Motif-based Music Representation Learning and Structure Analysis

Submitted in partial fulfillment of the requirements for the degree of

Master of Science in Music and Technology

School of Music

Carnegie Mellon University, Pittsburgh, PA

Yuxuan Wu

Thesis Committee:
Roger B. Dannenberg, Chair
Gus Xia
Thomas Sullivan

July 2023

Acknowledgements

I would like to express my heartfelt gratitude to my advisor, Roger Dannenberg, for his warm mentorship throughout my journey in this program. Your knowledge, expertise and sense of humor supported me through my difficult times and helped me grow into a better researcher, as well as a better human being. It has been an honor to be your student, and in my future academic journey, you will continue to shine as a beacon of inspiration in my heart. I would also like to thank my thesis committee for your valuable advice. This study would not have been possible without you. Lastly, I want to express my gratitude to my family and my significant other, who have been with me throughout tears and joy. Your love, encouragement, and belief in me have been my pillars of strength. You are the reasons I am here today, living and thriving, and I am forever thankful for your presence in my life.

Abstract

The formation of music structure heavily relies on repetitions and variations of music motifs. Understanding the manifestations and behaviors of these motifs is crucial for effective music structure analysis and high-quality automatic music composition. However, capturing music motifs' implicit nature is often challenging. In this study, we employ deep learning techniques to explore an efficacious method for learning robust representations of music motifs. Specifically, we propose self-supervised pretraining for the music motif encoder, followed by fine-tuning on a manually labeled dataset. Additionally, we adopt a novel training strategy based on pure regularization, which yields competitive results during the pretraining phase. We also study different fine-tuning schemes and propose a method that leverages the strengths of different training strategies and yields the best results. Lastly, we conduct an intuitive visual analysis of music motifs based on the acquired representations. We hope this work can offer valuable insights for future discussions on the potential of music motifs in structured music generation, as well as music information retrieval.

Keywords Representation learning · Music structure · Music motif · Self-supervised learning · Pretrain

Contents

Chapter 1

Introduction

"Music is organized sound." The structured relationship among music elements forms the meaning of music. The structure in music presents on different levels, which from low to high include (1) individual notes and their pitch and time intervals (2) chords, phrases, and chord progressions (3) broader musical sections and parts over the entire piece [1].

Music structure exists in the forms of repetition, imitation, conversation, transposition, and so on, but relies mostly on repetition to create coherence. The coherence can come from repetitions at any level and any form. For instance, the thematic phrase in the chorus part is often repeated three times in an R&B song; the same chord progression might be cycling throughout a jazz improvisation, regardless of how wildly the melody is evolving; two distinct types of left-hand patterns mark the clear boundary between sections in a Beethoven piano sonata. Among the elements recurring in music that form music meanings, motifs are often the smallest part of a piece or section of a piece that, despite change and variation, is recognizable as present throughout [2]. Be it a remarkable distribution of three or four notes in the melody like the opening punches in Beethoven's *Symphony No. 5, First Movement*, or an impressive polyphonic pattern that carries different chords like the exquisite voicing in Bill Evans' *Waltz for Debby*, one thing these motifs have in common is that they reappear multiple times in diverse forms, yet retaining their recognizability and even iconic significance, as shown in Fig. 1.1. Unlike the often confused concept of musical theme, motifs can be very diverse and varied within each piece of music, as shown in Fig. 1.2, whereas themes are generally limited to very few, often one or two, and are usually the most important motifs. The coherence of motifs and variety in different statements of motifs endow music with the aesthetics of duality between constancy and changes, predictability and surprise, etc. If music can ever be considered a language, it must be a poetic language consisting of metaphors, as the references of motifs bring the feeling of "this equals that" [3]. Naturally, music motifs serve as the nouns or characters in poems, and their manifestations and

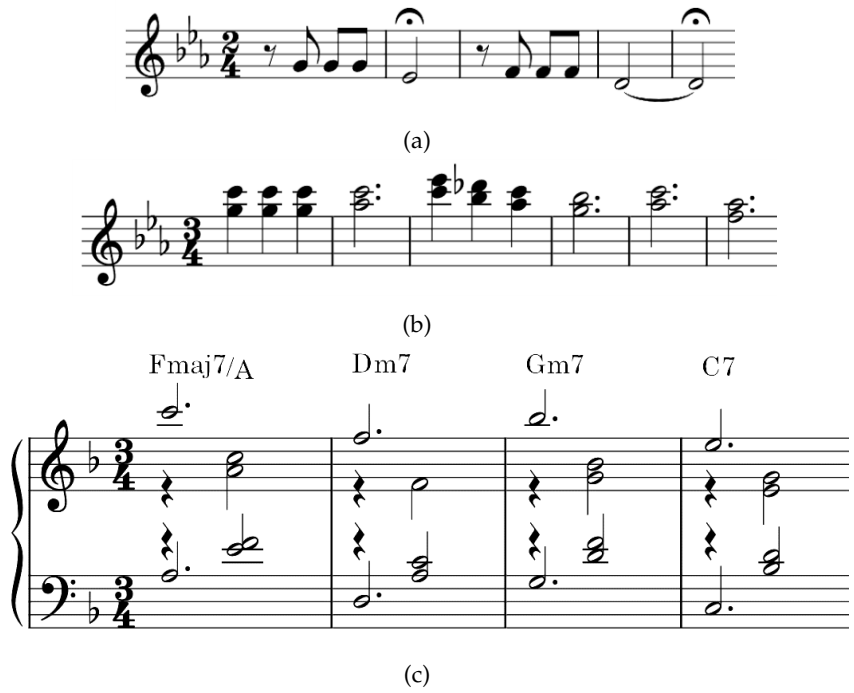

Figure 1.1: The existence of music motifs in different forms. (a) The opening melody in Beethoven's *Symphony No. 5, First Movement* (b) The thematic phrase in Beethoven's *Symphony No. 5, Third Movement* (c) The opening measures in Bill Evans' *Waltz for Debby*. In both (a) and (b), three same short notes are followed by a separate long note, which leads to the perception of a repeating motif. The formation of this motif involves both temporal and pitch information. It first came out in (a) and echoed in (b), demonstrating that motif occurrences can span across movements. In (c), the motif is the melody and bassline followed by a close voicing. This pattern transforms with the variation of the chord it carries.

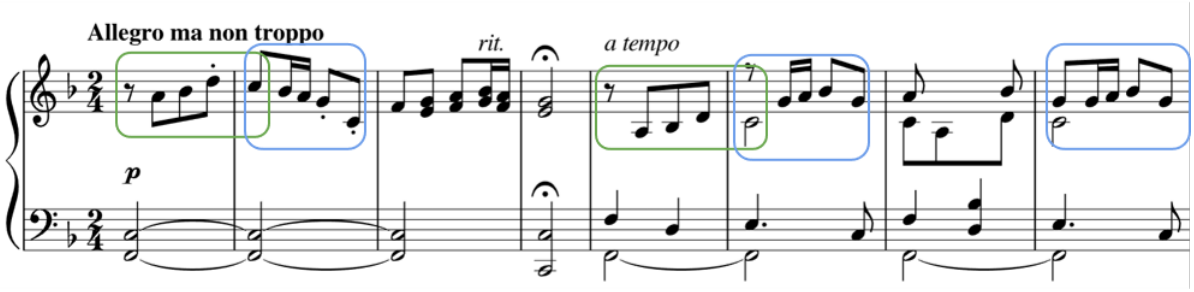

Figure 1.2: Diversity of music motifs within a piece. Parts inside green boxes are from the same motif, and parts inside blue boxes are from another. They can develop their respective transformations as the music progresses.

behaviors witness the unfolding of the whole story.

Unfortunately, though it is usually easy for human appreciators to capture and understand these "metaphors" in musical language, it is usually a lot more difficult for computer programs, because the musical meaning in the language is more often implicit than straightforward. However, understanding

music motifs is an essential step toward understanding the structure of music for machine music generation and high-level music information retrieval (MIR).

Statements of the same motif are intrinsically similar at the perception level. The prevalence of similarities among music units has long attracted wide attention from music understanding research. Measuring similarity between melodies is a major concern, as the human cognition of melodic similarity is often complex. Most traditional methods comprehensively use statistical or alignment methods to measure the similarities between two query melodies [4, 5, 6, 7, 8, 9]. Recently, deep learning also shows great potential in measuring melodic similarity more robustly, especially through the progress in music style transfer [10, 11]. However, these research works focus more on the music generation side than the understanding side, and the potential of these learned representations in the mining of music semantics is yet to be explored.

Another direction to study the repetitions in symbolic music is pattern discovery, which aims to find important patterns and their occurrences in a given music piece. Efforts have been made to design ingenious algorithms to discover repeated patterns in symbolic music [12, 13, 14]. Similar methods can be applied to music motif analysis [15, 16], but they lack the learning of effective deep representations, thereby limiting their broader applications when combined with deep learning models. Other direct trials of motif discovery also suffer from the same plight [17, 18, 19].

Nowadays, Deep learning techniques have been proved to achieve remarkable performance across the majority of modalities. The abstract outcomes or intermediate products of those deep neural networks are considered useful representations that captured complex and important information in the data. Representations of multiple aspects of symbolic music have been well studied, such as the composition style, chord progression, melody contour and rhythm patterns [11, 10, 20, 21, 22, 23], but learning useful representations of music motifs remains a challenge. An existing method leveraged contrastive learning on segmented music to learn representations for the thematic information in music, and demonstrated their potential in downstream tasks like music generation [24]. However, their method hasn't been comprehensively evaluated in terms of validity or effectiveness.

To promote the research on motif-based music representations, in this thesis, we propose a novel pretraining-based approach for music motif representation learning, and testified its effectiveness on a hand-labeled motif dataset. We conduct the study on pop piano accompaniments, for they embody clearer motif boundaries, typical variations like transposition and inversion, and the polyphonic nature that covers the universality of music. Specifically, we propose to pretrain a neural network encoder with self-supervised learning, and use fine-tuning to adapt the model to a hand-labeled real dataset. We propose to use a novel regularization-based training strategy in the pretraining stage, which significantly improves performance

compared to contrastive learning. We also compare the performances of different transfer learning schemes on the real dataset, and prove the outstanding capability of the proposed method. In addition, we conduct an intuitive visual analysis based on the learned representations, demonstrating the broad potential of motif representations on downstream research applications. To sum up, the main contributions of this work include:

- Proposing an effective approach to learn deep representations for music motifs. By reflecting on the drawbacks of existing methods and gaining insights from other modalities, we design a better training approach that substantially improves the modeling performance.
- Introducing a hand-labeled dataset of music motifs in pop piano accompaniment. The dataset is based on the well-known POP909 [25], and can be easily extended or customized.
- Providing a visualization method for motif-based music structure analysis.

We hope this work can offer valuable insights for structured music generation and music information retrieval, as deep representations of motifs are crucial for analyzing their impact on coherence and diversity in compositions. The findings aim to inspire discussions and collaborations among researchers, musicians, and enthusiasts, fostering creativity and enhancing musical experiences.

Chapter 2

Related Work

2.1 Motivic Analysis of Music

The perception of motifs comes from the sense of similarity. One natural direction to study music motifs is finding a better measurement for symbolic music similarity. The research problem of melody similarity measurement is vital for the field of music information retrieval (MIR), application scenarios like query-by-humming [26], and also music generation. A perhaps counter-intuitive fact here is that direct metrics of distances like the edit distance can't always reflect the level of similarity in the musical meaning, indicating the cognitive complexity of the task [8]. As a result, despite years of research, finding a proper similarity measurement between symbolic music remains a challenge. Traditional methods incorporate combinations of a variety of similarity functions, including edit distance, geometric distance, correlation coefficient, N-gram similarity measures and dynamic time warping (DTW) distance [4, 5, 6, 7, 9]. Park et al. propose a cross-scape plot representation to visualize and measure multi-scaled melody similarity [27]. Neural network-based approaches also demonstrate great capabilities in melody similarity measurement. Melody2Vec, an extension of the Word2Vec framework to melodies shows its effectiveness in representing the semantic relatedness between melodic phrases [28, 29]. Karsdorp et al. explore the application of RNNs for end-to-end melody similarity measurement learning using contrastive learning and show great robustness [30].

Another closely related research direction is music pattern discovery. Music pattern discovery aims to identify regularities in music, including repetitions and variations. Though closely related, music pattern discovery is slightly different from music motif analysis in that patterns can be very long, while motifs are often localized music ideas. A classical general approach is the geometrical-inspired method, attempting to identify groups of notes with similar geometric features and extract these groups as musical patterns.

[31, 32, 33, 34, 12, 14]. MotivesExtractor [35] combines the Gestalt grouping principles [36] with clustering techniques, and adjusts the parameters according to human feedback. PatMinr uses an incremental one-pass approach to identify pattern repetitions, considering various music attributes such as chromatic features, diatonic pitches and metrical positions [16]. It addresses close pattern mining and pattern cyclicity to avoid redundancy. SymCHM employs the compositional hierarchical model to learn hierarchical melodic structures in an unsupervised manner [37]. Its underlying assumption is that the repetitions of patterns can be captured by observing statistics of occurrences of their sub-patterns. Although research on music pattern recognition is abundant, it often involves different assumptions [38]. Some of these assumptions are based on theoretical compositional rules, while others are grounded in the statistical features of the data itself. However, they may not integrate well with each other, which to some extent hinders broader application.

2.2 Deep Music Representations

Representation learning has evolved as a crucial concept in the field of machine learning and artificial intelligence over the years, particularly with the emergence of powerful neural networks like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). These successes showcase the power of representation learning in transforming raw data into meaningful and useful representations. Transformer further revolutionized representation learning by introducing a novel and highly effective approach for capturing long dependencies in sequential data. In a typical classification system, such as a CNN-based image classifier, the final layer is often considered the classifier, which gives final predictions based on the learned representation, and the output of all previous layers is considered the learned representation. By learning through the iterative training process, the latent representation often has a significantly lower dimension than its original form. Under the unsupervised setting, architectures like Autoencoder (AE) and Variational Autoencoder (VAE) learn deep representations of data through self-supervision of reconstruction from compressed encodings.

Like other modalities, the music representations learned by deep learning depend on the specific task being addressed. Over the years, efforts have been made to model a variety of aspects of music. For example, the representations of chords are well studied to resolve the task of automatic chord recognition from audio or symbolic music [39, 40, 21, 41, 42]; other works incorporate proper inductive biases to achieve the disentanglement of composition style and musical content during style transfer, acquiring valuable representations for those styles [43, 11, 10]; other widely studied music representations include melody contours [23, 44], rhythm patterns [22, 20] and so on.

Studies on the above representations of music are often closely bound to downstream tasks like music generation and MIR. In music generation, a ubiquitous problem of current systems is the lack of overall structuredness compared to human composers [45]. Current efforts to improve the structuredness of generated music mostly focus on the higher-level structures and long-term dependencies [46, 23, 47, 48, 49]. ThemeTransformer elaborates that thematic materials outperform simple prompts as conditions for music generation [24], since thematic musical ideas tend to be repeated by human composers to create coherence. This fact suggests that incorporating fine-grained musical ideas is a crucial step for realistic music generation. In the field of MIR, various types of music representations are widely used in search and recommendation systems, particularly main melody features for cover song recognition [50, 51, 52, 53] and genre features for recommendation systems [54, 55]. However, music retrieval and recommendation based on more local features remains an underexploited problem.

This work aims at promoting the research on representation learning for music motif, which is a complex yet essential aspect of music. Compared to traditional methods for motivic analysis, we reduce the impact of predefined assumptions and explore more possibilities for integration with downstream models. Compared to other music representation learning methods that focus on different aspects of music, our work can be seen as a complementary approach.

Chapter 3

Method Overview

An overview of the proposed motif-based music representation learning system will be introduced in this section. The overall target of this work is to train an effective encoder neural network for music motifs, so that music segments that are occurrences of the same motif will lie close to each other in the learned latent space. With this learned embedding space, we demonstrate the potential of motif-based music analysis using visualization.

We first pretrain an encoder neural network with self-supervised learning. We perform data augmentation for self-supervised learning by constructing a synthetic dataset with weak motif labels. Afterwards, we perform transfer learning on a manually labeled real dataset. We constructed both of the two datasets involved based on POP909 because it contains clean pop piano accompaniment data with clear beat and onset labels [25]. The overview of the dataset construction processes is shown in Fig.3.1. We first process the raw accompaniment tracks in POP909 MIDI files into data chunks, then use pre-defined "metaphorizing functions" to transfer these data chunks into their new forms, while keeping the music ideas behind. For

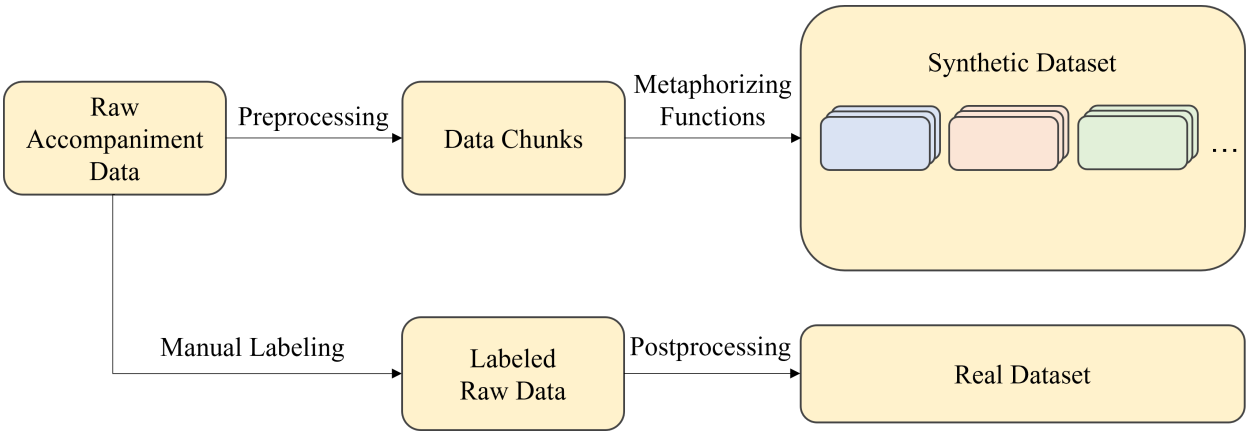

Figure 3.1: Overview of the dataset construction pipeline

every data chunk, new chunks derived from it are considered repetitions in the musical idea, thereby they are occurrences of the same motif. All data chunks together with their derivations compose the synthetic dataset. As for the real dataset, an additional track is added to the original POP909 MIDI to represent motif labels. Motifs are labeled considering the musical ideas conveyed in the accompaniment tracks. Parts carrying repeated musical ideas are labeled with the same pitch in the new track, presenting as the same MIDI number ranging from 0 to 127. The labeled data is further processed to align every data chunk with its motif label. It should be noted that the two datasets share the same source of validation set, and that the synthetic dataset is significantly larger than the real dataset.

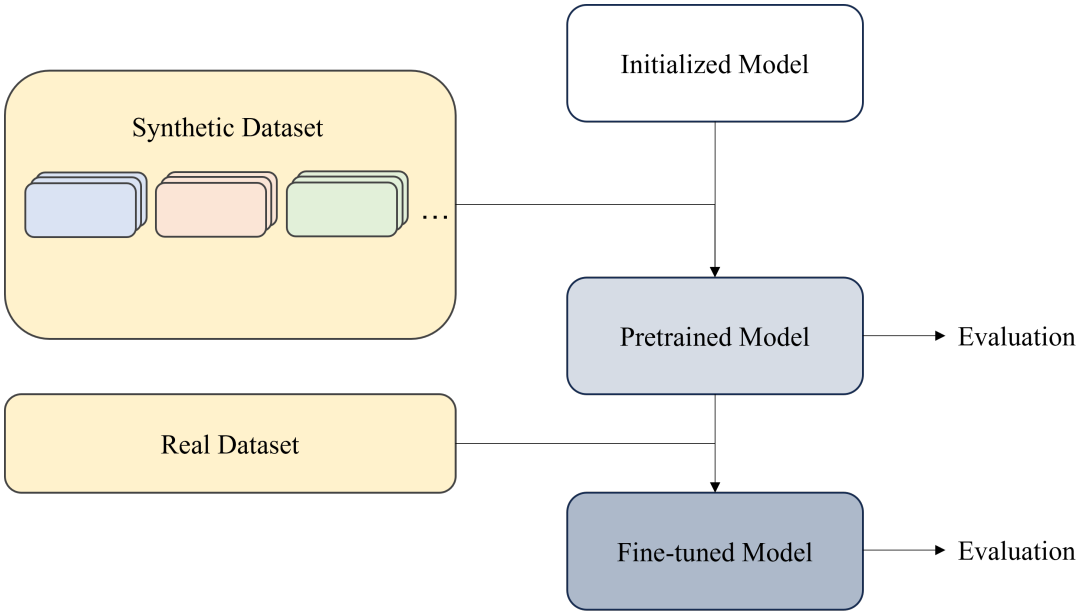

Figure 3.2: Overview of the model training process

The initialized model is first pretrained with the synthetic dataset, and then fine-tuned using the manually labeled real dataset. In both the pretraining stage and the fine-tuning stage, we run experiments on different training strategies. We compare the performances of (1) the proposed approach with a pure regularization-based training strategy (2) an existing method and its revision (3) a straightforward neural network-free baseline. The model training workflow is demonstrated in Fig.3.2. Finally, we select the best-performing model on the real dataset to conduct visualization. The details of dataset construction, model training methods and experiment results will be described in the following chapters.

Chapter 4

Motif Encoder Pretraining

4.1 Data Augmentation: The Synthetic Dataset

Self-supervised learning is a learning paradigm in artificial intelligence that works well in utilizing the intrinsic structure in the data as the supervision signal. Its effect is especially remarkable when labeled data is scarce. It involves the design of an auxiliary task, from which the loss is computed to guide the optimization of models. One typical auxiliary task is training the network to produce similar embeddings for different views of the same input, namely the Siamese network architecture [56]. The practice of producing different views of the same input can be viewed as a form of data augmentation.

In this work, we achieve similar data augmentation by constructing a synthetic dataset for self-supervised pretraining. In the synthetic dataset, every data chunk from POP909 has 5 transformations that are considered different views of itself. First, we take the accompaniment track of every MIDI file in POP909. We unify the tempo of every track to 60, and trim each track to start at a global onset using the onset information in the original dataset. We then quantize the onset and offset of every note with a resolution of 128th notes to eliminate fractions. Afterwards, we segment all the accompaniments into chunks. All the chunks have the same length of 1 bar, as one bar is a reasonable length for music motifs and especially for pop piano accompaniments, which are mainly driven by polyphonic patterns that carry the chords. Chunks with no more than three notes are discarded.

The data chunks are then transformed into different views using a set of "metaphorizing functions" designed to mimic possible variations of motifs in real music. The overall designing principle is to apply slight changes to the score while keeping the major musical idea unaltered. The functions used in this study include:

- Random transposition. A common variation of music segments is transposing to another key. The

motif expressed in music remains after transposition. The music chunk will be transposed as a whole to a randomly chosen key, with a maximum shift of 6 semitones in either direction.

- Random dropout. A random note will be deleted from the music chunk. Dropping out one note will not affect the musical idea behind in most cases, especially in pop piano accompaniments where notes are pretty dense. There is also a probability that this function will be disabled, and no note will be deleted.

- Random note shift. Similar to random dropout, a random note will be selected and shifted up or down with a maximum shift of 2 semitones.

- Random duration variation of the last note. The idea of this function is inherited from Theme-Transformer [24]. If the last note is longer than a 16th note, its duration will be randomly changed, provided that the length of the whole chunk does not exceed 1 bar.

We modify every chunk by applying the above functions one by one, so that the final modification is a random combination of these functions. Every chunk is processed 5 times to produce 5 different views,

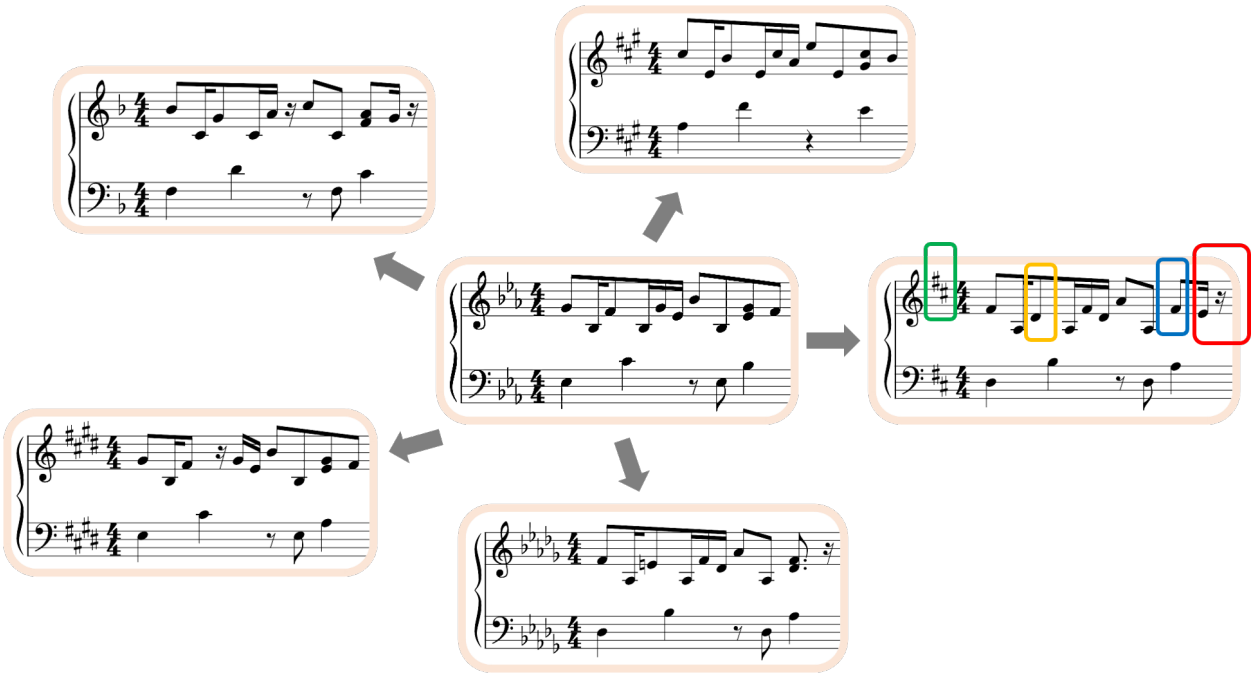

Figure 4.1: Using "metaphorizing functions" to transform a data chunk (in the middle) to 5 different views. The data chunk and all views derived from it are considered variations of the same motif. The view on the right is annotated with the effects of each function. Green: Random transposition. Blue: Random dropout. Yellow: Random note shift. Red: Random duration variation of the last note.

resulting in a total of 6 views, including the original chunk. The outputs are saved using the MIDI format. Fig. 4.1 demonstrates the process of metaphorizing a sample data chunk.

The resulting synthetic dataset contains 443028 MIDI files, produced from the data chunks segmented from 909 songs. The average number of chunks in each song is 81.23. We select the first 20 songs as the validation set, and the rest are used as the training set.

It should be noted that although these functions are designed to mimic real variations of music motifs, it is impossible to exhaust all possible cases of motif variations. With these functions, we aim to cover as many cases as possible while ensuring the credibility of the motif labels. The label space in the synthetic dataset is a subset of the label space in real data to keep the label shift controllable.

4.2 Model Pretraining

4.2.1 Model Structure

The MIDI files in the synthetic dataset can be parsed into the piano roll format. The piano roll is a matrix-based representation of music, where the height represents the number of valid pitches or MIDI numbers, the length corresponds to the temporal resolution, and the matrix values represent the velocity of every note. In this study, we limit the number of valid pitches to 84, as this range is sufficient to cover the pitches involved in the data. We set the temporal resolution to 32, so the MIDI data is quantized with 32th notes. We simplify the note velocities to only two values: 0 and 1. Here, 0 means that the note is not triggered, while 1 represents that the note is triggered. Consider a sample music chunk $\mathbf{X}$ in the synthetic dataset:

$$\mathbf{X} = [X_{ij}]_{84\times 32} \in \mathbb{D}_{syn}$$

where i denotes the pitch number, j represents the temporal grid index, and X_{ij} is either 0 or 1, indicating whether pitch i is played at time j. $\mathbb{D}_{syn}$ represents the synthetic dataset of music chunks. The goal is to train a neural network encoder $Enc(\cdot)$ to obtain the deep representation of $\mathbf{X}$:

$$\mathbf{z} = [z_1, z_2, \ldots, z_d] = Enc(\mathbf{X})$$

where z is an embedding vector of dimension d.

We select the Transformer encoder as the model backbone of the encoder. Transformer is a highly influential deep learning model primarily designed for sequence-to-sequence tasks, such as machine translation and text generation. Transformer originally has an encoder-decoder structure. The encoder processes the input sequence and produces contextualized representations for each sequence element,

and the decoder generates the output sequence based on the encoded representations. Both the encoder and decoder rely on the self-attention mechanism to capture long-range dependencies and contextual information by weighing the importance of each element relative to others [57]. Due to Transformer's strong ability in modeling sequential information, its encoder alone has also been applied in many influential works as representation extracters, such as BERT and XLNet [58, 59].

Our encoder network has a BERT-like structure [58], utilizing the encoder part of the original Transformer model [57] as our motif encoder. The model structure is demonstrated in Fig. 4.2. The piano roll representation for input $\mathbf{X}$ is considered a sequence in the time domain, and passed to an input embedding layer. A sequence of positional encodings generated from a sinusoid oscillator is added to the embedding. The positional encoding helps the model understand the sequential order of the input by providing unique positional information to each element in the sequence. The combined embedding is passed to a cascaded stack of Transformer encoder layers. In each Transformer encoder layer, we compute the multi-head self-attention from the input embedding by assigning the input to all of the queries, keys and values. The result is then added to the input, normalized, and further added to its own output through a fully connected layer before being normalized again. The input of each Transformer encoder layer after the first one is the output features from the previous layer. The output of these layers is projected to a lower-dimensional space and then pooled in the time domain. The final output is a latent representation $\mathbf{z}$ of the input $\mathbf{X}$, with a lower dimension and thus more compact information.

In this study, we set the latent dimension d to 128, the model size in fully connected layers and multi-head attention layers to 256, and use 6 cascaded Transformer encoder layers. As a result, every input X of size 84×32 is encoded to its latent representation $\mathbf{z} = Enc(\mathbf{X})$ of size 128. The total number of trainable parameters in this model is around 4.8M.

4.2.2 Baselines

We implement two baseline methods to obtain representations for music motifs. The first baseline involves contrastive learning, a commonly-used training strategy in self-supervised learning [60]. Instead of requiring explicit labels, contrastive learning is capable of learning with only positive and negative pairs. The second baseline is a model-free method, using the direct piano roll features as the representations.

We first describe the baseline method based on contrastive learning, which bears resemblance to the self-supervised training method for music themes proposed in ThemeTransformer [24]. For each sample data chunk from the dataset, also known as the anchor, the other views of this sample are considered positive samples. Data chunks from other songs in the dataset are all considered negative samples. Other

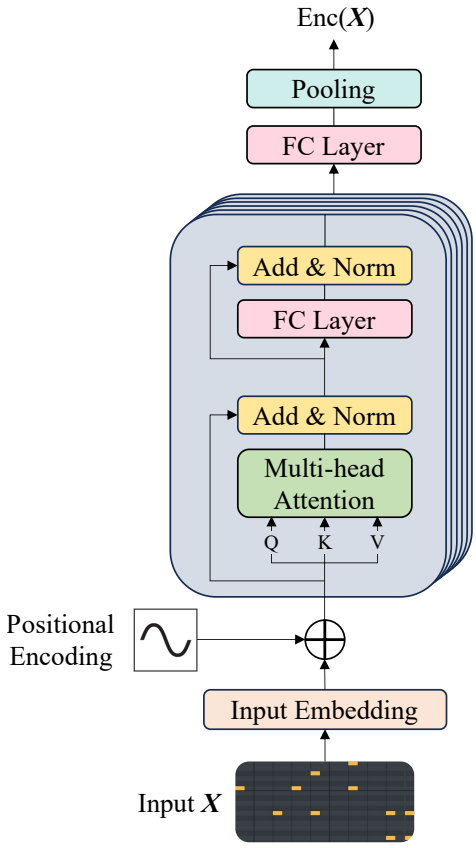

Figure 4.2: Backbone model structure of the BERT-like encoder

chunks from the same song, although not different views of the anchor, are not considered negative samples, because motifs are likely to be repeated in many places of a song. For every anchor $\mathbf{X}$, one positive sample $\mathbf{X}^+$ and one negative sample $\mathbf{X}^-$ are stochastically selected to form a positive pair and a negative pair. All three samples are passed through the encoder $Enc(\cdot)$ to obtain their respective latent representations $\mathbf{z}$, $\mathbf{z}^+$ and $\mathbf{z}^-$. The training objective is a contrastive loss function among the three representations. In this study, we adopt triplet loss [61], which can be formulated as:

$$\mathscr{L} = \max\left(||\mathbf{z} - \mathbf{z}^+|| - ||\mathbf{z} - \mathbf{z}^-|| + margin, 0\right)$$

where $||\cdot||$ is the distance metric between two points, and *margin* is a positive constant. The presence of *margin* could prevent the model from collapsing. Otherwise, the loss can be minimized to zero if the model produces the same output for every input. The presence of the hinge function prevents the loss from becoming negative, which could potentially lead to unintended behaviors in the optimization process. In our experiments, the *margin* is set to 1 as this value allows the model to converge after an appropriate amount of training time. An example iteration of training via contrastive loss is shown in Fig. 4.3.

The trained encoder can map the input samples into a latent representation space, where information

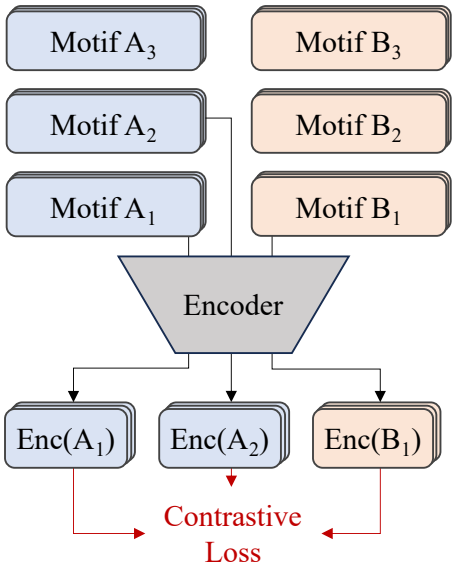

Figure 4.3: An example iteration of training the encoder using contrastive loss. A_1, A_2 and A_3 are occurrences of the same motif, and B_1, B_2 and B_3 are occurrences of another motif. Motif objects are stacked to represent multiple samples in a batch. In this iteration, A_1 and A_2 form a positive pair because they are different views of each other, while A_1 and B_1 form a negative pair because they come from different songs. Their encodings are used to compute the contrastive loss.

of music motifs is better reflected than in the original form. Ideally, in the latent space, the embeddings of two occurrences should be close to each other, while the embeddings of two unrelated motifs should be distant. The distances can be computed using the Euclidean distance. The trained encoder can be viewed as a music similarity metric from the perspective of music motif.

Another baseline method adopted in this study uses the direct piano roll features as the representations. This method does not require a neural network-based encoder, but relies on the straightforward geometric similarity measurement from piano roll representations. The similarity between two piano rolls is computed by their Euclidean distance. To eliminate the influence of chord and key differences, the piano rolls' matrices are first pitch-shifted so that their lowest note appears in the first row. We refer to this representation as the interval-based piano roll, since it eliminates the effect of global transpositions and only preserves the pitch interval information.

4.2.3 Regularization-based Training

We notice during preliminary experiments that the contrastive learning method described previously is slightly problematic in principle. During training, although the positive sample is guaranteed to be another view of the anchor, the randomly selected negative sample might not be truly negative, since even different

music pieces can share similar local music ideas. It is common that ideas can be shared across music pieces, whether in the melody or the accompaniment, and music motif is no exception. Typical cases include accompaniment textures composed by the same artist in different songs, drum patterns by different bands of the same genre, and even some catchy phrases in pop songs of the same era. These similarities, even among works by different composers, generally do not constitute plagiarism; instead, they can usually quickly resonate with and be recognized by the listeners. Considering these similar segments from different pieces as distinct motifs is not accurate and will harm the performance of contrastive learning.

To address this issue, we propose to use a regularization-based training strategy for music motif representation learning. We design our method based on VICReg, a regularization-based training method in the field of computer vision [62]. The main idea is to train the model using only reliable information, which means using only positive sample pairs and discarding negative sample pairs. However, in the contrastive learning method, the function of negative pairs is not only providing weak labels for motif information, but also regularizing the representation space to prevent it from collapsing. If no negative samples are used in training, the loss function can easily converge to zero by mapping all inputs to the same output embedding. As a result, proper regularization methods need to be applied if only positive pairs are used during training.

VICReg resolves this problem by regularizing the latent space of expanded embeddings. First, a post-processing module, known as the expander, is attached to the encoder. The expander $Epd(\cdot)$ takes in the encoded embeddings $\mathbf{z} = Enc(\mathbf{X})$ and further projects them to the expanded embeddings $\mathbf{z}' = Epd(\mathbf{X})$ using a multi-layer perceptron (MLP). The model structure with the added expander can be viewed in Fig. 4.4.

The training and regularization are achieved simultaneously by three loss terms, each playing a different role in training:

- The Invariance loss. This term uses the mean square error (MSE) as a measurement of the distance between the expanded embeddings of the anchor and the positive sample. The closer the two expanded embeddings are, the lower the Invariance loss. It helps the model learn the information in the data labels by pushing positive pairs closer in the latent space. The Invariance loss of one training batch can be formulated as:

$$\mathscr{L}_{inv} = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{z}'_{\mathbf{i}} - \mathbf{z}'^{+}_{\mathbf{i}}||^2$$

  where n denotes the batch size, i denotes the sample index in a batch, $\mathbf{z}'_{\mathbf{i}}$ and $\mathbf{z}'^{+}_{\mathbf{i}}$ denote the ith anchor and the positive sample, respectively.

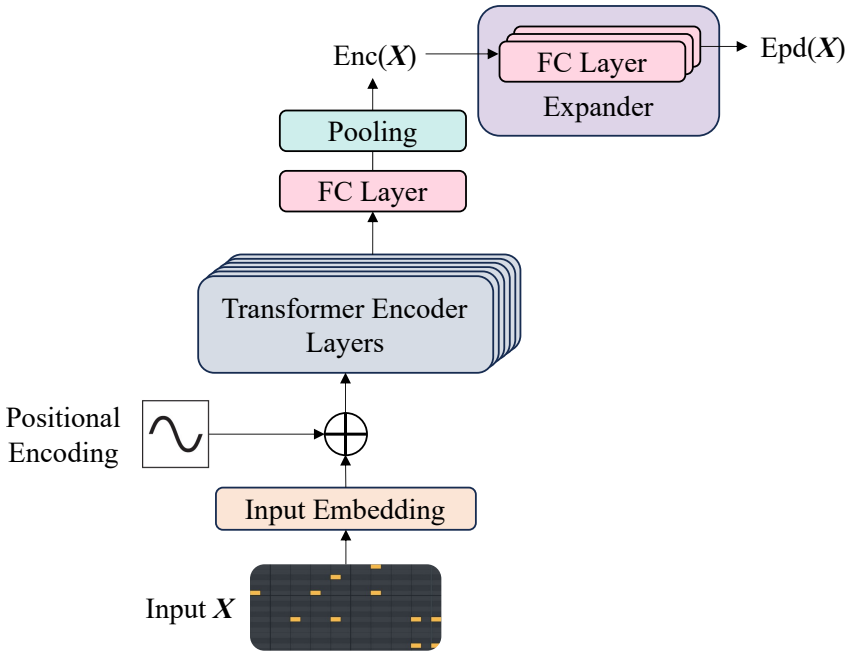

Figure 4.4: Model structure of the encoder and the expander

- The Variance loss. This loss term leverages the characteristics of neural network batch training and aims to maintain sufficient variance within each batch. It can effectively prevent the model from taking the shortcut that outputs the same value for every input. The standard deviation of every dimension is computed, and the loss term maintains the standard deviation values of every dimension using a hinge function. Both the anchor batch and the positive sample batch are considered in the computation. The Variance loss of one training batch can be formulated as:

$$\mathscr{L}_{var} = \frac{1}{2}(\frac{1}{d'}\sum_{j=1}^{d'}\max(0, 1 - \sqrt{var([\mathbf{z'_0}_j, \ldots, \mathbf{z'_n}_j]) + \epsilon}) + \frac{1}{d'}\sum_{j=1}^{d'}\max(0, 1 - \sqrt{var([\mathbf{z'^+_0}_j, \ldots, \mathbf{z'^+_n}_j]) + \epsilon}))$$

where d' denotes the dimension of the expanded embeddings, j denotes the dimension index, and $\mathbf{z'_i}_j$ denotes the jth dimension of $\mathbf{z'_i}$. $var(\cdot)$ computes the variance of a sequence, and ϵ is a small scalar used to prevent numerical instabilities.

- The Covariance loss. This term encourages the off-diagonal coefficients of the covariance matrix of the batched $\mathbf{z}'$ to be close to 0. As a result, it decorrelates the dimensions of the expanded embeddings, ensuring that the dimensions can encode as distinct information as possible. The decorrelation of the expanded embedding dimensions also has a decorrelation effect on the embedding dimensions. The covariance loss can be formulated as:

$$\mathscr{L}_{cov} = \frac{1}{d'}\sum_{i\neq j}[Cov([\mathbf{z'_0}, \mathbf{z'_1}, \ldots, \mathbf{z'_n}])]^2_{i,j} + \frac{1}{d'}\sum_{i\neq j}[Cov([\mathbf{z'^+_0}, \mathbf{z'^+_1}, \ldots, \mathbf{z'^+_n}])]^2_{i,j}$$

where $Cov([\mathbf{z'_0}, \mathbf{z'_1}, \ldots, \mathbf{z'_n}])$ is the covariance matrix of the batch output, which can be further formulated as:

$$Cov([\mathbf{z'_0}, \mathbf{z'_1}, \ldots, \mathbf{z'_n}]) = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{z'_i} - \bar{\mathbf{z}'})(\mathbf{z'_i} - \bar{\mathbf{z}'})^T$$

It should be noted that although the ultimate target of regularization is the latent embedding space, all the above loss terms are computed in the expanded embedding space. The presence of the expander acts as a non-linear projector so that decorrelating the expanded dimensions will reduce the dependencies among the embedding dimensions, which excels just decorrelating embedding dimensions. In our study, the expander comprises three fully connected layers, and the dimension of each layer is 512. The expander contains 600k trainable parameters. Even though the expander slightly increases the number of trainable parameters, it is only used for regularization and we still take the output of the encoder as the final learned representation.

The final training objective is a weighted sum of the Invariance loss, Variance loss and Covariance loss:

$$\mathscr{L} = \alpha \mathscr{L}_{inv} + \beta \mathscr{L}_{var} + \gamma \mathscr{L}_{cov}$$

The training procedure of one iteration is as shown in Fig.4.5.

4.3 Experiments

In this section, we evaluate the performances of the baselines and the proposed approach by conducting a retrieval-based experiment.

To better investigate the problem that contrastive learning might introduce false negative samples that might harm the model performance, we conduct two sets of control experiments for contrastive learning. In one set, negative samples are randomly selected from songs different from the anchor sample during training. In the other set, we attempt to improve the performance by applying a filtering process when selecting negative samples: if the played notes in the chosen negative sample have more than 50% overlap with those of the anchor sample, the negative sample will be rejected. During training, random sampling continues until a negative sample meeting the condition is found. We refer to this set of experiments as negative-enhanced contrastive learning.

In our experiments, we empirically set $\alpha = 25$, $\beta = 25$ and $\gamma = 1$. All models are trained with the AdamW optimizer [63], with a warm-up phase of 2000 steps, an initial learning rate of 1e-4, and a weight decay rate of 0.1. The dropout rate of the models is set to 0.1 during the training stage. All the models that require training reach convergence within 15 epochs.

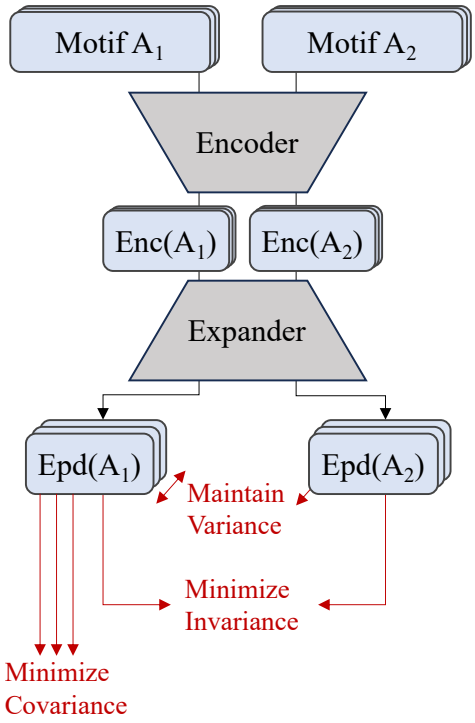

Figure 4.5: An example iteration of training the encoder using VICReg. A_1 and A_2 are occurrences of the same motif, constituting a positive pair. Motif objects are stacked to represent multiple samples in a batch. The encoded embeddings of A_1 and A_2 are further projected to the expanded embedding space, where the loss is computed to maintain variance within the batch, minimize invariance in a positive pair, and minimize the covariance of embedding dimensions.

Every data chunk in the validation set of $\mathbb{D}_{syn}$ is encoded with each of the trained encoders, and also the interval-based piano roll representation. In each representation space, we compute the Euclidean distances from an anchor data sample to every other sample in the validation set, and analyze the precision and recall rates of the nearest K retrievals. Precision reflects how many retrievals in the nearest K findings are really of the same motif as the anchor. Recall reflects how many congenetic samples are discovered in the K retrievals. The precision and recall are computed as follows:

$$P = \frac{TP}{TP + FP}$$
$$R = \frac{TP}{TP + FN}$$

where TP is the number of true positive samples, FP is the number of false positive samples, and FN is the number of false negative examples. We also compute the F1 score:

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

In an ideal representation space, every data sample should be close to the samples that are of the same

Table 4.1: Retrieval experiment result of different encoders at different retrieval numbers K. Reg: The proposed pure regularization-based training method, based on VICReg. Neg-enhanced CL: Negative-enhanced contrastive learning. CL: Contrastive learning method with direct negative sampling. Interval-based PR: Interval-based piano roll representation with all pitches shifted to start with the lowest.

	K=5			K=10			K=20			K=50			K=100		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Reg (Proposed)	**42.2**	**42.2**	**42.2**	**31.7**	**63.4**	**42.3**	**20.6**	**82.5**	**33.0**	**9.6**	**95.6**	**17.4**	**4.9**	**97.9**	**9.3**
Neg-enhanced CL	34.8	34.8	34.8	26.5	53.0	35.3	18.0	71.9	28.7	8.9	89.1	16.2	4.8	95.2	9.1
CL	35.4	35.4	35.4	26.9	53.9	35.9	18.2	72.9	29.2	9.0	90.4	16.4	4.8	96.3	9.2
Interval-based PR	36.6	36.6	36.6	27.2	54.3	36.2	17.3	69.4	27.7	7.9	78.5	14.3	4.0	80.4	7.7

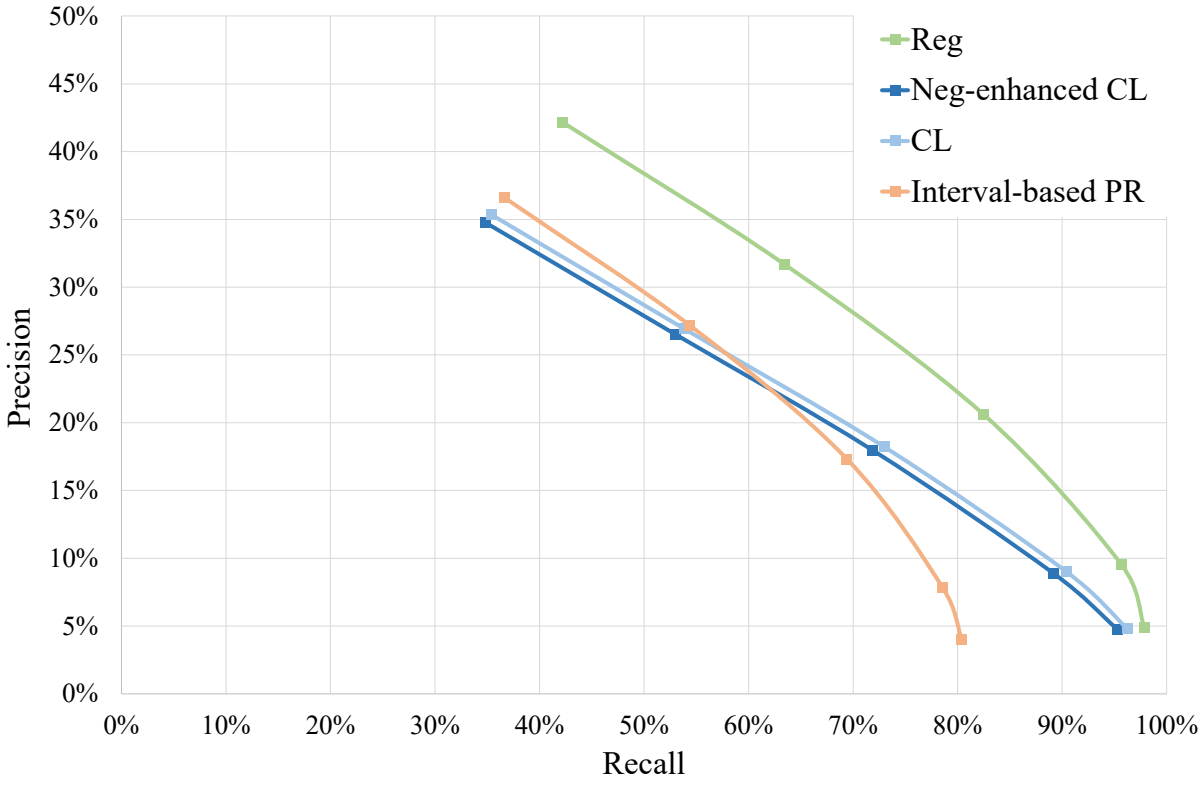

Figure 4.6: Precision-recall curves of the evaluated methods on the synthetic dataset

motif, hence the precision, recall and F1 should all be high.

We compute the mean precision, recall and F1 score of every anchor sample under the K value setting of 5, 10, 20, 50 and 100, respectively. The numbers are reported in Tab. 4.1. We also plot the precision-recall curve of every evaluated method in Fig.4.6. The closer the curve is to the top-right corner, the better the performance, indicating high precision and high recall are achieved simultaneously.

We have observed that the proposed method significantly outperforms baseline methods, especially when K is small, which shows that the proposed method can effectively improve model performance by replacing unreliable self-supervised sampling strategies with regularization approaches. Even if we

introduce negative-enhancement to contrastive learning, its performance still falls behind the proposed method, as false negative samples are still hard to eliminate through this method. Also, biased negative sampling might harm the model training by making the model converge faster to a local optimum.

Meanwhile, we also observe that at low values of K, the interval-based piano roll representation slightly outperforms the contrastive learning-based approaches, while at high values of K, it performs considerably worse than other methods. This indicates that representations based on direct geometric similarities do have certain advantages in retrieving highly similar targets due to their simplicity and interpretability, but they are not sufficient to handle cases where the geometric similarity is not as strong, and such cases are more common in the context of motif variations. In contrast, neural-network-based methods are more robust in capturing the inherent similarity that leads to the perception of motif repetitions, which is indicated by the fact that as K increases, the performance gap between contrastive learning and the proposed method gradually narrows, until at $K = 100$ where their performances become very close and both substantially surpass the piano roll-based baseline.

Through this retrieval experiment, we have preliminarily demonstrated the effectiveness and superiority of the proposed method in pretraining the music motif representation encoder. However, since both the pretraining and experiments are conducted on synthetic data, we can only conclude that the proposed method has better self-supervised training performance compared to contrastive learning and the direct piano roll representation. In the next chapter, we will explore the performance of these methods on real data.

Chapter 5

Transfer Learning on Real Data

5.1 Constructing the Real Dataset

Compared to other fields of music representation learning, there is a significant scarcity of effective datasets designed for music motifs, which to some extent, has constrained the progress of research in music motifs [64, 65].

In order to facilitate precise evaluation and validation of music motif encoders, a carefully curated hand-labeled dataset is assembled as an extension of the POP909 dataset. An additional track is added to the original POP909 MIDI files to store the labels of music motifs. The labels refer to the third track, which is the piano accompaniment track of songs. We use notes in the new label track to represent the motif labels. Accompaniment parts associated with the same motif are covered with notes of the same pitch. Accompaniment parts that are too ambiguous or occur only once are left unlabeled. Empty accompaniment parts are also unlabeled. The notes on the new track do not overlap with each other, which means that there is at most one label for each part of the accompaniment. The notes used to label the motifs start from C4 and progressively ascend in pitch by one semitone for each new motif that appears. An example data-label pair is shown in Fig. 5.1. It should be noted that although the label tracks of every song all start with C4, they convey different motif labels as they are notations of the first motif of different songs.

The hand-labeled real dataset comprises the accompaniment tracks and the label tracks of 80 POP909 songs, including the 20 songs for validation. We further process the raw data as follows. First, the tracks are all trimmed to start at the global onset of each song as annotated in POP909. Second, the tracks are all quantized with a resolution of 128th notes. Next, we segment both the accompaniment tracks and label tracks into 1-bar chunks. Due to the same global onset, the segmentation times for the accompaniment track and the label track of each song are also identical, and every 1-bar accompaniment chunk comes

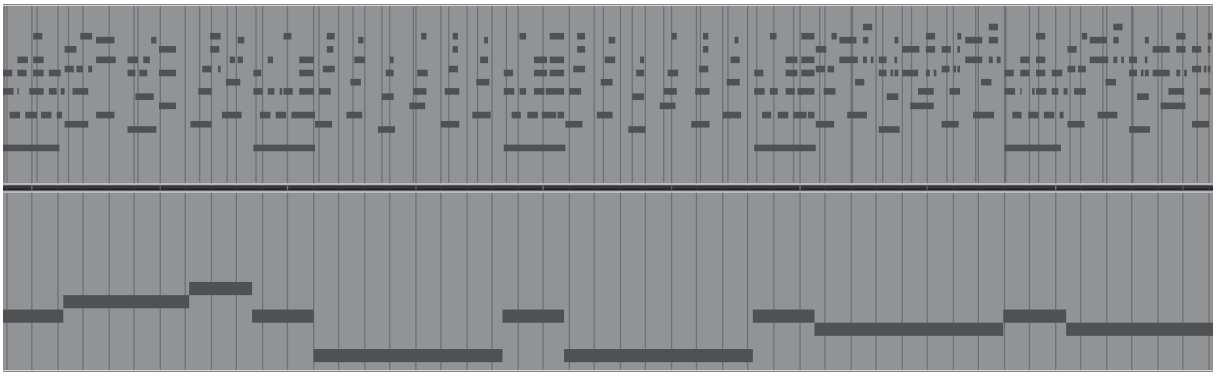

Figure 5.1: An example pair of the original data and its label in a visualized multitrack piano roll. The horizontal axis represents time, and the vertical axis represents pitch. The small rectangular bars denote note events. The upper row is the accompaniment track of a POP909 song, and the lower row is the added label track. Every note in the label track is a motif label, and notes with the same pitch are the same motif label.

with a 1-bar label chunk. Accompaniment chunks that have a label note covering over 75% of their length are finally labeled with that pitch. Chunks that do not meet the criteria are discarded. In this way, each data chunk is assigned a specific motif label. These labels indicate which chunks within the same song are associated with the same motif. In other words, chunks with the same labels and from the same song are different views of each other. The finalized dataset $\mathbb{D}_{real}$ has a mean number of motifs per song of 5.63, and a mean number of occurrences per motif of 7.58.

It is worth noting that due to the intrinsic ambiguity of music motifs, the perception difference of motifs among individuals and the limited number of annotators, the labels in this dataset are not guaranteed to be completely accurate. However, this dataset demonstrates that motifs can be annotated in a relatively concise manner, and the dataset is sufficient for validating and comparing the effectiveness of the methods.

5.2 Fine-tuning Schemes

The general objective of transfer learning is to leverage the knowledge learned from one task to improve the performance on a related but different task. In this study, we exploit the knowledge in the self-supervised pretrained models to improve the learning process on the real dataset. The pretrained models are fine-tuned on the real dataset to adapt its learned features for real labels. Typical ways of fine-tuning a neural network include further training all parameters in the pretrained model, and only training some layers close to the output while freezing the other layers. In our preliminary experiments, freezing the transformer encoder layers and training only the output fully connected layers make only slight changes to the performance. Therefore, no model layers are frozen in the experiments in this chapter.

In the fine-tuning stage, similar to the pretraining stage, we employ either contrastive learning or the regularization-based approach. As for the VICReg-based pretrained model, both the encoder and the expander are fine-tuned.

An intuitive way of fine-tuning the pretrained models on a target dataset is using the same training technique as is used in the pretraining stage to keep the learning process consistent. For example, the model pretrained by contrastive learning is also fine-tuned by selecting positive and negative samples for an anchor in the target dataset. From the conclusions drawn in the previous chapter, we can infer that using the proposed method for pretraining and fine-tuning should yield better results compared to using contrastive learning. However, we notice that, similar to contrastive learning, where false negative samples can harm the learning process, the VICReg-based approach also suffers from a problem. VICReg uses three terms for regularization, in which the variance term prevents the latent space from collapsing by maintaining the embedding difference among batch samples. However, there is also an underlying assumption that the samples in the batch are generally quite different, and should not be associated with the same source motif. This assumption holds when the dataset is large enough and it is unlikely to have occurrences of the same motif in a training batch. Since the dataset for fine-tuning is significantly smaller, there are likely samples from the same motif in a training batch. As a consequence, attempting to maintain the variance in the batch results in a larger distance between these samples, which counteracts the effect of the invariance term that reduces the distance.

To resolve this issue, we propose to combine the regularization-based method and contrastive learning by adopting contrastive learning to fine-tune the model pretrained by VICReg. We propose this fine-tuning method based on a hypothesis: when the dataset is small enough, the potential negative effects of incorrect negative sampling in contrastive learning are smaller than the negative effects of VICReg on maintaining variance among potential positive instances. The explanation for this hypothesis is as follows. The overall distribution of the motif representations of all existing music can be seen as approximately a Gaussian distribution. The corresponding embedding set of the real dataset can be viewed as a set of samples from the Gaussian. Therefore in contrastive learning, the expectation of the proportion of extracted samples with the same motifs as the anchors is uncorrelated with the dataset size. However, as for the VICReg method, the expectation of the proportion of positive pairs negatively impacted by the variance term decreases as the dataset size increases. Since the dataset is relatively small, contrastive learning should be preferable to VICReg as a training strategy.

Table 5.1: Retrieval experiment result of different encoders at different retrieval numbers K. Neural network-based encoders are pretrained with self-supervised learning and evaluated on the real dataset.

	K=5			K=10			K=20			K=50			K=100		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Reg (Proposed)	**37.3**	**19.6**	**20.9**	**30.8**	**28.2**	**24.2**	**24.5**	**38.9**	**25.6**	**16.5**	**58.1**	**23.4**	**10.6**	**71.5**	**17.5**
Neg-enhanced CL	34.4	18.2	19.2	27.3	24.3	21.1	21.5	33.6	22.3	14.5	51.0	20.6	9.7	65.5	16.0
CL	33.3	17.3	18.5	27.9	24.6	21.4	22.2	34.6	23.0	14.9	51.2	21.1	9.9	66.0	16.4
Interval-based PR	28.4	14.7	15.3	19.1	17.9	14.8	13.1	22.6	13.8	8.1	31.9	11.6	5.4	40.7	9.0

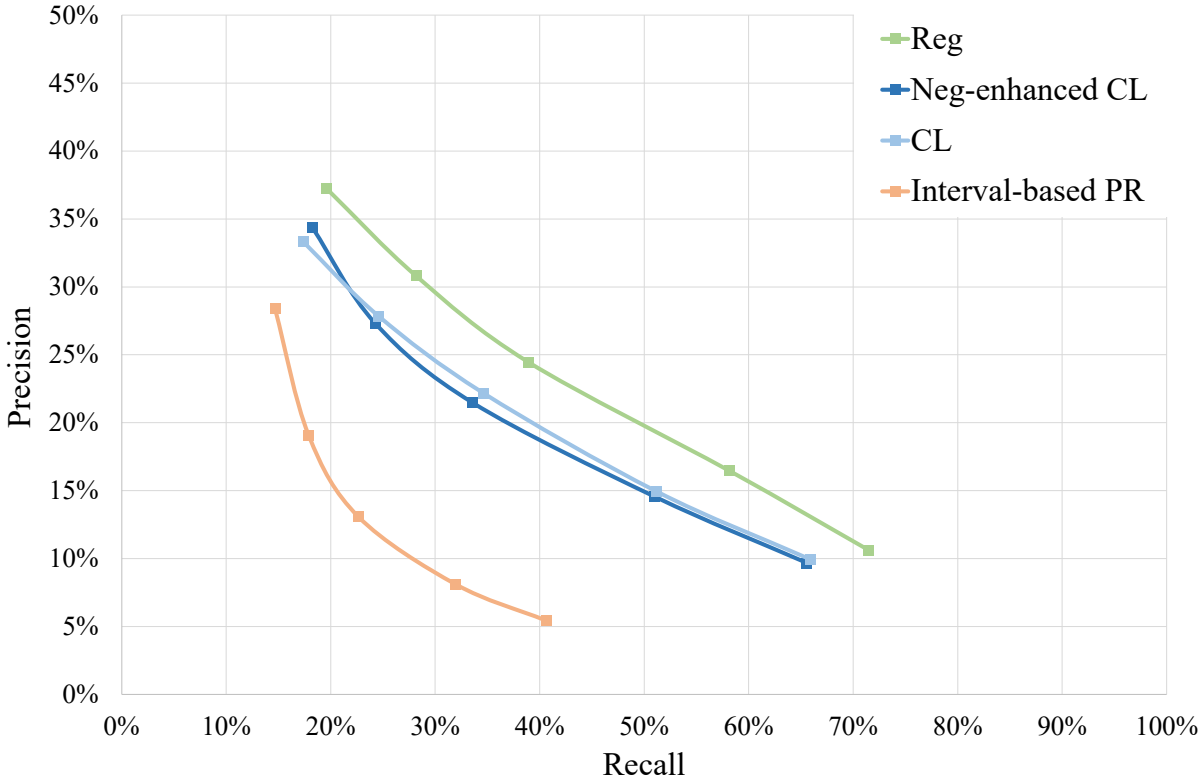

Figure 5.2: Precision-recall curves of the evaluated methods on the real dataset. Models are pretrained with self-supervised learning without fine-tuning.

5.3 Experiments

We evaluate the proposed fine-tuning method and verify the hypothesis by conducting retrieval-based experiments similar to the last chapter. First, before testing the performances of fine-tuned models, we evaluate the retrieval performances of the pretrained models on the real dataset. Every data chunk in the validation set of $\mathbb{D}_{real}$ is encoded with each of the pretrained encoders and the interval-based piano roll representation. We compute the Euclidean distances from every anchor to every other sample, and count the mean precision and recall values in the nearest K samples. The results are reported in Tab. 5.1, and the precision-recall curves are plotted in Fig. 5.2.

Table 5.2: Retrieval experiment result of different encoders at different retrieval numbers K. Neural network-based encoders are trained from scratch on the real dataset.

	K=5			K=10			K=20			K=50			K=100		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Reg	19.6	9.0	10.3	15.4	12.8	11.8	11.8	18.1	12.5	8.6	30.7	12.4	6.5	45.1	10.9
Neg-enhanced CL	**30.7**	**14.8**	**16.4**	**24.1**	**21.4**	**18.6**	**19.0**	**30.0**	**20.0**	**13.2**	**47.6**	**19.0**	**9.2**	**62.8**	**15.3**
CL	29.5	13.4	15.4	23.2	19.8	17.7	18.0	27.7	18.8	12.4	44.4	17.7	8.9	61.4	14.8
Interval-based PR	28.4	14.7	15.3	19.1	17.9	14.8	13.1	22.6	13.8	8.1	31.9	11.6	5.4	40.7	9.0

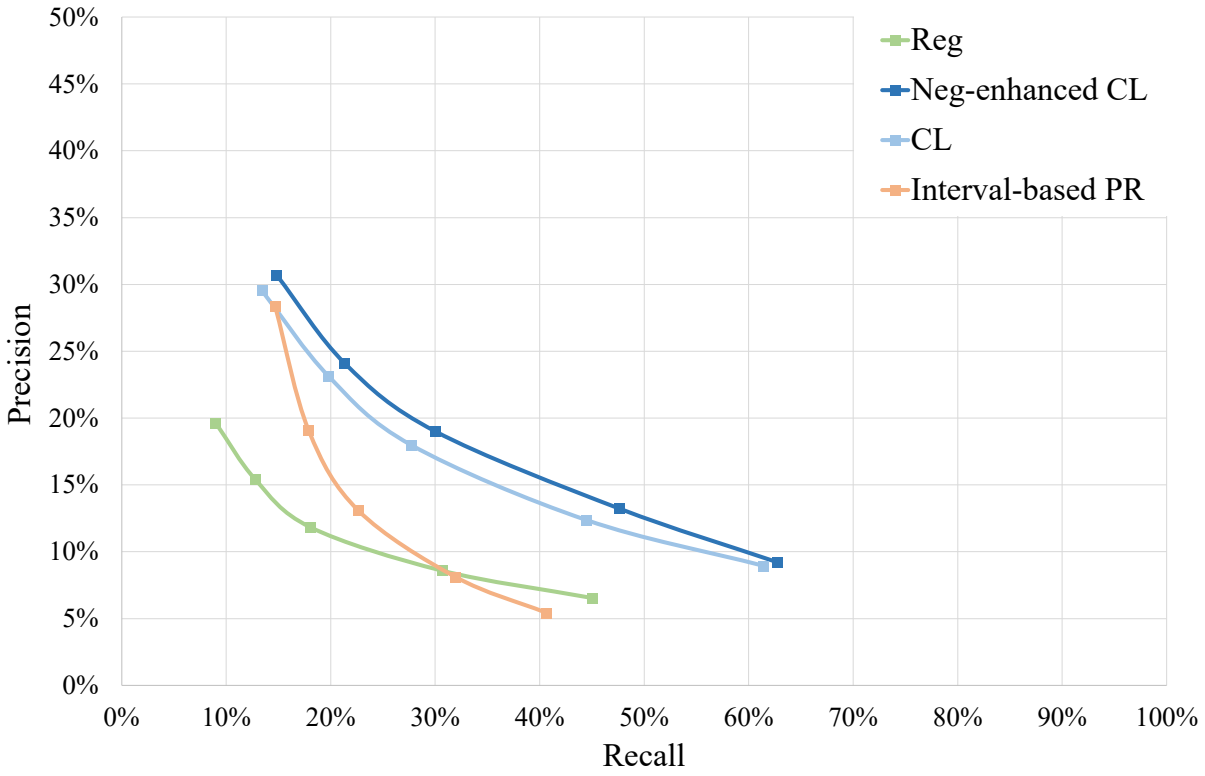

Figure 5.3: Precision-recall curves of the evaluated methods trained from scratch on the real dataset. Self-supervised pretraining is not applied to any neural network-based approach here.

Although there's a significant performance drop when changing the evaluation dataset from $\mathbb{D}_{syn}$ to $\mathbb{D}_{real}$, the proposed training method still outperforms all baselines by a clear margin. This fact shows that the generalizability of the learned representations can be substantially improved by adopting the proposed regularization-based method. Also, the interval-based piano roll representation falls far behind neural-network-based approaches, demonstrating its drawback in generalizability when transformations of motif occurrences in real data are a lot more complex than being geometrically similar.

Second, to preliminarily evaluate the effectiveness of using pretraining for music motif representation learning, we also conduct the same retrieval experiment on the models trained only on the real dataset $\mathbb{D}_{real}$, while the pretraining stage is discarded. The hyperparameters are kept the same as in the previous

pretraining experiment. The models trained with two contrastive learning methods both converge in 400 epochs, while the model trained with the regularization-based method converges in only 50 epochs. The models are then evaluated on the validation set of $\mathbb{D}_{real}$. The results are reported in Tab. 5.2, and the precision-recall curves are plotted in Fig. 5.3.

We observe from comparing Fig. 5.2 and Fig. 5.3 that the performances of pretrained models are remarkably better than the performances of models trained directly on the real dataset. This fact preliminarily demonstrates the effectiveness of pretraining for learning music motif representations, addressing the issue of small datasets. However, we also notice that the VICReg-based approach performs poorly compared to other methods when training directly on the real dataset. This gives evidence to the observed issue of VICReg that optimizing the variance term can seriously harm the learning process when the dataset is small, as more positive pairs tend to appear in a single batch. Another observation is that unlike in the pretraining experiment, the trick of applying negative-enhancement to contrastive learning does have a noticeable improvement on the model performance. A possible explanation is that the limited size of data raises the necessity of choosing negative samples more carefully.

Finally, we carry out the fine-tuning experiment. We fine-tune both the contrastive learning pretrained model and the VICReg pretrained model using the same methods as in their pretraining stage. We also experiment with the proposed fine-tuning scheme, which is fine-tuning the VICReg pretrained model with contrastive learning. The learning rates are reduced to 1e-5 to prevent steep learning steps, and all other hyperparameters are unchanged. In our experiments, the model fine-tuned with the VICReg approach reaches convergence within 150 epochs, while models fine-tuned with contrastive learning converge within 500 epochs. The models are then evaluated on the validation set by the mean precision and recall values in the nearest K samples of every anchor sample. The result numbers are shown in Tab. 5.3. Fig. 5.4 demonstrates the precision-recall curves of the retrieval experiments.

The overall performance of fine-tuned models shows obvious improvements compared to the models only pretrained and the models trained only on the real dataset, confirming that pretraining by self-supervised learning before training on real data is an effective way of learning music motif representations. As for the fine-tuning schemes, VICReg loses its advantage to contrastive learning when also fine-tuned with VICReg. This provides further evidence for the hypothesis that the variance term does harm the performance when data is scarce. However, by fine-tuning the VICReg pretrained model with contrastive learning, the model remarkably surpasses other models on retrieval performance. This outcome supports the hypothesis that the problem of negative sampling in contrastive learning is not as severe as the problem of variance in VICReg when the dataset is small, and that we can make use of their respective advantages to optimize a motif representation encoder.

Table 5.3: Retrieval experiment result of different encoders at different retrieval numbers K. Neural network-based encoders are pretrained with self-supervised learning and fine-tuned on the real dataset. CL → CL: Encoder pretrained and fine-tuned both with contrastive learning; Reg → Reg: Encoder pretrained and fine-tuned both with the VICReg approach. Reg → CL: Encoder pretrained with the VICReg approach, but fine-tuned with contrastive learning.

	K=5			K=10			K=20			K=50			K=100		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
CL → CL	40.1	18.9	21.1	33.2	27.8	25.1	26.7	40.6	27.6	17.4	60.1	24.6	11.4	75.7	18.8
Reg → Reg	38.7	20.5	21.5	32.2	29.6	25.2	26.2	41.7	27.4	16.9	59.8	24.0	10.7	72.5	17.6
Reg → CL	**42.9**	**21.4**	**23.2**	**36.0**	**31.1**	**27.5**	**29.0**	**44.3**	**30.0**	**18.8**	**64.7**	**26.6**	**11.5**	**77.7**	**19.0**
Interval-based PR	28.4	14.7	15.3	19.1	17.9	14.8	13.1	22.6	13.8	8.1	31.9	11.6	5.4	40.7	9.0

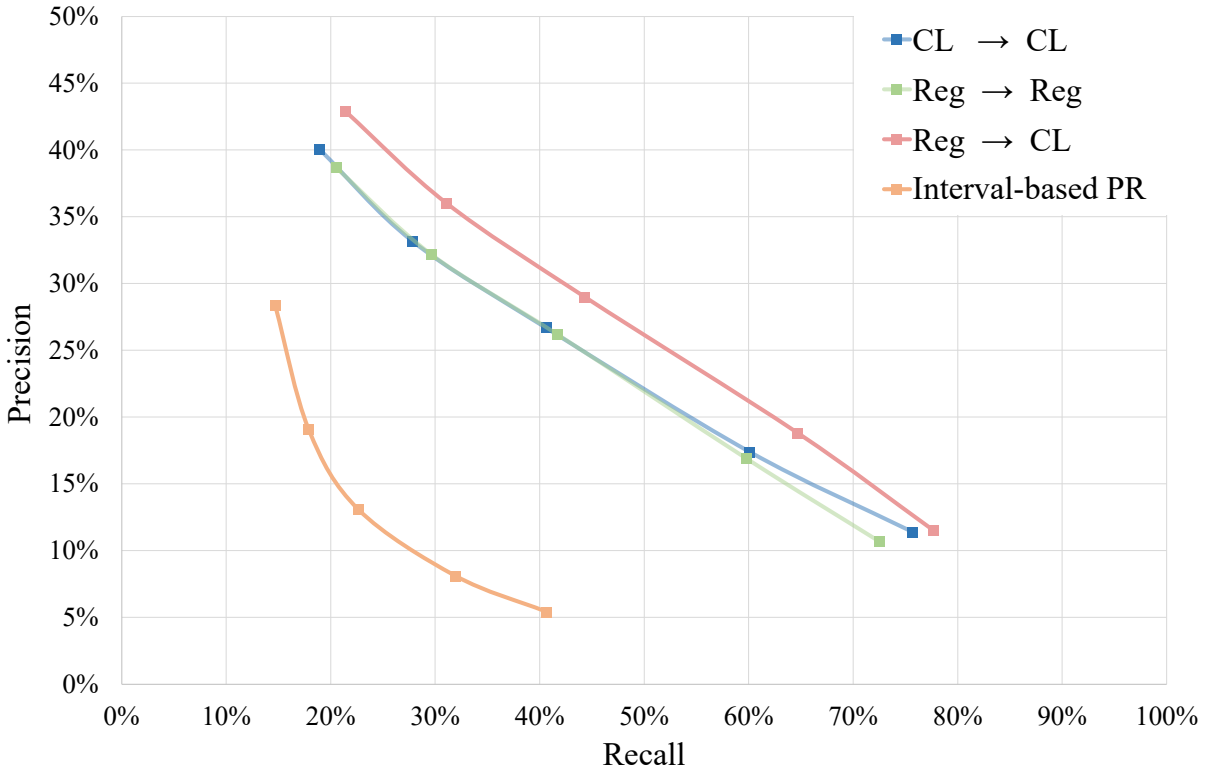

Figure 5.4: Precision-recall curves of the evaluated transfer learning methods on the real dataset.

So far, through experiments on real data, we have demonstrated the effectiveness of the pretraining-fine-tuning strategy for learning music motif representations. The experiments also provide further support for the assertion made earlier that the proposed VICReg-based training approach outperforms contrastive learning in terms of generalization. Simultaneously, we have also highlighted the limitations of the VICReg method on small datasets. As a remedy, we propose utilizing VICReg for self-supervised pretraining and employing contrastive learning for finetuning. This allows us to leverage the strengths of both training strategies while mitigating their weaknesses.

Chapter 6

Motif-based Music Structure Visualization

Music motif, as a feature of music revealing the deep dependencies in the music context, is of special importance in music composition and analysis. The deep representations of music hold valuable insights for automatic music composition and analysis. In this chapter, we showcase the potential usage of learned music motif representations by visualizing the music structure of pop piano accompaniments from the perspective of motifs. We selected the best-performing model from the previous experiments, namely the transformer encoder pretrained with VICReg self-supervision and fine-tuned on real data with contrastive loss.

We first parse the target accompaniment in MIDI format to the piano roll representation, then segment the piano roll into 1-bar chunks. These chunks have the same dimensionality as the samples in the synthetic dataset and the real dataset, so we can feed them as inputs to the trained encoder and obtain their embeddings. These embeddings are points in the trained motif latent space, and points with smaller distances from each other are likely to be associated with the same motif. We adopt clustering to classify the encoded points in the latent space. Points in the same cluster are considered to be associated with the same motif. Generally, clustering methods can be categorized into partition-based methods, density-based methods and hierarchical methods. Although partition-based methods such as K-means are more commonly used in many scenarios, we select DBSCAN, a classical density-based clustering method [66]. DBSCAN groups data points based on their density within a specified radius ϵ and a minimum number of points n_{min}. Clusters are initially constructed from identified core points with sufficient neighboring points, and expanded by absorbing new points within the distance of ϵ. We choose DBSCAN for the following reasons. (1) DBSCAN doesn't require specifying the class number in advance. This advantage meets the need for motif analysis since the number of involved motifs of a music piece is unknown. (2) DBSCAN is capable of identifying noisy samples among meaningful clusters. Many music segments, such

as transitional parts and some improvisation parts, may not represent any motif, or they may appear only once in the entire composition to highlight their uniqueness. We empirically set ϵ to 3 and n_{min} to 6.

Taking the accompaniment track of a POP909 song in the validation set as an example, we first visualize the distribution of music chunks using their learned embeddings. Due to the high dimensionality of the embeddings, we use the t-SNE dimensionality reduction technique to project the embeddings to a 2-dimensional space [67]. Since there are usually identical music chunks in the piece, we add small Gaussian noises to those embeddings to make them distinguishable. The embeddings and their clustering labels are visualized as colored scatters as shown in Fig. 6.1. It can be observed that chunks that associate with the same motif are close to each other in the embedding space.

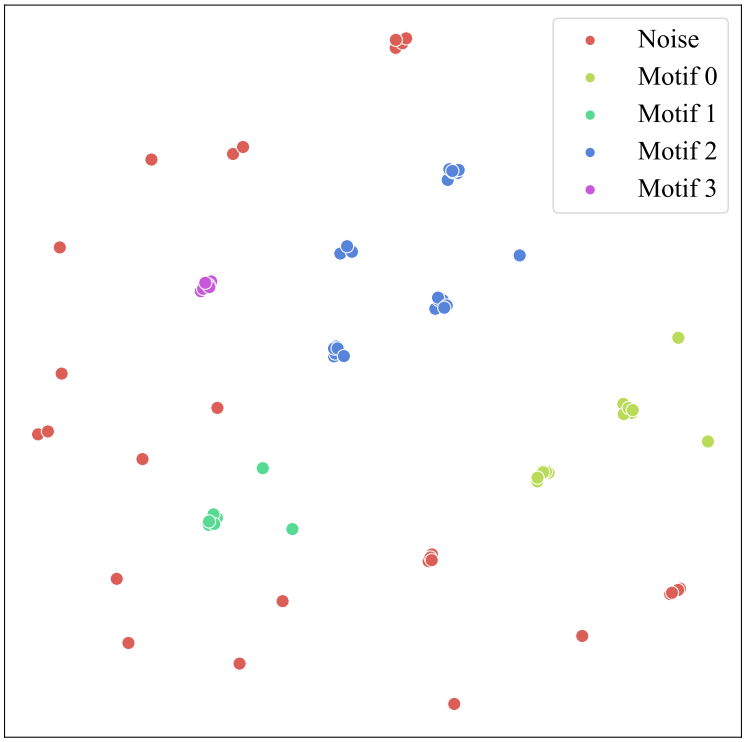

Figure 6.1: An example 2-D scatter plot of chunks in a song. Each dot represents the embedding of a music chunk. Chunks that associate with the same motif according to the clustering results have the same color.

We then visualize the temporal structure of music from the perspective of motifs. We compute the embedding values of every cluster center, and compute the distances from every chunk embedding to every cluster center. Two different ways of visualization are adopted to obtain a comprehensive analysis of the structure. First, we plot a heat map of the distance matrix to reveal the distribution of every motif over time, as shown in Fig. 6.2. Second, we plot the tendency curves of the motifs with the distance matrix, as shown in Fig. 6.3. The distances from cluster centers are projected to their negative exponential to represent the level of presence, and the curves are smoothed with interpolation for a clearer view of the emergence

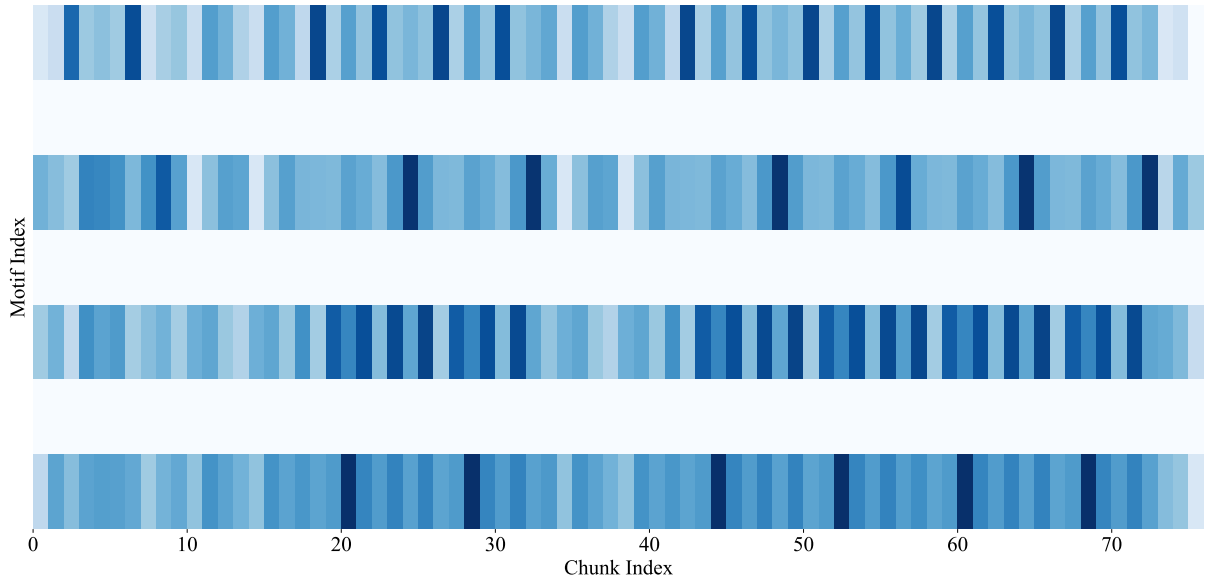

Figure 6.2: An example heat map showing the distributions of motifs over time. Every row is the distribution of a motif. The horizontal axis represents time as measured by chunk index. The deeper the color, the stronger the presence of a motif at a certain time.

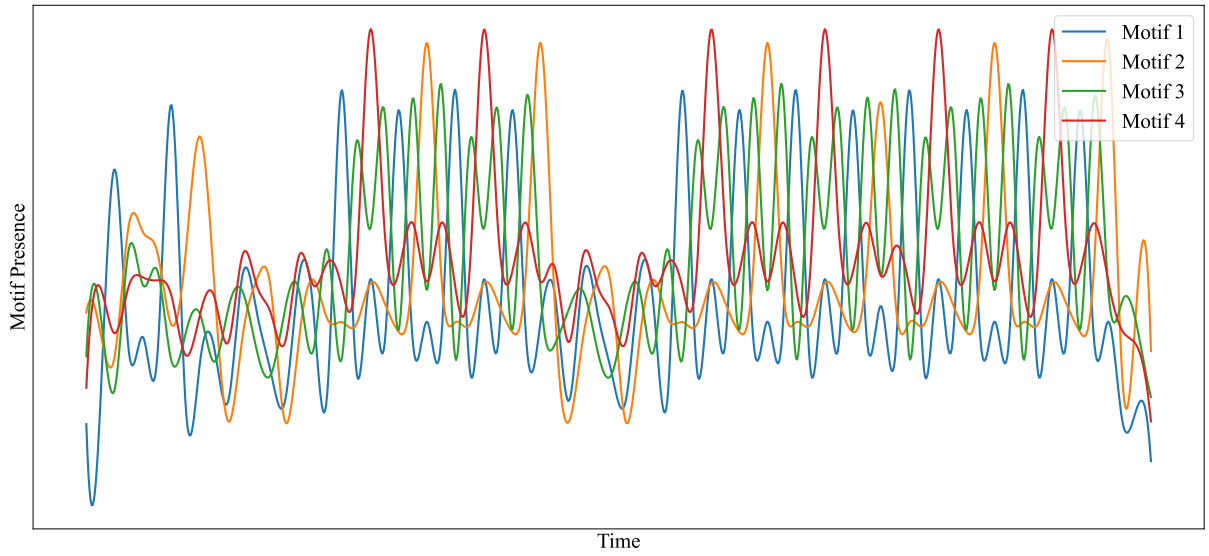

Figure 6.3: The tendency curves of an example piece. Every curve represents the presence change of a motif over time. The higher the curve, the stronger the presence of a motif at a certain time.

and fading of motifs.

It can be observed from Fig. 6.2 that every motif has its unique pattern of distribution. Some tend to emerge periodically, possibly as certain transition motifs that play their role in every phase. Some tend to appear intensively at certain parts of the song. These are likely unique motifs carried by the chorus, appearing very rarely in the verse but shining prominently in the chorus. From Fig. 6.3, we can observe

clearly that the intro and outro are distinct from the motifs carried in the main part of the song, and also that the presence curves naturally form "blocks" that clearly outline the high-level structure of music.

It should be noted that we observed varying performances of the clustering algorithm, which visualization relies upon, across different songs. Finding clustering hyperparameters that work well for all songs is a challenging problem. Therefore, the examples of visualization serve mainly as an initial exploration, and further research is warranted on how to use the encoded motifs to robustly conduct structure analysis of music under a unified setting.

Chapter 7

Conclusions

In this study, we propose to address the problem of motif-based music representation learning using pretraining and fine-tuning methods. We introduce the use of a pure regularization-based approach in the pretraining phase and compare it with contrastive learning. Experiment results demonstrate that the proposed method exhibits significant improvements in pretraining performance compared to the baselines. By discussing the strengths and weaknesses of both training methods at different training stages, we propose a strategic combination of them in the pretraining and fine-tuning phases. Experiment results on a manually annotated music motif dataset show the superiority of the proposed method in learning representations of music motifs. We further demonstrate the potential application of the learned music motif representations through visualizations of music structure.

While this study has provided insights for effective representation learning of music motifs, it is important to acknowledge some limitations that should be considered when interpreting the results. These limitations also offer a path to future topics to promote research progress in this field. Firstly, the scope of this study is limited to piano accompaniments, hence we have used bars as a simply substitute for motif occurrences. This assumption appears arbitrary for many kinds of music, thus prompting the need to explore more advanced semantic segmentation methods for music motifs and integrate them with representation learning. Secondly, there is an underlying assumption in this study that the music semantics expressed in motifs are shared across different compositions, similar to the semantics of natural language. However, music semantics is more context-dependent than language semantics, as motifs are perceptually identified through dependencies in the context, such as repetitions. There is a necessity for further discussion on music semantics and syntax, and representation learning methods that better align with the characteristics of music. Lastly, the hand-labeled dataset introduced in this study is still relatively small, limiting the effectiveness of direct use of the learned representations.

Nevertheless, effective deep representations of music motifs can serve as the cornerstone for many downstream tasks in automatic music composition and analysis, as motif is often the entry point of inspiration in composition. For example, exploring the utilization of music motifs and their distributions as conditions for music generation algorithms could be valuable to control the semantics and structure of the generated music. Also, the study of music motifs could broaden the horizons of music information retrieval and amplify its applicability. Effective searching based on music motifs could potentially empower artists with better understanding and evaluation of their music, and assist music lovers in identifying songs with catchy phrases. We hope this study can pave the way for further research on deep learning applications on music motifs, expanding the capabilities of deep learning and enriching people's lives with the companionship of music.

Bibliography

[1] Jouni Paulus, Meinard Müller, and Anssi Klapuri. State of the art report: Audio-based music structure analysis. 2010.

[2] Arnold Schoenberg. *The musical idea and the logic, technique, and art of its presentation*. Indiana University Press, 2006.

[3] Leonard Bernstein. *The unanswered question: Six talks at Harvard*, volume 33. Harvard University Press, 1976.

[4] Klaus Frieler and Daniel Müllensiefen. The simile algorithm for melodic similarity. *Proceedings of the Annual Music Information Retrieval Evaluation exchange*, 3:59, 2005.

[5] Klaus Frieler. Generalized n-gram measures for melodic similarity. In *Data science and classification*, pages 289–298. Springer, 2006.

[6] Iman SH Suyoto and Alexandra L Uitdenbogerd. Simple orthogonal pitch with ioi symbolic music matching. *Proceedings of the Annual Music Information Retrieval Evaluation exchange*, 2010.

[7] Naresh N Vempala and Frank A Russo. An empirically derived measure of melodic similarity. *Journal of New Music Research*, 44(4):391–404, 2015.

[8] Valerio Velardo, Mauro Vallati, and Steven Jan. Symbolic melodic similarity: State of the art and future challenges. *Computer Music Journal*, 40(2):70–83, 2016.

[9] Berit Janssen, Peter Van Kranenburg, and Anja Volk. Finding occurrences of melodic segments in folk songs employing symbolic similarity measures. *Journal of New Music Research*, 46(2):118–134, 2017.

[10] Ziyu Wang, Dingsu Wang, Yixiao Zhang, and Gus Xia. Learning interpretable representation for controllable polyphonic music generation. In *Proceedings of 21st International Conference on Music Information Retrieval (ISMIR)*, 2020.

[11] Ziyu Wang, Yiyi Zhang, Yixiao Zhang, Junyan Jiang, Ruihan Yang, Junbo Zhao, and Gus Xia. Pianotree vae: Structured representation learning for polyphonic music. In *Proceedings of 21st International Conference on Music Information Retrieval (ISMIR)*, 2020.

[12] Tom Collins, Andreas Arzt, Sebastian Flossmann, and Gerhard Widmer. Siarct-cfp: Improving precision and the discovery of inexact musical patterns in point-set representations. In *International Society for Music Information Retrieval Conference*, 2013.

[13] Darrell Conklin. Mining contour sequences for significant closed patterns. *Journal of Mathematics and Music*, 15(2):112–124, 2021.

[14] Otso Björklund. Siatec-c: Computationally efficient repeated pattern discovery in polyphonic music. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022.

[15] Olivier Lartillot and Petri Toiviainen. Motivic matching strategies for automated pattern extraction. *Musicae Scientiae*, 11(1_suppl):281–314, 2007.

[16] Olivier Lartillot. Automated motivic analysis: An exhaustive approach based on closed and cyclic pattern mining in multidimensional parametric spaces. In *Computational Music Analysis*, pages 273–302. Springer, 2015.

[17] Alberto Pinto. Relational motif discovery via graph spectral ranking. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, pages 102–109, 2010.

[18] Olivier Lartillot. In-depth motivic analysis based on multiparametric closed pattern and cyclic sequence mining. In *International Symposium on Music Information Retrieval: ISMIR*, 2014.

[19] Riyadh Benammar, Christine Largeron, Véronique Eglin, and Mylène Pardoen. Discovering motifs with variants in music databases. In *Advances in Intelligent Data Analysis XVI: 16th International Symposium, IDA 2017, London, UK, October 26–28, 2017, Proceedings 16*, pages 14–26. Springer, 2017.

[20] I-Chieh Wei, Chih-Wei Wu, and Li Su. Generating structured drum pattern using variational autoencoder and self-similarity matrix. In *ISMIR*, pages 847–854, 2019.

[21] Junyan Jiang, Ke Chen, Wei Li, and Gus Xia. Large-vocabulary chord transcription via chord structure decomposition. In *ISMIR*, pages 644–651, 2019.

[22] Stefan Lattner and Maarten Grachten. High-level control of drum track generation using learned patterns of rhythmic interaction. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 35–39. IEEE, 2019.

[23] Shuqi Dai, Zeyu Jin, Celso Gomes, and Roger B Dannenberg. Controllable deep melody generation via hierarchical music structure representation. In *Proceedings of 22st International Conference on Music Information Retrieval (ISMIR)*, 2021.

[24] Yi-Jen Shih, Shih-Lun Wu, Frank Zalkow, Meinard Muller, and Yi-Hsuan Yang. Theme transformer: Symbolic music generation with theme-conditioned transformer. *IEEE Transactions on Multimedia*, 2022.

[25] Ziyu Wang, Ke Chen, Junyan Jiang, Yiyi Zhang, Maoran Xu, Shuqi Dai, Xianbin Gu, and Gus Xia. Pop909: A pop-song dataset for music arrangement generation. In *Proceedings of 21st International Conference on Music Information Retrieval (ISMIR)*, 2020.

[26] Roger B Dannenberg, William P Birmingham, Bryan Pardo, Ning Hu, Colin Meek, and George Tzanetakis. A comparative evaluation of search techniques for query-by-humming using the musart testbed. *Journal of the American Society for Information Science and Technology*, 58(5):687–701, 2007.

[27] Saebyul Park, Taegyun Kwon, Jongpil Lee, Jeounghoon Kim, and Juhan Nam. A cross-scape plot representation for visualizing symbolic melodic similarity. In *ISMIR*, pages 423–430, 2019.

[28] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[29] Tatsunori Hirai and Shun Sawada. Melody2vec: Distributed representations of melodic phrases based on melody segmentation. *Journal of Information Processing*, 27:278–286, 2019.

[30] Folgert Karsdorp, Peter van Kranenburg, and Enrique Manjavacas. Learning similarity metrics for melody retrieval. In *ISMIR*, pages 478–485, 2019.

[31] Dave Meredith, Geraint A Wiggins, and Kjell Lemström. Pattern induction and matching in polyphonic music and other multidimensional datasets. In *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI2001)*, volume 10, pages 61–66, 2001.

[32] David Meredith, Kjell Lemström, and Geraint A Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321–345, 2002.

[33] David Meredith. Cosiatec and siateccompress: Pattern discovery by geometric compression. In *International society for music information retrieval conference*. International Society for Music Information Retrieval, 2013.

[34] Thomas Edward Collins. *Improved methods for pattern discovery in music, with applications in automated stylistic composition*. Open University (United Kingdom), 2011.

[35] Oriol Nieto and Morwaread Mary Farbood. Perceptual evaluation of automatically extracted musical motives. In *Proceedings of the 12th International Conference on Music Perception and Cognition*, pages 723–727, 2012.

[36] Fred Lerdahl and Ray S Jackendoff. *A Generative Theory of Tonal Music, reissue, with a new preface*. MIT press, 1996.

[37] Matevž Pesek, Aleš Leonardis, and Matija Marolt. Symchm—an unsupervised approach for pattern discovery in symbolic music with a compositional hierarchical model. *Applied sciences*, 7(11):1135, 2017.

[38] Iris Ren, Anja Volk, Wouter Swierstra, and Remco C Veltkamp. A computational evaluation of musical pattern discovery algorithms. *arXiv preprint arXiv:2010.12325*, 2020.

[39] Filip Korzeniowski and Gerhard Widmer. A fully convolutional deep auditory model for musical chord recognition. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2016.

[40] Brian McFee and Juan Pablo Bello. Structured training for large-vocabulary chord recognition. In *ISMIR*, pages 188–194, 2017.

[41] Tsung-Ping Chen and Li Su. Attend to chords: Improving harmonic analysis of symbolic music using transformer-based models. *Transactions of the International Society for Music Information Retrieval*, 4(1), 2021.

[42] Chen Li, Yu Li, Hui Song, and Lihua Tian. Deep semi-supervised learning with contrastive learning in large vocabulary automatic chord recognition. In *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1065–1069, 2023.

[43] Ruihan Yang, Dingsu Wang, Ziyu Wang, Tianyao Chen, Junyan Jiang, and Gus Xia. Deep music analogy via latent representation disentanglement. 2019.

[44] Yi Zou, Pei Zou, Yi Zhao, Kaixiang Zhang, Ran Zhang, and Xiaorui Wang. Melons: generating melody with long-term structure using transformers and structure graph. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 191–195. IEEE, 2022.

[45] Shuqi Dai, Huiran Yu, and Roger B Dannenberg. What is missing in deep music generation? a study of repetition and structure in popular music. In *Proceedings of 23st International Conference on Music Information Retrieval (ISMIR)*, 2022.

[46] Junyan Jiang, Gus G Xia, Dave B Carlton, Chris N Anderson, and Ryan H Miyakawa. Transformer vae: A hierarchical model for structure-aware and interpretable music representation learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 516–520. IEEE, 2020.

[47] Shiqi Wei, Gus Xia, Yixiao Zhang, Liwei Lin, and Weiguo Gao. Music phrase inpainting using long-term representation and contrastive loss. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 186–190. IEEE, 2022.

[48] Xueyao Zhang, Jinchao Zhang, Yao Qiu, Li Wang, and Jie Zhou. Structure-enhanced pop music generation via harmony-aware learning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1204–1213, 2022.

[49] Botao Yu, Peiling Lu, Rui Wang, Wei Hu, Xu Tan, Wei Ye, Shikun Zhang, Tao Qin, and Tie-Yan Liu. Museformer: Transformer with fine-and coarse-grained attention for music generation. *Advances in Neural Information Processing Systems*, 35:1376–1388, 2022.

[50] Xingjian Du, Zhesong Yu, Bilei Zhu, Xiaoou Chen, and Zejun Ma. Bytecover: Cover song identification via multi-loss training. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 551–555. IEEE, 2021.

[51] Ken O'Hanlon, Emmanouil Benetos, and Simon Dixon. Detecting cover songs with pitch class key-invariant networks. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2021.

[52] Xingjian Du, Ke Chen, Zijie Wang, Bilei Zhu, and Zejun Ma. Bytecover2: Towards dimensionality reduction of latent embedding for efficient cover song identification. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 616–620. IEEE, 2022.

[53] Xingjian Du, Zijie Wang, Xia Liang, Huidong Liang, Bilei Zhu, and Zejun Ma. Bytecover3: Accurate cover song identification on short queries. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[54] Ahmet Elbir, Hilmi Bilal Çam, Mehmet Emre Iyican, Berkay Öztürk, and Nizamettin Aydin. Music genre classification and recommendation by using machine learning techniques. In *2018 Innovations in intelligent systems and applications conference (ASYU)*, pages 1–5. IEEE, 2018.

[55] Alexander AS Gunawan, Derwin Suhartono, et al. Music recommender system based on genre using convolutional recurrent neural networks. *Procedia Computer Science*, 157:99–109, 2019.

[56] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[58] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[59] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[60] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[61] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[62] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *10th International Conference on Learning Representations, ICLR 2022*, 2022.

[63] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

[64] Tom Collins. Mirex 2017: Discovery of repeated themes & sections. `https://www.music-ir.org/mirex/wiki/2017:Discovery_of_Repeated_Themes_%26_Sections`, 2017. Last accessed on August 5th, 2023.

[65] Peter van Kranenburg, Berit Janssen, and Anja Volk. The meertens tune collections: The annotated corpus (mtc-ann) versions 1.1 and 2.0. 1. *Meertens Online Reports*, 2016(1), 2016.

[66] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining*, volume 96, pages 226–231, 1996.

[67] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.