic-based, self-supervised, and unified for re-instrumentation, piano cover generation, and orchestration.

2.2 Multi-Track Music Modelling

Multi-track music is an arrangement form commonly seen in accompaniment, symphony, ensembles, *etc.* However, it is very challenging for machines to understand multi-track data. To capture inter-track dependency, mainstream approaches either distribute vari-instrument tracks into parallel data channels [Dong *et al.*, 2018; Zhu *et al.*, 2018; Hung *et al.*, 2019] or incorporate instrument labels as part of note event tokens [Donahue *et al.*, 2019; Payne, 2019; Ren *et al.*, 2020]. Such methods are ideal for CNN-based and language models, respectively, yet both inevitably merging co-instrument tracks together. The event-based approach additionally enforces a positional relation to parallel tracks that are not sequentially ordered, which can damage the intrinsic structure of multi-track music [Wang and Xia, 2021]. More recently, several works target at these issues and support generating co-instrument tracks without a sequential assumption [Ens and Pasquier, 2020; Liu *et al.*, 2022]. However, these models are not applicable to general music rearrangement.

In this work, apart from instrument, we introduce track function as an equally (if not more) important feature to describe and distinguish individual tracks in multi-track music. We define and use the function of a track to represent its texture and voicing structures. We further model multi-track music via self-supervised learning, *i.e.*, using each function to query and separate corresponding tracks from a mixture.

[1] https://github.com/zhaojw1998/Query-and-reArrange

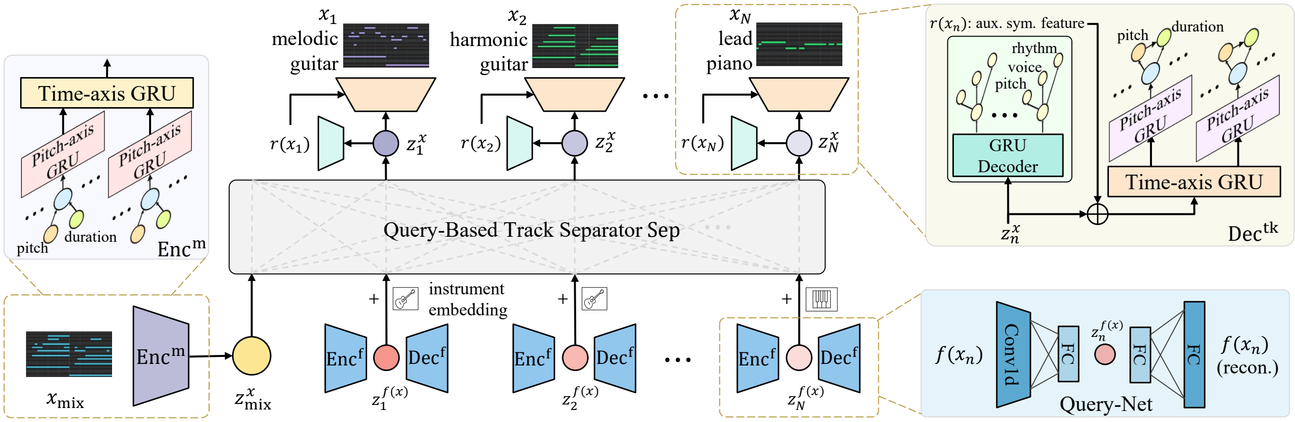

Figure 2: The architecture of Q&A consists of four key components: mixture encoder Enc^{m}, function query-net with encoder Enc^{f} and decoder Dec^{f}, track separator Sep, and track decoder Dec^{tk}. Q&A learns both a content representation from the mixture and function representations from each individual track, while the latter queries the former in order to rearrange a new piece.

3 Methodology

We propose Q&A, a query-based algorithm for multi-track music rearrangement under an encoder-decoder framework. An overview of our model is shown in Figure 2. In this section, we first introduce our data representation of multi-track music and track functions in Section 3.1. Then, we introduce our model architecture and training objectives in Section 3.2 and 3.3. Finally, in Section 3.4, we elaborate on how Q&A can be applied to music rearrangement at inference time.

3.1 Data Representation

Multi-Track Music

Given multi-track music x with N tracks, our model aims to reconstruct each track x_n, where $n = 1, 2, \cdots, N$, from a track-wise condensed mixture x_{mix}. We represent x_n in the modified piano-roll format proposed by [Wang *et al.*, 2020b], and x as an N-track collection. Formally,

$$x = x_{1:N} \coloneqq \{x_n\}_{n=1}^{N}, \tag{1}$$

where individual track x_n is a $P \times T$ matrix. $P = 128$ represents 128 MIDI pitches, and T is the time dimension. Each data entry (p, t) of x_n is an integer value representing note duration on the onset positions. The condensed mixture x_{mix} is also a $P \times T$ matrix where each entry is the position-wise maximum value across N tracks. In this paper, we consider 2-bar (8-beat) music data segments in $\frac{4}{4}$ time signature quantized at $\frac{1}{4}$ beat unit, deriving $T = 32$ time steps for each music sample. We also focus on composition-level aspects while disregarding performance-level dynamics like MIDI velocity.

Track Function

We define the function of each track x_n as its texture density features computed from the modified piano-roll format. Specifically, we define descriptors of pitch function $f^{\text{p}}(\cdot)$ and time function $f^{\text{t}}(\cdot)$ as follows:

$$f^{\text{p}}(x_n) = \text{rowsum}(\mathbf{1}_{\{x_n>0\}})/T, \tag{2}$$

$$f^{\text{t}}(x_n) = \text{colsum}(\mathbf{1}_{\{x_n>0\}})/P, \tag{3}$$

where $\mathbf{1}_{\{\cdot\}}$ is the indicator function expressing individual note onset entries as 1. $\text{rowsum}(\cdot)$ and $\text{colsum}(\cdot)$ each sums up one dimension, resulting in a P-D and T-D vector, respectively. $f^{\text{p}}(x_n)$ is essentially a pitch histogram, which is related to key, chord, and the pitch range of x_n. $f^{\text{t}}(x_n)$ indicates voice densities of x_n and is related to rhythmic patterns and grooves. Each vector is normalized to $[0, 1]$.

3.2 Model Architecture

Figure 2 shows the overall architecture of our model consisting of four key components: 1) a mixture encoder, 2) a function query-net, 3) a track separator, and 4) a track decoder.

Mixture Encoder

As x_{mix} is a single-track polyphony, we use the encoder module of PianoTree VAE [Wang *et al.*, 2020c], the state-of-the-art polyphonic representation learning model, to encode a 256-D mixture representation z_{mix}^x. The PianoTree encoder first converts x_{mix} to a compact and ordered note event format, where each event contains pitch and duration attributes. It then applies a pitch-wise bi-directional GRU to summarize concurrent notes at time step t to an intermediate representation simu_note_t. On top of $\text{simu\_note}_{1:T}$, it further applies a time-wise GRU to encode the full mixture representation z_{mix}^x. The encoding process of PianoTree VAE reflects hierarchical musical understanding from note via chord to grouping, which is interpretable and has proved beneficial for a range of downstream generation tasks [Yi *et al.*, 2022; Wuerkaixi *et al.*, 2021; Zhao *et al.*, 2022; Wang *et al.*, 2022].

Function Query-Net

The function query-net consists of two VAEs that encode 128-D representations $z_n^{\text{p}(x)}$ and $z_n^{\text{t}(x)}$ for track functions $f^{\text{p}}(x_n)$ and $f^{\text{t}}(x_n)$, respectively. The pitch and time function encoders each consist of a 1-D convolutional layer with kernel size 12 and 4, respectively. Both are followed by ReLU activation [Nair and Hinton, 2010] and 1-D max-pooling with kernel size 4 and stride 4. The decoders consist of two fully-connected layers with ReLU activation in between.

It is noted that, with the encoder design, we leverage the translation invariance property of convolution and the blurry effect of pooling [Krizhevsky *et al.*, 2017] to discourage the separator from simply retrieving notes that are implied in the track functions. By doing so, our model learns a general style representation instead of the exact density values from the track function. Similar method is also adopted in other VAE architectures to realize disentanglement [Wang *et al.*, 2020b].

Track Separator

The track separator is a 2-layer Transformer encoder with 8 attention heads, 0.1 dropout ratio, and GELU activation [Hendrycks and Gimpel, 2016]. The hidden dimensions of self-attention d_{model} and feed-forward layers d_{ff} are 512 and 1024, respectively. The input to the separator is a sequence of $N+1$ latent codes including mixture z^x_{mix} and track functions $z^{f(x)}_{1:N}$, where $z^{f(x)}_n$ denotes the concatenation $[z^{\text{p}(x)}_n; z^{\text{t}(x)}_n]$ as a unified track function representation. We also add learnable instrument embeddings to the corresponding tracks. It is noted that Transformer is permutation-invariant to the index of track functions so that no sequential assumption is enforced. While the self-attention mechanism allows each track function as query to attend to the mixture, it also encourages queries to attend to each other for inter-track dependency. We denote the output of the Transformer as $z^x_{1:N}$, which are the expected latent representations for individual tracks $x_{1:N}$.

Track Decoder

We use the decoder module of PianoTree VAE to reconstruct each track x_n from representation z^x_n. The decoder involves time- and pitch-wise uni-directional GRUs, which mirror the structure of the encoder. To better distinguish parallel tracks, we additionally provide the decoder with an auxiliary time sequence of symbolic features, which are priorly predicted from z^x_n. Specifically, we consider three auxiliary features: *pitch centre*, *voice intensity*, and *rhythm*, which can serve as strong hints to determine if one track has *melodic*, *harmonic*, and *static* properties [Couturier *et al.*, 2022a; Couturier *et al.*, 2022b]. Both pitch centre and voice intensity are time sequences of scalar values, which indicate centre pitch curve and voice number progression of a track, both normalized to $[0, 1]$. The rhythm feature is a time sequence of onset probabilities, which represents the rhythmic pattern in time. We use a uni-directional GRU to predict the symbolic features from z^x_n and feed them to the corresponding time steps of the time-wise GRU in the PianoTree Decoder. Similar method is also applied for disentanglement and reconstruction in [Yang *et al.*, 2019; Wang *et al.*, 2022].

3.3 Training Objectives

The loss terms in our model include 1) reconstruction loss for each track, track functions, and auxiliary symbolic features, and 2) KL loss between all latent representations and standard normal distribution. Our model is essentially a variational autoencoder since the loss function can be formalized as the evidence lower bound (ELBO) of distribution $p(x)$, where $x = x_{1:N}$ is the multi-track music.

The posterior distribution of the VAE is defined as the product of three modules including mixture encoder, query-net encoder, and track separator:

$$q_{\boldsymbol{\phi}}(\mathbf{z} \mid x) := q_{\phi_1}(z^x_{\text{mix}} \mid x_{\text{mix}}) \prod_{n=1}^{N} q_{\phi_2}(z^{f(x)}_n \mid f(x_n)) \prod_{n=1}^{N} q_{\phi_3}(z^x_n \mid z^x_{\text{mix}}, z^{f(x)}_{1:N}), \tag{4}$$

where $\boldsymbol{\phi} := [\phi_1, \phi_2, \phi_3]$ denotes the parameters of the three modules, and $\mathbf{z} := [z^x_{\text{mix}}, z^x_{1:N}, z^{f(x)}_{1:N}]$. In Equation (4), we collectively express two types of track functions as $f(x_n)$ for conciseness. It is noted that both x_{mix} and $f(x_n)$ are deterministically transformed from x and hence are not explicitly written in the left-hand side of Equation (4).

The reconstruction distribution is defined as the product of three reconstruction terms of query-net decoder, symbolic feature decoder, and track decoder:

$$p_{\boldsymbol{\theta}}(x \mid \mathbf{z}) := \prod_{n=1}^{N} p_{\theta_1}(f(x_n) \mid z^{f(x)}_n) \prod_{n=1}^{N} p_{\theta_2}(r(x_n) \mid z^x_n) \prod_{n=1}^{N} p_{\theta_3}(x_n \mid z^x_n, r(x_n)), \tag{5}$$

where $\boldsymbol{\theta} := [\theta_1, \theta_2, \theta_3]$ denotes the parameters of the three decoders. $r(x_n)$ denotes the auxiliary symbolic features for track x_n. The p_{θ_1} term can be interpreted as a regularizer to the overall output distribution $p_{\boldsymbol{\theta}}(x \mid \mathbf{z})$.

Finally, the overall loss function is as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; x) = & -\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}} \log p_{\boldsymbol{\theta}}(x \mid \mathbf{z}) \\ & + \beta\, \mathbb{KL}(q_{\boldsymbol{\phi}}(\mathbf{z} \mid x) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{1})),\end{aligned} \tag{6}$$

where β is a balancing parameter [Higgins *et al.*, 2017].

3.4 Style Transfer

At inference time, Q&A can rearrange a multi-track source piece $x = x_{1:N}$ using the track system (style) from a reference piece $y = y_{1:M}$, which can be freely selected. Let Enc^{m}, Enc^{f}, Sep, and Dec^{tk} be the mixture encoder, function encoder, track separator, and track decoder, respectively, the rearrangement process takes a pipeline as follows:

$$\begin{aligned} z^x_{\text{mix}} &= \text{Enc}^{\text{m}}(x_{\text{mix}}), \\ z^{f(y)}_m &= \text{Enc}^{\text{f}}(f(y_m)),\ m = 1, 2, \cdots, M, \\ z^{x'}_{1:M} &= \text{Sep}(z^x_{\text{mix}},\ z^{f(y)}_{1:M}), \\ x'_m &= \text{Dec}^{\text{tk}}(z^{x'}_m),\ m = 1, 2, \cdots, M, \end{aligned} \tag{7}$$

where $x' = x'_{1:M}$ is the rearrangement result. x' inherits the general harmonic structures from x, while also introducing y's track system with new textures, grooves, and track voicing played by a different set of instruments.

In addition to manual selection, reference y can be automatically searched from a database $\mathcal{D}$. To guarantee faithful and natural rearrangement results, we develop a simple heuristic to sample y that is "matched" with x as follows:

$$y = \underset{y\in\mathcal{D}}{\text{argmax}}[\cos(f(y_{\text{mix}}), f(x_{\text{mix}})) + \alpha \cdot \epsilon_y], \tag{8}$$

where $\cos(\cdot,\,\cdot)$ measures cosine similarity between the functions (essentially, texture densities) of mixture y_{mix} and x_{mix}, $\epsilon_y \sim \mathcal{N}(0,\,1)$ is a noise term for balancing with generality, and α is a balancing parameter. In cases when y and x are very dissimilar, our model robustly follows x's harmony and y's texture and voicing in a general sense of style transfer.

4 Experiments

4.1 Dataset

Our model is trained on Slakh2100 [Manilow *et al.*, 2019] and POP909 [Wang *et al.*, 2020a] datasets. In specific, Slakh2100 contains 2K MIDI files of multi-track music, most of which are in pop style. Instruments in Slakh2100 are categorized into 34 classes (while co-instrument tracks are not merged) and each piece contains at least one track of piano, guitar, bass, and drum. In our experiment, we discard the drum track because it does not follow the standard 128-pitch protocol used in other tracks. POP909 is a dataset of 1K pop songs in piano arrangement created by professional musicians. Each piece consists of three piano tracks for vocal melody, lead instrument melody, and piano accompaniment, respectively. By jointly training our model on both datasets, our model can rearrange multi-track music to piano and vice versa.

4.2 Training

For training Q&A, we use the official training split of Slakh2100 while randomly splitting POP909 (at song level) into training, validation, and test sets with a ratio of 8:1:1. We further augment training data by transposing each piece to all 12 keys. Our model comprises 19M learnable parameters and is trained with a mini-batch of 64 2-bar segments for 30 epochs on an RTX A5000 GPU with 24GB memory. We use Adam optimizer [Kingma and Ba, 2014] with a learning rate from 1e-3 exponentially decayed to 1e-5. We apply teacher forcing [Toomarian and Barhen, 1992] for the decoder GRUs in PianoTree VAE with a rate from 0.8 to 0. For the parameter β in Equation (6), we apply KL annealing following [Wang *et al.*, 2022] and set β increasing from 0 to 0.5 for $z_n^{f(x)}$ and from 0 to 0.01 for the other two factors.

4.3 Rearrangement Showcase

An 8-bar rearrangement example (by processing every 2 bars independently) is shown in Figure 3. In specific, this is an orchestration example, where we use Q&A to rearrange a piano piece into multi-track music. The piano source x is from POP909 while we sample reference y of the same length from Slakh2100 following Equation (8) with $\alpha = 0.2$. We add $f(x_{\mathrm{mel}})$, the function of x's melody track, to y's track functions as an additional query to guarantee the preservation of the theme melody. Meanwhile, we conduct posterior sampling over $z_{\mathrm{mel}}^{x'}$ to encourage melody improvisation.

In this example, our model rearranges the piano piece into 11 tracks with coherent and delicate multi-track textures. Among the 11 tracks, guitar and organ are each used twice for melodic and harmonic purposes, respectively. Our model preserves the original harmony quite faithfully. Particularly, it captures the added chord notes in x (highlighted by red dotted lines in Figure 3) and retains the tension from the original piece. At the same time, it introduces new groove patterns, bass lines, lead instrument melodies, and a theme melody variation to reconceptualize the piece with more creativity.

4.4 Subjective Evaluation on Rearrangement

Based on composition style transfer, Q&A is a unified solution for a range of music rearrangement tasks. In this paper, we focus on orchestration, piano cover generation, and reinstrumentation. For evaluation, we introduce three existing models as baselines for each of the three tasks as follows:

BL-Orch.: We introduce *Arranger* by [Dong *et al.*, 2021] as our baseline for the orchestration task. We select the BiLSTM variant pre-trained on Lakh MIDI dataset [Raffel, 2016], which is a superset of Slakh2100 that we use. Orchestration by Arranger is a note-by-note mapping process, where each note in the piano source is mapped to a multi-track target by assigning instruments under a classification framework.

BL-Pno.: We introduce *Poly-Dis* by [Wang *et al.*, 2020b] as our baseline for the piano cover generation task. This model is pre-trained on POP909 and is also based on style transfer. Specifically, it can generate piano cover for a multi-track source piece by reconceptualizing its chord progression using the texture from a piano reference. In our case, we provide Poly-Dis with the same piano reference as our model and extract the chord progression of the source music using the algorithm in [Jiang *et al.*, 2019].

BL-ReIns.: We introduce the model by [Hung *et al.*, 2019] as our baseline for the re-instrumentation task. This model rearranges a source piece using the synthesized audio timbre feature from a reference piece as instrumentation style. We train this model on Slakh2100 using both the MIDI and the aligned audio that is synthesized using professional-level sample-based virtual instruments [Manilow *et al.*, 2019].

Besides the baselines, we also introduce three variants of our model to analyze the impact of each key component. Specifically, ***Q&A-T*** uses only time function as query to rearrange a piece. ***Q&A-P***, on the other hand, uses pitch function only. The final variant ***Q&A w/o Ins.*** uses both functions but is trained without instrument embedding.

Evaluation Details

We invite participants to subjectively evaluate the rearrangement quality of all models through a double-blind online survey. Our survey consists of 15 rearrangement sets, each of which contains one original source piece followed by five rearrangement samples (four by our model variants and the rest by one of the baselines). Among the 15 sets, there are 5 for piano cover generation, orchestration, and reinstrumentation, respectively. The original piece x is an 8-bar musical phrase randomly selected from the validation/test set of either POP909 or Slakh2100 depending on the task. The reference piece y is sampled from the other dataset following Equation (8), where we set $\alpha = 0.2$. For re-instrumentation, both x and y are in Slakh2100 but from different splits (validation and test sets, respectively). We use the same y for each model that requires a reference piece for style transfer.

In our survey, we request each participant to listen to 5 rearrangement sets and evaluate each sample. Both the set order and the sample order in each set are randomized. The

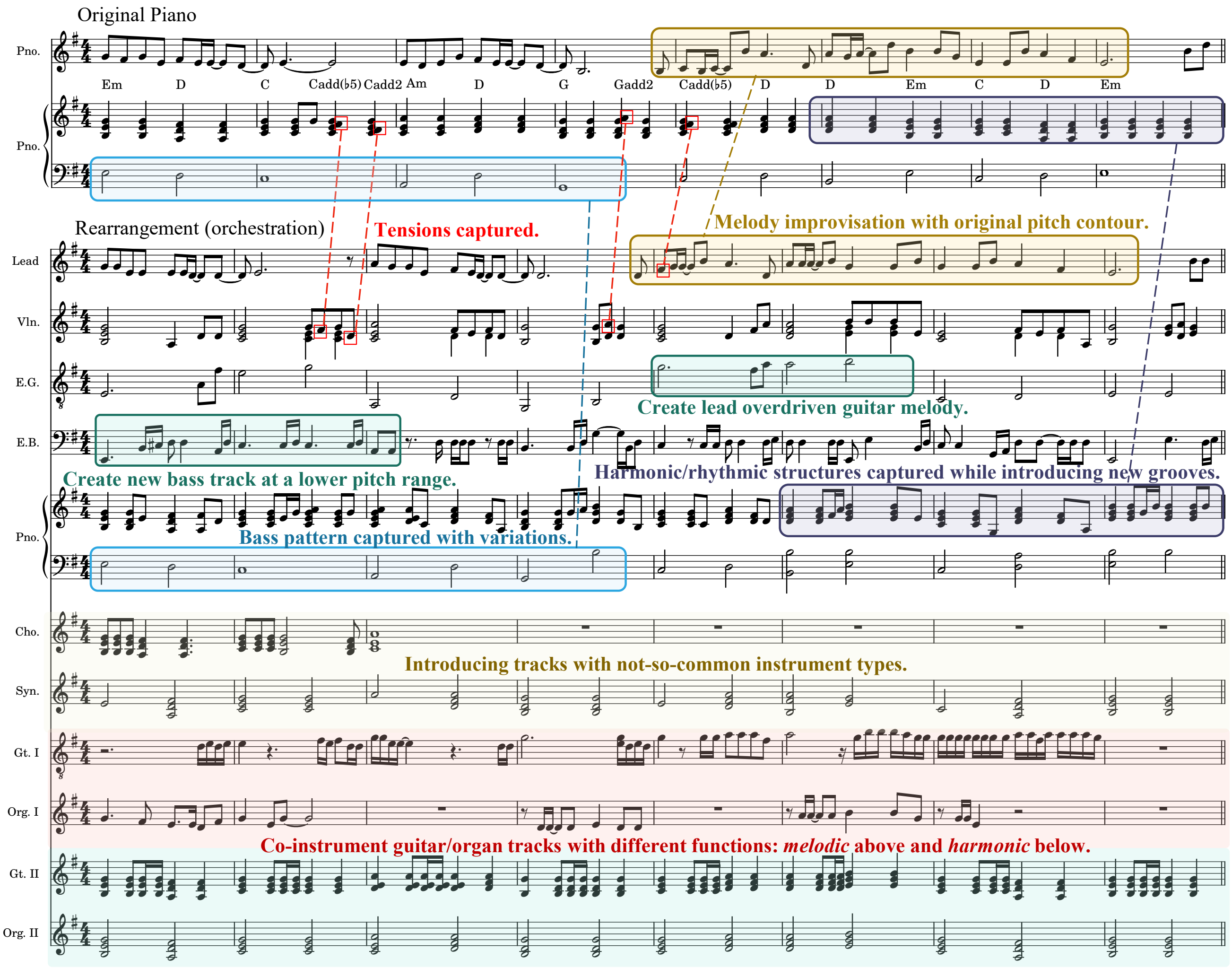
Original Piano
Pno.
Em
D
C
Cadd(♭5)
Cadd2
Am
D
G
Gadd2
Cadd(♭5)
D
D
Em
C
D
Em
Pno.
Rearrangement (orchestration)
Tensions captured.
Melody improvisation with original pitch contour.
Lead
Vln.
E.G.
Create lead overdriven guitar melody.
E.B.
Create new bass track at a lower pitch range.
Harmonic/rhythmic structures captured while introducing new grooves.
Pno.
Bass pattern captured with variations.
Cho.
Introducing tracks with not-so-common instrument types.
Syn.
Gt. I
Org. I
Co-instrument guitar/organ tracks with different functions: melodic above and harmonic below.
Gt. II
Org. II

Figure 3: An orchestration example for song_283 from POP909 by our proposed Q&A model. The result has 11 tracks with varied instruments. Annotations illustrate that the rearrangement is both faithful and creative with a delicate multi-track structure.

evaluation is based on a 5-point scale from 1 (very low) to 5 (very high) for four criteria as follows:

- **DOA**: The degree of arrangement. A low DOA refers to a note-by-note copy-paste from the original music, while a high DOA means the music is appropriately restructured to fit the new track system and instruments.
- **Creativity**: How creative the rearrangement is.
- **Naturalness**: How likely a human arranger creates it.
- **Musicality**: The overall musicality.

Overall Rearrangement Performance

A total of 26 participants (8 females and 18 males) with various musical backgrounds have completed our survey. We first show the statistical results for *overall* rearrangement performance disregarding the specific tasks. As shown in Figure 4, the height of each bar represents the mean rating value and the error bar represents the standard error computed via within-subject (repeated-measures) ANOVA [Scheffe, 1999]. Among our model variants, *Q&A-T* queries the mixture by time function only, and *Q&A w/o Ins.* has no instrument embedding. Both models essentially have fewer constraints during training and hence can produce more diverse results, which may explain the higher ratings on DOA and Creativity for both models. However, such results can also be less natural or musical. On the other hand, *Q&A-P* uses pitch function only and yields results inferior to other variants. This finding shows that pitch function alone is not sufficient to capture track structures in multi-track music. Indeed, pop music (at least in our datasets) is generally better characterized in grooves than in chords, as the latter can often fit in a few off-the-shelf template progressions. In terms of Naturalness and Musicality, our standard *Q&A* model makes a better balance and acquires significantly better results (p-value $p < 0.01$) than all variants and the baselines ensemble.

Task-Specific Performance

We are also interested in our model's performance on each concrete rearrangement task. As shown in Figure 5, we show our model's *task-specific* ratings (top in four variants) on the

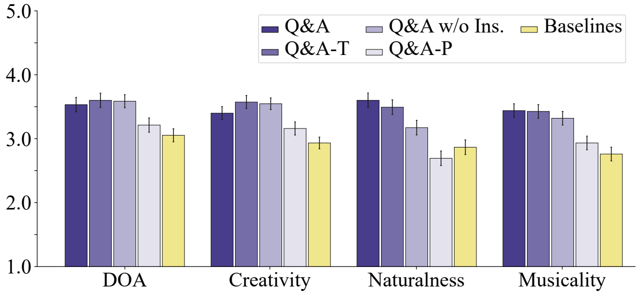

Figure 4: Evaluation on overall rearrangement performance.

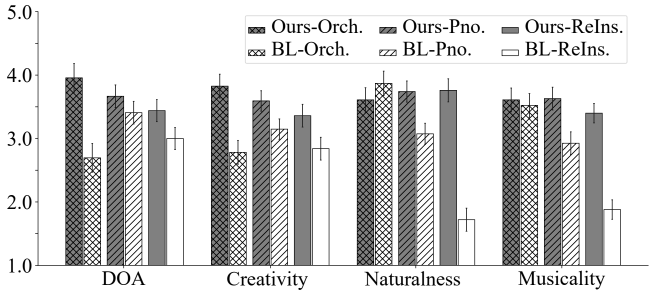

Figure 5: Evaluation on task-specific rearrangement performance.

Model	Chorales	Quartets
Q&A-V	94.84[†]	73.47[†]
Transformer	96.81	58.86
BiLSTM	**97.13**	**74.38**
Q&A-V (+ entry hints)	95.11[†]	**78.71**[†]
Transformer (+ entry hints)	93.81	56.72
BiLSTM (+ entry hints)	**97.39**	71.51

Table 1: Objective evaluation on voice separation comparing to note-by-note architectures. We use [†] to denote test results under 10-fold cross validation. Baseline results are from [Dong *et al.*, 2021].

same set of criteria in comparison to corresponding baseline models. We notice that *BL-Orch.* earns the highest rating for Naturalness in the orchestration task, which is not surprising because it adopts a note-by-note mapping strategy that virtually reproduces the original human-created music. Accompanied by this strategy is a lower degree of orchestration (DOA) and Creativity. On the other hand, our model demonstrates a more balanced and superior performance. It also outperforms *BL-Orch.* in Musicality as it can introduce more diversified instruments and properly rearrange the source music with new texture and voicing. In particular, we report a significantly better performance (p-values $p < 0.05$) of our model in Musicality than all baselines in all three tasks.

4.5 Objective Evaluation on Voice Separation

One may wonder if the exceptional performance of our model in creativity and musicality sacrifices the faithfulness to the original music. To this end, we conduct an additional experiment on the task of voice separation and compare our model with note-by-note decision models — a BiLSTM and a Transformer encoder tailored for this task in [Dong *et al.*, 2021]. Specifically, voice separation is a special case of orchestration, where we only aim to separate a mixture into individual voicing tracks without any creative factor. Note that, in such a case, note-by-note classification methods have a natural advantage over representation learning based methods because the latter gives less importance to accurate control on low-level tokens. Still, the faithfulness of our model can be validated if it can also tackle this problem.

In voice separation, since the goal is to separate individual tracks, the ground-truth track functions cannot be the model input. Hence we introduce a new variant *Q&A-V*, which applies an additional GRU decoder to infer function representations $z_{1:N}^{f(x)}$ from mixture representation z_{mix}^{x}, and then generate each track with inferred track functions. In our case, $N = 4$ is preset and the inference process is conducted from high voice to low voice autoregressively. We load the rest part of the model with pre-trained parameters from standard *Q&A* and fine-tune the whole model on string quartets in MusicNet [Thickstun *et al.*, 2017] and Bach chorales in Music21 [Cuthbert and Ariza, 2010], respectively. We process the data into 8-beat segments irrespective of time signature. At test time, if a certain note in the mixture is not recalled by our model, we look for its nearest-neighbour note that is generated and assign its voice. If two note assignments form polyphonic voice, we then re-assign the note with least added distance to its second-nearest voice, which is a simple greedy-based rule. As both datasets are tiny and prone to unbalanced train-test split, we evaluate our model by 10-fold cross validation.

We show the test results (percentage accuracy) in Table 1. Compared to the baselines, our *Q&A-V* model yields generally comparable results, although with a noticeable gap on Bach chorales. Specifically, Bach chorales come with very regular and transparent counterpoints, which are a good fit for note-by-note classification frameworks to separate voices. On the other hand, string quartets have much more complex and even overlapped voices that are harder to separate. For this case, our model yields highly competitive performance in general. When entry hints are provided, our model achieves the best with a good margin to both baselines.

5 Conclusion

In conclusion, we contribute Q&A, a novel query-based framework for multi-track music rearrangement. The main novelty lies first in our application of a style transfer methodology to interpret the general rearrangement problem. By defining and utilizing track functions, we effectively capture the texture and voicing structure of multi-track music as composition style. Under a self-supervised query system, the number of tracks and instruments to rearrange a piece is virtually unconstrained. Q&A serves as a unified solution for piano cover generation, orchestration, re-instrumentation, and voice separation. Extensive experiments prove that it can both creatively rearrange a piece and faithfully conserve the essential structures. We believe that our contributions will inspire further advancements in computer music research, opening doors to broader possibilities for universal music co-creation.

References

[Couturier *et al.*, 2022a] Louis Couturier, Louis Bigo, and Florence Levé. Annotating symbolic texture in piano music: a formal syntax. In *Proceedings of the 19th Sound and Music Computing Conference*, pages 577–584, 2022.

[Couturier *et al.*, 2022b] Louis Couturier, Louis Bigo, and Florence Levé. A dataset of symbolic texture annotations in mozart piano sonatas. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, pages 509–516, 2022.

[Crestel and Esling, 2017] Léopold Crestel and Philippe Esling. Live orchestral piano, a system for real-time orchestral music generation. In *Proceedings of the 14th Sound and Music Computing Conference*, pages 434–442, 2017.

[Crestel *et al.*, 2017] Léopold Crestel, Philippe Esling, Lena Heng, and Stephen McAdams. A database linking piano and orchestral MIDI scores with application to automatic projective orchestration. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*, pages 592–598, 2017.

[Cuthbert and Ariza, 2010] Michael Scott Cuthbert and Christopher Ariza. Music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 637–642, 2010.

[Dai *et al.*, 2018] Shuqi Dai, Zheng Zhang, and Gus G Xia. Music style transfer: A position paper. *arXiv preprint arXiv:1803.06841*, 2018.

[Donahue *et al.*, 2019] Chris Donahue, Huanru Henry Mao, Yiting Ethan Li, Garrison W. Cottrell, and Julian 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*, pages 685–692, 2019.

[Dong *et al.*, 2018] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 34–41, 2018.

[Dong *et al.*, 2021] Hao-Wen Dong, Chris Donahue, Taylor Berg-Kirkpatrick, and Julian J. McAuley. Towards automatic instrumentation by learning to separate parts in symbolic multitrack music. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, pages 159–166, 2021.

[Ens and Pasquier, 2020] Jeff Ens and Philippe Pasquier. Mmm: Exploring conditional multi-track music generation with the transformer. *arXiv preprint arXiv:2008.06048*, 2020.

[Hendrycks and Gimpel, 2016] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[Higgins *et al.*, 2017] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, Conference Track Proceedings*. OpenReview.net, 2017.

[Hsiao *et al.*, 2021] Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 178–186, 2021.

[Huang and Yang, 2020] Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1180–1188, 2020.

[Huang *et al.*, 2019] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *7th International Conference on Learning Representations*. OpenReview.net, 2019.

[Hung *et al.*, 2019] Yun-Ning Hung, I-Tung Chiang, Yi-An Chen, and Yi-Hsuan Yang. Musical composition style transfer via disentangled timbre representations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4697–4703, 2019.

[Jiang *et al.*, 2019] Junyan Jiang, Ke Chen, Wei Li, and Gus Xia. Large-vocabulary chord transcription via chord structure decomposition. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 644–651, 2019.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Krizhevsky *et al.*, 2017] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[Lee *et al.*, 2019] Jie Hwan Lee, Hyeong-Seok Choi, and Kyogu Lee. Audio query-based music source separation. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 878–885, 2019.

[Liu *et al.*, 2022] Jiafeng Liu, Yuanliang Dong, Zehua Cheng, Xinran Zhang, Xiaobing Li, Feng Yu, and Maosong Sun. Symphony generation with permutation invariant language model. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, pages 551–558, 2022.

[Manilow *et al.*, 2019] Ethan Manilow, Gordon Wichern, Prem Seetharaman, and Jonathan Le Roux. Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 45–49. IEEE, 2019.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814, 2010.

[Nakamura and Sagayama, 2015] Eita Nakamura and Shigeki Sagayama. Automatic piano reduction from ensemble scores based on merged-output hidden markov model. In *Proceedings of the 41st International Computer Music Conference*, 2015.

[Nakamura and Yoshii, 2018] Eita Nakamura and Kazuyoshi Yoshii. Statistical piano reduction controlling performance difficulty. *APSIPA Transactions on Signal and Information Processing*, 7, 2018.

[Payne, 2019] Christine Payne. Musenet. https://openai.com/blog/musenet/, 2019. Accessed: 2023-05-29.

[Raffel, 2016] Colin Raffel. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. PhD thesis, Columbia University, USA, 2016.

[Ren *et al.*, 2020] Yi Ren, Jinzheng He, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Popmag: Pop music accompaniment generation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1198–1206, 2020.

[Scheffe, 1999] Henry Scheffe. *The analysis of variance*, volume 72. John Wiley & Sons, 1999.

[Takamori *et al.*, 2017] Hirofumi Takamori, Haruki Sato, Takayuki Nakatsuka, and Shigeo Morishima. Automatic arranging musical score for piano using important musical elements. In *Proceedings of the 14th Sound and Music Computing Conference*, pages 35–41, 2017.

[Terao *et al.*, 2022] Moyu Terao, Yuki Hiramatsu, Ryoto Ishizuka, Yiming Wu, and Kazuyoshi Yoshii. Difficulty-aware neural band-to-piano score arrangement based on note-and statistic-level criteria. In *International Conference on Acoustics, Speech and Signal Processing*, pages 196–200. IEEE, 2022.

[Thickstun *et al.*, 2017] John Thickstun, Zaïd Harchaoui, and Sham M. Kakade. Learning features of music from scratch. In *5th International Conference on Learning Representations*. OpenReview.net, 2017.

[Toomarian and Barhen, 1992] Nikzad Benny Toomarian and Jacob Barhen. Learning a trajectory using adjoint functions and teacher forcing. *Neural networks*, 5(3):473–484, 1992.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 5998–6008, 2017.

[Wang and Xia, 2021] Ziyu Wang and Gus Xia. Musebert: Pre-training music representation for music understanding and controllable generation. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, pages 722–729, 2021.

[Wang *et al.*, 2020a] Ziyu Wang, Ke Chen, Junyan Jiang, Yiyi Zhang, Maoran Xu, Shuqi Dai, and Gus Xia. POP909: A pop-song dataset for music arrangement generation. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, pages 38–45, 2020.

[Wang *et al.*, 2020b] Ziyu Wang, Dingsu Wang, Yixiao Zhang, and Gus Xia. Learning interpretable representation for controllable polyphonic music generation. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, pages 662–669, 2020.

[Wang *et al.*, 2020c] Ziyu Wang, Yiyi Zhang, Yixiao Zhang, Junyan Jiang, Ruihan Yang, Gus Xia, and Junbo Zhao. PIANOTREE VAE: structured representation learning for polyphonic music. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, pages 368–375, 2020.

[Wang *et al.*, 2022] Ziyu Wang, Dejing Xu, Gus Xia, and Ying Shan. Audio-to-symbolic arrangement via cross-modal music representation learning. In *International Conference on Acoustics, Speech and Signal Processing*, pages 181–185. IEEE, 2022.

[Wuerkaixi *et al.*, 2021] Abudukelimu Wuerkaixi, Christodoulos Benetatos, Zhiyao Duan, and Changshui Zhang. Collagenet: Fusing arbitrary melody and accompaniment into a coherent song. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, pages 786–793, 2021.

[Yang *et al.*, 2019] Ruihan Yang, Dingsu Wang, Ziyu Wang, Tianyao Chen, Junyan Jiang, and Gus Xia. Deep music analogy via latent representation disentanglement. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 596–603, 2019.

[Yi *et al.*, 2022] Li Yi, Haochen Hu, Jingwei Zhao, and Gus Xia. Accomontage2: A complete harmonization and accompaniment arrangement system. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, pages 248–255, 2022.

[Zhao *et al.*, 2022] Jingwei Zhao, Gus Xia, and Ye Wang. Domain adversarial training on conditional variational auto-encoder for controllable music generation. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, pages 925–932, 2022.

[Zhu *et al.*, 2018] Hongyuan Zhu, Qi Liu, Nicholas Jing Yuan, Chuan Qin, Jiawei Li, Kun Zhang, Guang Zhou, Furu Wei, Yuanchun Xu, and Enhong 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*, pages 2837–2846, 2018.