WHOLE-SONG HIERARCHICAL GENERATION OF SYMBOLIC MUSIC USING CASCADED DIFFUSION MODELS

Ziyu Wang[12], **Lejun Min**[2], **Gus Xia**[12]
[1]Computer Science Department, NYU Shanghai, [2]Machine Learning Dapartment, MBZUAI
`ziyu.wang@nyu.edu, {lejun.min, gus.xia}@mbzuai.ac.ae`

ABSTRACT

Recent deep music generation studies have put much emphasis on *music structure* and *long-term* generation. However, we are yet to see high-quality, well-structured whole-song generation. In this paper, we make the first attempt to model a full music piece under the realization of *compositional hierarchy*. With a focus on symbolic representations of pop songs, we define a hierarchical language, in which each level of hierarchy focuses on the context dependency at a certain music scope. The high-level languages reveal whole-song form, phrase, and cadence, whereas the low-level languages focus on notes, chords, and their local patterns. A cascaded diffusion model is trained to model the hierarchical language, where each level is conditioned on its upper levels. Experiments and analysis show that our model is capable of generating full-piece music with recognizable global verse-chorus structure and cadences, and the music quality is higher than the baselines. Additionally, we show that the proposed model is *controllable* in a flexible way. By sampling from the interpretable hierarchical languages or adjusting external controls, users can control the music flow via various features such as phrase harmonic structures, rhythmic patterns, and accompaniment texture.

1 INTRODUCTION

In recent years, we have witnessed a lot of progress in the field of deep music generation. With significant improvements on the quality of generated music (Copet et al., 2023; Thickstun et al., 2023) on short segments (typically ranging from a measure up to a phrase), researchers start to put more emphasis on *long-term structure* as well as how to *control* the generation process in a musical way. The current mainstream approach of structural generation involves first learning disentangled latent representations and then constructing a predictive model that can be controlled by the learned representations or external labels (Yang et al., 2019; Wang et al., 2020b; Wei et al., 2022; Chen et al., 2020). However, generating an entire song remains an unresolved challenge. As compositions extend in length, the number of involved music representations and their combinations grow exponentially, and therefore we need to organize various music representations in a structured way.

We argue that *compositional hierarchy* of music is the key to the solution. In this study, we focus on symbolic pop songs, proposing a computational *hierarchical music language* and modeling such language with cascaded diffusion models. The proposed music language has four levels. The top-level language describes the phrase structure and key progression of the piece. The second-level language reveals music counterpoint using a reduction of the melody and a rough chord progression, focusing on the music flow within phrases. The third-level language consists of the complete lead melody and the finalized chord progression, which is usually known as a lead sheet, further detailing the local music flow. At the last level, the language is defined as piano accompaniment. Intuitively, the language aims to characterize the intrinsic homophonic and tonal features of most pop songs — a verse-chorus form, a chord-driven tonal music flow, and a homophonic accompaniment texture.

We represent all levels of the symbolic languages as multi-channel images and train four layers of image diffusion models in a cascaded fashion, one for each level of the music language. The scope of the first layer is full-song and up to 256 measures, the scope of the second layer is 32 measures, and the third and the fourth layer each has a scope of 8 measures. Additional autoregressive controls are added to the low-level diffusion models to strengthen long-term temporal coherency. Experimental results show that our model is capable of generating well-structured full-piece music with recognizable verse-chorus structure and high music quality.

Moreover, at each level, optional *external conditions* can be added via the cross-attention mechanism of diffusion models to control the generation process at each level of the hierarchy. As a demonstration, we add long-term control of chord progression, local control of rhythmic and accompaniment pattern to the corresponding levels of the hierarchy. All the external controls uses pre-trained latent codes from existing music representation learning models. We show that these controls can effectively guide hierarchical generation in a more customizable way.

In summary, the contribution of the paper is as follows:

- **We achieve high-quality and well-structured whole-song generation** using a cascaded diffusion model approach. Objective and subjective measurement show that both monophonic lead sheets and polyphonic accompaniment generated by our model have more identifiable phrase boundaries, better-structured phrase development in similarity and contrast, and higher music quality compared to baselines.
- **We propose a computational hierarchical music language**, which serves as a structural inductive bias making the training process decomposable and efficient in terms of data and computing power utilization. Also, the hierarchical languages can be extracted automatically without manual annotation of music structure.
- **Our model enables flexible and interpretable controls**, with not only our proposed hierarchical language but also with external pre-trained latent representations in various music scopes.

2 RELATED WORK

In this section, we first review music structure in musicology in section 2.1, followed by music structure modeling in deep music generation approaches in section 2.2. Finally, in section 2.3 we review the state-of-the-art deep generative methods relevant to the problem of whole-song generation.

2.1 MUSIC STRUCTURE MODELING

Traditional music theory focuses on the analysis of music structure in terms of counterpoint (Clementi et al., 2010), harmony (Schoenberg, 1983), forms (Koch, 1787), etc. In the early 20th century, a more comprehensive theory, *Schenkerian analysis* (Cadwallader et al., 1998), emerged with a focus on the *generative* procedure of music. The theory introduces a compositional hierarchy of music, aiming to show how a piece is composed from its *background*, the normal form of music, to its *middle ground*, where music form and rough music development are realized, and finally to the *foreground*, the actual composition.

Nowadays, compositional hierarchy is still prevalent in modern musicology. Notable developments include Tagg (1982), a general compositional hierarchy for pop music, and *Generative Theory of Tonal Music* (GTTM) (Lerdahl & Jackendoff, 1996), a theory focusing on the definition and analysis of formal musical syntax. From a computational perspective, these studies provide more formal music features and computer-friendly generative processes (Hamanaka et al., 2015; 2016). The focus of this paper is to further leverage the compositional hierarchy of music to develop a fully computable language and to model it with deep generative modeling.

2.2 STRUCTURED DEEP MUSIC GENERATION

Recent advances in deep generative models have greatly improved music generation quality, primarily by more effective modeling of the local musical structure in two ways: implicit and explicit. Implicit approaches, exemplified by models such as Music Transformer (Huang et al., 2019), MuseBERT (Wang & Xia, 2021), and Jukebox (Dhariwal et al., 2020), learn structures by predicting and filling musical events, often revealing context dependencies via attention weights. Explicit approaches leverage domain knowledge to acquire interpretable music representations, allowing the learning of structures like measure-level pitch contour and accompaniment (Yang et al., 2019; Wang et al., 2020b). This study aims to combine both explicit and implicit approaches and further model

Table 1: Definition of the four-level hierarchical music language. We use `m` for measure, `b` for beat, `s` for step to represent the temporal resolution of each level. M denotes the number of measures in a piece, γ denote the number of beats in a measure, and δ denotes the number of steps in a beat.

Languages (res.)	**Specification**	**Data Representation**	**Structural Focus**
Form (`m`)	Key changes Phrase division	$\boldsymbol{X}^1 \in \mathbb{R}^{8\times M\times 12}$	Music form
Counterpoint (`b`)	Melody reduction Simplified chord	$\boldsymbol{X}^2 \in \mathbb{R}^{2\times \gamma M\times 128}$	Phrase similarity, phrase development & cadence
Lead Sheet (`s`)	Lead melody Chord	$\boldsymbol{X}^3 \in \mathbb{R}^{2\times \delta\gamma M\times 128}$	Melodic pattern, similarity & coherence
Accompaniment (`s`)	Accompaniment	$\boldsymbol{X}^4 \in \mathbb{R}^{2\times \delta\gamma M\times 128}$	Acc. pattern, similarity & coherence, Mel-acc relations

phrase and whole-song structures. The explicit modeling lies in our definition of a computational hierarchical music language, and the implicit modeling of the structure lies in the cascaded diffusion models.

2.3 DIFFUSION AND CASCADED MODELING FOR MUSIC GENERATION

Diffusion models, after their success in image and audio domains, have very recently been applied to music generation (Mittal et al., 2021; Li & Sung, 2023; Min et al., 2023). Besides high sample quality, diffusion models naturally lead to coherent local structures with the innate inpainting method Lugmayr et al. (2022), i.e., by generating music segments conditioned on surrounding contexts. As for long-term structures, we recently saw the design of cascaded diffusion modeling in Moûsai Schneider et al. (2023), which generates high-fidelity audios using multi-scale sampling.

In this study, our focus is on symbolic music and we adopt the idea of multi-scale generation in cascaded models. Additionally, we integrate the cascaded process with the proposed hierarchical music language, so that each layer of the diffusion model focuses on a certain interpretable aspect of music composition. In particular, all levels of music languages are defined as image-like representations. Inspired by sketch- and stroke-based image synthesis Cheng et al. (2023), we model hierarchical music generation by regarding high-level and low-level music languages as the background and foreground "strokes", respectively.

3 METHODOLOGY

Our model for whole-song generation is a realization of the music compositional hierarchy. In this section, we first introduce the definition of our hierarchical music languages in section 3.1. Then, we discuss how to model these languages via cascaded diffusion models, where each level of the language is conditioned on its upper levels. We show the data representation of these languages in section 3.2. The training and inference of the model are discussed in section 3.3 and section 3.4, respectively.

3.1 DEFINITION OF HIERARCHICAL MUSIC LANGUAGES

We define four levels of hierarchical music languages to reveal the generative procedure of music. As shown in Table 1, the four levels of languages, from highest to lowest, are: 1) *Form* of music key and phrase, 2) *Counterpoint* of reduced melody and simplified harmony, 3) *Lead Sheet* of melody and chords, and 4) *Accompaniment*. The key idea behind this hierarchical design lies in the relationship among the four levels — more abstract music concepts at higher levels are realized by stylistic specifications at lower levels. For example, a lead sheet is an abstraction implying many possible ways to arrange the accompaniment that share the same melodic and harmonic structure, while an instantiated accompaniment is one of the possible realizations showing the accompaniment structure in more detail.

Among the four levels, *Form*, *Lead Sheet*, and *Accompaniment* all involve common concepts in computer music, and there exist either labeled datasets or reliable algorithms to automatically extract the information from music. In contrast, melody reduction and simplified chords, as defined in *Counterpoint*, is our tailored design to show intermediate music structure involving cadence of phrase and phrase similarity. Similar music structures are rarely defined for automatic music generation. To this end, we contribute a *tonal reduction algorithm* to ensure the availability of this level of information (see Appendix A.1 for details) in order to complete the language hierarchy.

3.2 DATA REPRESENTATION

While music scores are inherently symbolic, we transform them into continuous, image-like piano-roll representations for better compatibility with diffusion models. Specifically, languages at all levels are represented by multi-channel images (examples are shown in Appendix A.2. The width of the images represents the music time under different resolutions, and the height represents 128 MIDI pitches or 12 pitch classes. We denote the length of a piece to be M measures, each measure containing γ beats, and each beat containing δ steps.

The language *Form* contains key signature and phrase division under the resolution of one measure. We represent key by $\boldsymbol{K} \in \mathbb{R}^{2\times M\times 12}$, where tonic information and scale information are stored on the two channels. As for phrase division, our representation is derived from the conventional string representation (Dai et al., 2020). For example, `"i4A8"` represents a 4-measure intro phrase followed by an 8-measure verse phrase. We use $\boldsymbol{P} \in \mathbb{R}^{6\times M\times 1}$ to represent phrase division where the 6 phrase types are mapped to 6 image channels (see Table 3 in Appendix A.2). We further introduce a measure countdown value to fill in the corresponding pixels. Formally, let $m_0, ..., m_0 + L - 1$ be the indices of a L-measure phrase of type i_0, then for $m_0 \leq m < m_0 + L$,

$$\boldsymbol{P}[i, m, :] := \mathbb{1}_{\{i=i_0\}}(1 - \frac{m - m_0}{L}). \tag{1}$$

We broadcast $\boldsymbol{P}$ to match the pitch-axis of $\boldsymbol{K}$ and define the first-level language *form* as the chromagram $\boldsymbol{X}^1 := \text{concat}(\boldsymbol{K}, \boldsymbol{P}) \in \mathbb{R}^{8\times M\times 12}$.

The other levels of languages use a similar piano-roll representation. The language *Counterpoint* is represented by $\boldsymbol{X}^2 \in \mathbb{R}^{2\times \gamma M\times 128}$ under the resolution of one beat, where two channels correspond to note onset and sustain. Both melody reduction and simplified chord progression share the same piano-roll using different pitch registers. Similarly, *Lead Sheet* uses $\boldsymbol{X}^3 \in \mathbb{R}^{2\times \delta\gamma M\times 128}$ to represent the actual melody and chords and *Accompaniment* uses $\boldsymbol{X}^4 \in \mathbb{R}^{2\times \delta\gamma M\times 128}$ to represent the accompaniment, both in the same resolution of one step.

Note that for the four levels $1 \leq k \leq 4$, $\boldsymbol{X}^k$ have different shapes. In the following sections, we write $\{\boldsymbol{X}^k | k \subset \{1, 2, 3, 4\}\}$ to denote the concatenation along the channel axes with possible broadcasting and repetition operations. For example, $\boldsymbol{X}^1$ can be expanded γ times in width and repeated 11 times in height to be concatenated with $\boldsymbol{X}^2$, resulting in a tensor in $\mathbb{R}^{10\times\gamma M,128}$. Additionally, we write the time-series expression $\boldsymbol{X}^k_t$ to denote $\boldsymbol{X}^k[:, t, :]$ for simplicity.

3.3 MODEL ARCHITECTURE

Whole-song music generation is achieved by generating the four levels of hierarchical music languages one after another in a top-down order (as shown in Figure 1). For each level, we train a diffusion model to realize the current-level language based on the existing upper-level languages. The *time scopes* (image widths) of these diffusion models are more or less the same (constrained only by computational resources) but the actual *music scopes* differ a lot, because the resolution in lower-level languages is finer. In this paper, for level $k = 1, ..., 4$, we set the time scope b_k to be $b_1 = 256$ and $b_{2:4} = 128$, which means the music scope for these levels are 256 measures, 128 beats, 128 steps, and 128 steps, respectively. In the usual setting when $\gamma = \delta = 4$, the scope of the models are 256 measures, 32 measures, 8 measures, and 8 measures, respectively. Consequently, except that the first layer is an unconditional generation of the whole sequence, the generation problems at all the other layers are essentially conditional generation of music segments sliced from the entire sequences.

The generation of a music language slice $\boldsymbol{X}^k_{t:t+b_k}$ at level $k \neq 1$ can be conditioned on multiple resources inside and outside the defined hierarchy. In this study, our model is designed to take in three sources of structural conditions:

Background condition. We regard the generation as a realization of existing higher-level languages at the corresponding scope $\boldsymbol{X}^{<k}_{t:t+b_k}$, where the higher-level language segments are like sketch im-

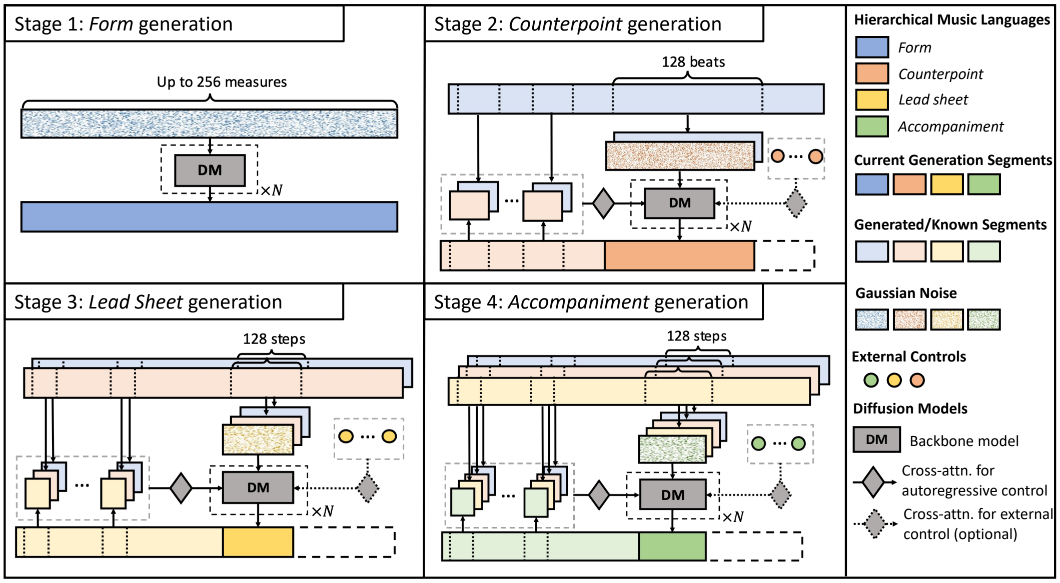

Stage 1: *Form* generation
Up to 256 measures
DM
×N
Stage 2: *Counterpoint* generation
128 beats
DM
×N
Stage 3: *Lead Sheet* generation
128 steps
DM
×N
Stage 4: *Accompaniment* generation
128 steps
DM
×N
Hierarchical Music Languages
Form
Counterpoint
Lead sheet
Accompaniment
Current Generation Segments
Generated/Known Segments
Gaussian Noise
External Controls
Diffusion Models
DM
Backbone model
Cross-attn. for autoregressive control
Cross-attn. for external control (optional)

Figure 1: Model diagram of cascaded diffusion models for hierarchical symbolic music generation.

ages directly guiding the current generation. Background condition is applied by concatenating the $\boldsymbol{X}^{<k}_{t:t+b_k}$ along the channel axis with the input image to the diffusion model.

Autoregressive condition. The segment should not only be a realization of the background condition, but also coherent with prior realizations $\boldsymbol{X}^{\leq k}_{<t}$. For example, the realization of a verse phrase in the end of the composition is usually similar to the realization in the beginning. In our model, we assume autoregression in the hierarchy that $\boldsymbol{X}^{k}_{<t}$ are known. We select S_k relevant music segments prior to t based on a defined similarity metric on $\boldsymbol{X}^{<k}$. These music segments are encoded into latent representations and being cross-attended in the diffusion models.

External condition. Besides the compositional hierarchy, music generation is usually controlled by other external conditions. These conditions can be possibly high-level or low-level stylistic controls (Wei & Xia, 2021), cross-modality control of text (Zhang et al., 2020), or audio (Wang et al., 2022). As an illustration of our model compatibility, we use pre-trained latent representations of long-term chord progression, rhythmic pattern, and accompaniment texture as the control for *Counterpoint*, *Lead Sheet*, and *Accompaniment* generation, respectively. At each level k, we denote the array of external control codes by $\boldsymbol{Z}^k$, which are cross-attended in our diffusion models.

Similar to Min et al. (2023), we adopt 2D-UNet with cross-attention as the backbone neural architecture for all four levels of the generation with several modifications. First, the input channels are increased to allow background condition. Second, autoregressive and external conditions are fed through cross-attention layers with classifier-free guidance (Baykal et al., 2023). In Appendix A.3 we include more detail on the model architecture and training method. The model is trained to model the conditional probability density of multiple levels of music segments. Formally, let the backbone model at level k be denoted by

$$\epsilon_{\theta_k}(x_n, n, y_1, y_2, y_3), \tag{2}$$

where θ_k is the model parameter, $n = 0, ..., N$ is the diffusion step, x_n is the input image mixed with Gaussian noise at diffusion step n, and y_1, y_2, y_3 are background, autoregressive, and external control, respectively. Our training objective is to model the probability

$$p_{\theta_k}(\boldsymbol{X}^{k}_{t:t+b_k} | \boldsymbol{X}^{<k}_{t:t+b_k}, \boldsymbol{X}^{\leq k}_{<t}, \boldsymbol{Z}^k) \tag{3}$$

under the loss function

$$\mathcal{L}(\theta_k) = \mathop{\mathbb{E}}_{\boldsymbol{X},t} \ell_{\theta_k}(\boldsymbol{X}^{k}_{t:t+b_k}, \boldsymbol{X}^{<k}_{t:t+b_k}, \boldsymbol{X}^{\leq k}_{<t}, \boldsymbol{Z}^k), \tag{4}$$

where

$$\ell_{\theta_k}(x, y_1, y_2, y_3) = \mathbb{E}_{\epsilon,n} || \epsilon - \epsilon_{\theta_k}(x_n, n, y_1, y_2, y_3)||_2^2. \tag{5}$$

Algorithm 1 Whole-song generation algorithm.

```
Constants: Resolution factor for each level $r_1 = 1, r_2 = \gamma, r_3 = r_4 = \delta\gamma$
Input: External control $\boldsymbol{Z}^k (2 \leq k \leq 4)$ (optional)
$\boldsymbol{X}^1 \sim p_{\theta_1}(\cdot|\emptyset, \emptyset, \emptyset)$
$M \leftarrow$ InferSongLength($\boldsymbol{X}^1$)
for $k = 2, \ldots, 4$ do
    $\boldsymbol{X}^k_{0:b_k /\!/ 2} \sim p_{\theta_k}(\cdot|\boldsymbol{X}^{<k}_{0:b_k}, \emptyset, \boldsymbol{Z}^k_{0:b_k})$
    for $t = 0, h_k, 2h_k, \ldots, r_k M$ do
        $\boldsymbol{X}^k_{t+h_k:t+b_k} \sim p_{\theta_k}(\cdot|\boldsymbol{X}^k_{t:t+h_k}; \boldsymbol{X}^{<k}_{t:t+b_k}, \boldsymbol{X}^{\leq k}_{<t}, \boldsymbol{Z}^k_{t:t+b_k})$
    end for
end for
return $\boldsymbol{X}^k (1 \leq k \leq 4)$
```

3.4 WHOLE-SONG GENERATION ALGORITHM

At the inference stage, we leverage the conditional probability eq. (3) to achieve whole song generation by autoregressively inpainting the generated segments using a hop length of $h_k := b_k /\!/ 2$. Inpainting is a commonly-used method in diffusion models for image editing, and is developed as a quasi-autoregressive method for sequential generation (Min et al., 2023). The whole-song generation algorithm is shown in Algorithm 1. Since $\boldsymbol{X}^1$ is zero-padded to 256 measures in training, in the inference algorithm, we derive the actual song length by finding the first all-zero entries of the generated $\boldsymbol{X}^1$. This process is denoted by INFERSONGLENGTH($\cdot$) in Algorithm 1. Here, we use

$$\boldsymbol{X}^k_{t+h_k:t+b_k} \sim p_{\theta_0}(\cdot|\boldsymbol{X}^k_{t:t+h_k}; \boldsymbol{X}^{<k}_{t:t+b_k}, \boldsymbol{X}^{\leq k}_{<t}, \boldsymbol{Z}^k_{t:t+b_k}) \quad (6)$$

to indicate the distribution of the second half of the sequence conditioned on the first half via inpainting.

4 ANALYSIS OF STRUCTURAL MUSIC GENERATION

In this section, we show an example of whole-song music generation of **40 measures** in Figure 2. The given *Form* of the piece has a simple verse-chorus structure with 4-measure verse and 8-measure chorus phrases appearing multiple times.

The generated music shows a clear music structure. The melodies of three verses all consist of syncopated rhythm in a narrow pitch range, while the melodies of two choruses are both relatively lyrical with a broader pitch range (indicated by shaded rectangles). The accompaniment pattern predominantly features eighth notes in verses and sixteenth notes in choruses. Moreover, the cadences at phrase boundaries are clearly indicated by the tonic or dominant chords and the "fill" in the accompaniment (indicated by dotted red rectangles). Furthermore, we notice the music intensity in the second half is stronger than in the first half, which is realized by more active pitch movements and higher pitches (indicated by shaded rectangles with dotted borders). Such intensity changes make the composition go to a climax point before ending, showing a well-formed chronological structure.

In Appendix A.4, we break down the hierarchical generation process and show examples of structural controllability of each level. More generation results are available at the demo page.[1]

5 EXPERIMENTS

We focus our experiments on evaluating the generated *Lead Sheet* and *Accompaniment*, the two lower levels of languages. The rationale is that the information of higher-level languages are difficult to evaluate directly, and they are implied at the lower levels.

We decompose whole-song evaluation into evaluation of music *structure* and *quality*. In section 5.3, we first propose an objective metric to measure structure, and then subjectively evaluate both structure and quality on music segments (8-measure) and whole-song samples (32 measures). An extra evaluation on generation plagiarism is discussed in Appendix A.5.

---

[1] Demo page: `https://wholesonggen.github.io`.

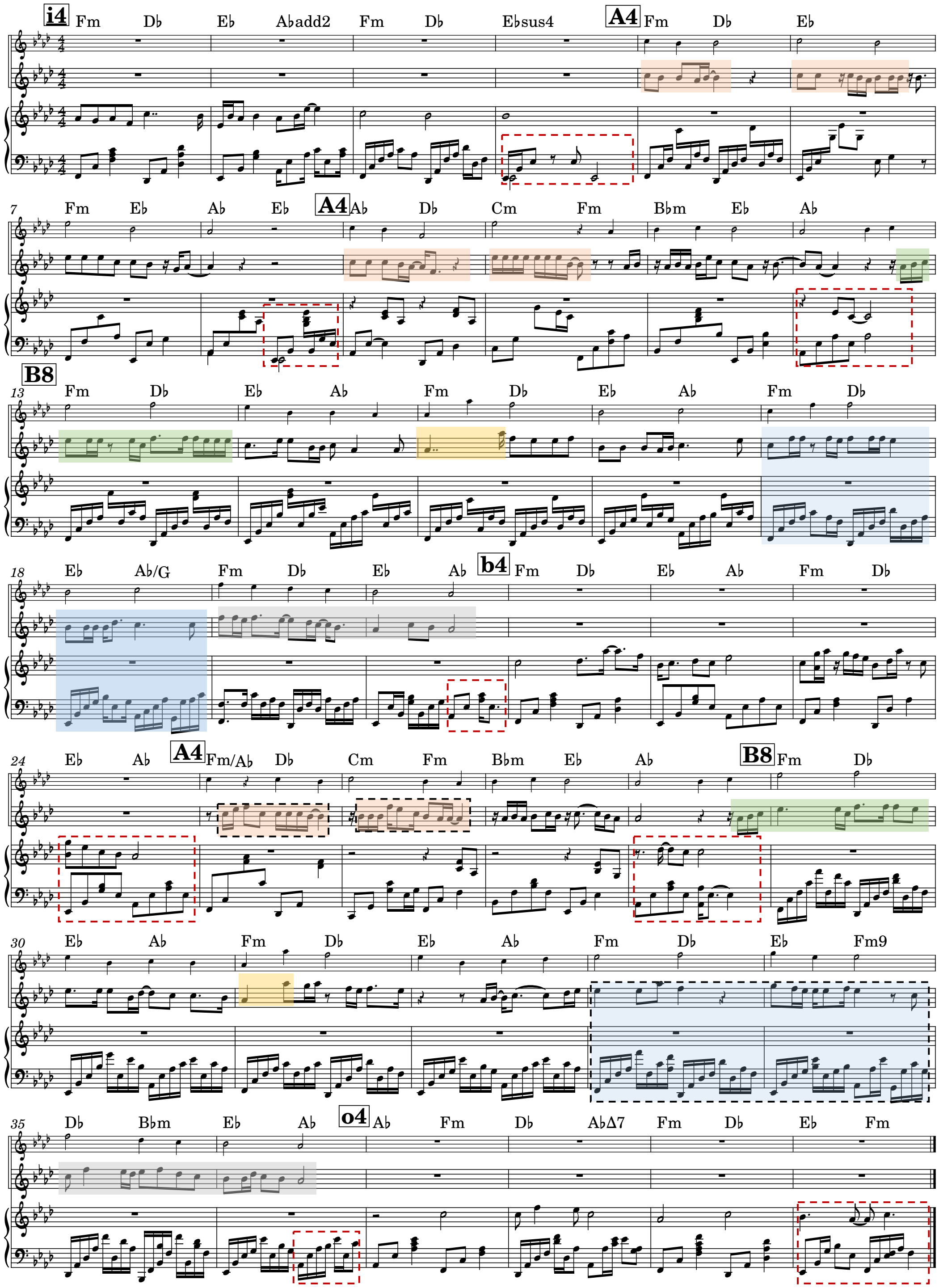


Figure 2: An example of whole-song generation of 40 measures under a given *Form* (A♭ major and `"i4A4A4B8b4A4A8o4"` phrase types). The three staves (from top to bottom) show the generated *Counterpoint*, *Lead Sheet*, and *Accompaniment*. Here, rectangles with colored background are used to indicate the appearance of the same motifs in verse and chorus sections. Dashed boarder rectangles with colored background indicate a variation of motifs. We use red dotted rectangles to show where the generated score show a strong implication of phrase boundary or cadence.

5.1 DATASET

We use the POP909 (Wang et al., 2020a) dataset to train our four-stage model. POP909 is a pop song dataset of 909 MIDI pieces containing lead melodies, secondary melodies, piano accompaniment tracks, key signatures, and chord annotations. We pad each song to 256 measures to train Stage 1, and segment each song into corresponding time scopes (128 beats for stage 2 and 128 steps for Stage 3 and 4) with a hop size of one measure. 90% of the songs are used for training and the rest 10% are used for testing. Training samples are transposed to all 12 keys. The annotations for key and phrase divisions are extracted using Dai et al. (2020). *Counterpoint* of each song are extracted using Tonal Reduction Algorithm proposed in Appendix A.1. All other annotations are available in the dataset.

5.2 BASELINE SETTINGS

We construct two baseline models for whole-piece lead sheet and accompaniment generation tasks. The two models are modified based on two state-of-the-art phrase-level generation models, respectively; one is diffusion-based, and the other is Transformer-based.

Diffusion-based (*Polyff.+ph.l.*). We augment Polyffusion (Min et al., 2023) with phrase label signals as external conditions, and use the iterative inpainting technique to generate whole pieces. We train two versions of diffusion models for lead sheet and accompaniment generation tasks on POP909, both adopting the same data representations as in section 3.2. This also serves as an ablation study on the effectiveness of our cascaded model design.

Transformer-based (*TFxl(REMI)+ph.l.*). Naruse et al. (2022) augment phrase label tokens on the REMI representation with Transformer-XL as the model backbone. Similarly, we train two versions for lead sheet and accompaniment generation on POP909, for lead sheet and accompaniment generation, respectively.

5.3 EVALUATION

Objective Evaluation. For whole-song well-structuredness, we design *Inter-phrase Latent Similarity* (ILS) to measure the content similarity among phrases of the same types (chorus or verse). We leverage pre-trained disentangled VAEs that encode music notes into latent representations and compare cosine similarities in the latent space. Given a similarity matrix showing pairwise similarity of 2-measure segments within a song, ILS is defined as the ratio between same-type phrase similarity and global average similarity, and therefore higher values indicate better structure.

We compute ILS on lead melody, chord, and accompaniment. Using pre-trained VAEs from Yang et al. (2019) and Wang et al. (2020b), we compute the latent representations of pitch contour and rhythm (i.e., $z_\mathrm{p}, z_\mathrm{r}$) for lead melody, latent z_chd for chord, and latent texture z_txt for accompaniment. We pre-define four types of common phrase divisions and let models generate 32 samples for each division, resulting in 128 full songs in total. $\mathrm{ILS}^\theta, \theta \in \{\mathrm{p}, \mathrm{r}, \mathrm{chd}, \mathrm{txt}\}$ are calculated for each song, and we show their mean and standard deviation in Table 2. The results show our model significantly outperforms baselines on the phrase content similarity of chord and accompaniment, indicating its effectiveness in preserving long-term structural dependency.

Table 2: Objective evaluation of phrase content similarity.

	Lead Melody		**Chord**	**Accompaniment**
	ILS^p ↑	ILS^r ↑	$\mathrm{ILS}^\mathrm{chd}$ ↑	$\mathrm{ILS}^\mathrm{txt}$ ↑
Ground Truth	2.28 ± 0.14	2.30 ± 0.13	1.42 ± 0.07	1.68 ± 0.09
Cas.Diff. (ours)	**2.05** ± 0.14	1.49 ± 0.07	**1.32** ± 0.05	**1.19** ± 0.06
Polyff. + ph.l.	0.60 ± 0.12	0.76 ± 0.05	0.52 ± 0.06	0.61 ± 0.04
TFxl(REMI) + ph.l.	1.89 ± 0.15	**1.71** ± 0.13	0.68 ± 0.06	0.74 ± 0.04

Subjective Evaluation. We design a double-blind online survey that consists of two parts: short-term (8 measures) evaluation of music quality, and whole-song (32 measures) evaluation of both music quality and well-structuredness. Participants rate *Creativity*, *Naturalness*, and *Musicality* for short-term music segments. For whole-song evaluation, we drop *Creativity* but introduce two more criteria: *Boundary Clarity* and *Phrase Similarity* to focus on the structure of the generation. All

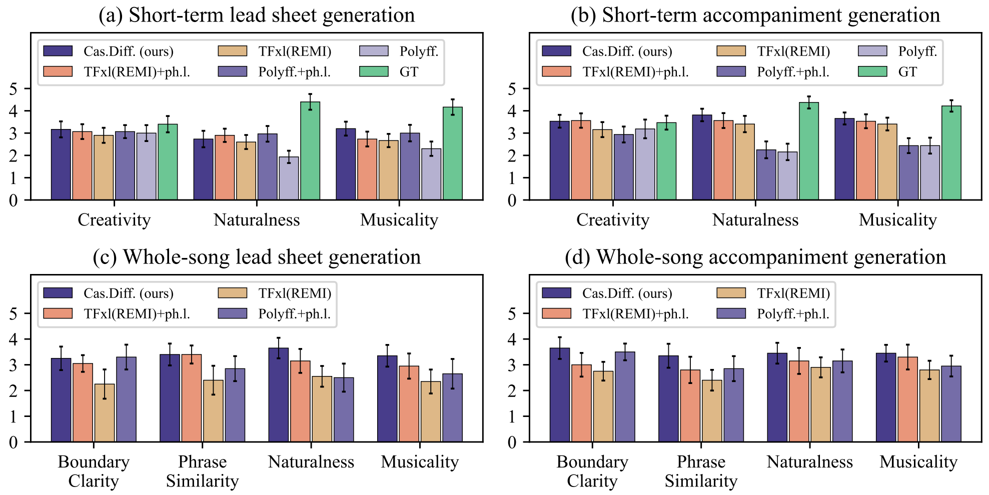


Figure 3: Subjective ratings on music quality and well-structuredness, evaluated over Cas.Diff. (Cascaded Diffusion), various baselines, and GT (human-composed ground truth).

metrics are rated based on a 5-point scale. We constrain whole-song length to 32 measures so that the participants can better memorize the samples and the survey has a reasonable duration. These generated pieces still preserve a condensed pop-song *Form* by specifying *Form* to contain intro, outro and repetitive verses or choruses.

Additionally, we use two more reference models (*Polyff.* and *TFxl(REMI)*, the two baseline models trained without phrase label conditions) for short-term evaluation. This is to investigate whether the inductive bias of phrase labels causes degradation in music quality. We let each model generate three samples for both short-term and whole-song levels as well as both lead sheet and accompaniment generation, resulting in $3 \times 2 \times 2 = 12$ groups of samples. Each group of samples share the same prompt (2 measures for 8-measure samples, and 4 measures for 32-measure samples) and phrase labels (for whole-song generation). In the survey, both the group order and the sample order are randomized.

A total of 57 people participated in our survey, and the evaluation result is shown in Figure 3. The bar height shows the mean rating, and the error bar shows its 95% confidence interval. Observe that our model significantly outperforms baselines in the structural metrics of whole-song generation, especially in accompaniment generation. Our model consistently outperforms baselines in terms of mean music quality in both short-term and whole-song generation, proving that our introduction of inductive bias does not degrade the generation quality.

6 CONCLUSION

In conclusion, we contribute the first hierarchical whole-song deep generative algorithm for symbolic music. The current study focuses on the pop music genre, and experimental results demonstrate that our model consistently generates more structured, natural, and musical outputs compared to baseline methods, both at the whole-song and the phrase scales. Additionally, our model offers extensibility, allowing flexible external controls via pre-trained music embeddings. Our approach relies on two key components: a hierarchical music language that balances human interpretability with computational tractability, and a cascaded diffusion architecture that effectively captures the hierarchical structure of entire compositions through both top-down and context-dependent mechanisms. It demonstrates that a strong structural inductive bias can lead to more effective and efficient learning for deep music generative models, and such methodology is potentially useful for other domains as well. In the future, we plan to extend our hierarchical language and generation approach to both multi-track symbolic music and music audio.

REFERENCES

Gulcin Baykal, Halil Faruk Karagoz, Taha Binhuraib, and Gozde Unal. Protodiffusion: Classifier-free diffusion guidance with prototype learning. *CoRR*, abs/2307.01924, 2023. doi: 10.48550/arXiv.2307.01924. URL `https://doi.org/10.48550/arXiv.2307.01924`.

Allen Clayton Cadwallader, David Gagné, and Frank Samarotto. Analysis of tonal music: a schenkerian approach. *(No Title)*, 1998.

Ke Chen, Cheng-i Wang, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Music sketchnet: Controllable music generation via factorized representations of pitch and rhythm. In Julie Cumming, Jin Ha Lee, Brian McFee, Markus Schedl, Johanna Devaney, Cory McKay, Eva Zangerle, and Timothy de Reuse (eds.), *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, pp. 77–84, 2020. URL `http://archives.ismir.net/ismir2020/paper/000146.pdf`.

Shin-I Cheng, Yu-Jie Chen, Wei-Chen Chiu, Hung-Yu Tseng, and Hsin-Ying Lee. Adaptively-realistic image generation from stroke and sketch with diffusion model. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, pp. 4043–4051. IEEE, 2023. doi: 10.1109/WACV56688.2023.00404. URL `https://doi.org/10.1109/WACV56688.2023.00404`.

Muzio Clementi, Carl Tausig, and Karl Friedrich Weitzmann. *Gradus ad parnassum*. Peters, 2010.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *arXiv preprint arXiv:2306.05284*, 2023.

Shuqi Dai, Huan Zhang, and Roger B Dannenberg. Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music. In *Proceedings of the 2020 Joint Conference on AI Music Creativity (CSMC-MuMe)*, 2020.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *CoRR*, abs/2005.00341, 2020. URL `https://arxiv.org/abs/2005.00341`.

Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo. σGTTM III: Learning-based time-span tree generator based on pcfg. In *International Symposium on Computer Music Multidisciplinary Research*, pp. 387–404. Springer, 2015.

Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo. deepGTTM-II: Automatic generation of metrical structure based on deep learning technique. In *13th Sound and Music Conference*, pp. 221–249, 2016.

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, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=rJe4ShAcF7`.

Heinrich Christoph Koch. *Versuch einer Anleitung zur Composition*, volume 72. bey Adam Friedrich Böhme, 1787.

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

Shuyu Li and Yunsick Sung. Melodydiffusion: Chord-conditioned melody generation using a transformer-based diffusion model. *Mathematics*, 11(8):1915, 2023.

Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 11451–11461. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01117. URL `https://doi.org/10.1109/CVPR52688.2022.01117`.

Lejun Min, Junyan Jiang, Gus Xia, and Jingwei Zhao. Polyffusion: A diffusion model for polyphonic score generation with internal and external controls. *CoRR*, abs/2307.10304, 2023. doi: 10.48550/arXiv.2307.10304. URL `https://doi.org/10.48550/arXiv.2307.10304`.

Gautam Mittal, Jesse H. Engel, Curtis Hawthorne, and Ian Simon. Symbolic music generation with diffusion models. In Jin Ha Lee, Alexander Lerch, Zhiyao Duan, Juhan Nam, Preeti Rao, Peter van Kranenburg, and Ajay Srinivasamurthy (eds.), *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, pp. 468–475, 2021. URL `https://archives.ismir.net/ismir2021/paper/000058.pdf`.

Daiki Naruse, Tomoyuki Takahata, Yusuke Mukuta, and Tatsuya Harada. Pop music generation with controllable phrase lengths. In Preeti Rao, Hema A. Murthy, Ajay Srinivasamurthy, Rachel M. Bittner, Rafael Caro Repetto, Masataka Goto, Xavier Serra, and Marius Miron (eds.), *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, pp. 125–131, 2022. URL `https://archives.ismir.net/ismir2022/paper/000014.pdf`.

Yi Ren, Jinzheng He, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Popmag: Pop music accompaniment generation. In Chang Wen Chen, Rita Cucchiara, Xian-Sheng Hua, Guo-Jun Qi, Elisa Ricci, Zhengyou Zhang, and Roger Zimmermann (eds.), *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*, pp. 1198–1206. ACM, 2020. doi: 10.1145/3394171.3413721. URL `https://doi.org/10.1145/3394171.3413721`.

Flavio Schneider, Zhijing Jin, and Bernhard Schölkopf. Moûsai: Text-to-music generation with long-context latent diffusion. *CoRR*, abs/2301.11757, 2023. doi: 10.48550/arXiv.2301.11757. URL `https://doi.org/10.48550/arXiv.2301.11757`.

Arnold Schoenberg. *Theory of harmony*. Univ of California Press, 1983.

Philip Tagg. Analysing popular music: theory, method and practice. *Popular music*, 2:37–67, 1982.

John Thickstun, David Hall, Chris Donahue, and Percy Liang. Anticipatory music transformer. *arXiv preprint arXiv:2306.08620*, 2023.

Ziyu Wang and Gus Xia. Musebert: Pre-training music representation for music understanding and controllable generation. In Jin Ha Lee, Alexander Lerch, Zhiyao Duan, Juhan Nam, Preeti Rao, Peter van Kranenburg, and Ajay Srinivasamurthy (eds.), *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, pp. 722–729, 2021. URL `https://archives.ismir.net/ismir2021/paper/000090.pdf`.

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

Ziyu Wang, Dingsu Wang, Yixiao Zhang, and Gus Xia. Learning interpretable representation for controllable polyphonic music generation. In Julie Cumming, Jin Ha Lee, Brian McFee, Markus Schedl, Johanna Devaney, Cory McKay, Eva Zangerle, and Timothy de Reuse (eds.), *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, pp. 662–669, 2020b. URL `http://archives.ismir.net/ismir2020/paper/000094.pdf`.

Ziyu Wang, Dejing Xu, Gus Xia, and Ying Shan. Audio-to-symbolic arrangement via cross-modal music representation learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pp. 181–185. IEEE, 2022. doi: 10.1109/ICASSP43922.2022.9747884. URL `https://doi.org/10.1109/ICASSP43922.2022.9747884`.

Shiqi Wei and Gus Xia. Learning long-term music representations via hierarchical contextual constraints. In Jin Ha Lee, Alexander Lerch, Zhiyao Duan, Juhan Nam, Preeti Rao, Peter van Kranenburg, and Ajay Srinivasamurthy (eds.), *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, pp. 738–745, 2021. URL `https://archives.ismir.net/ismir2021/paper/000092.pdf`.

Shiqi Wei, Gus Xia, Yixiao Zhang, Liwei Lin, and Weiguo Gao. Music phrase inpainting using long-term representation and contrastive loss. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pp. 186–190. IEEE, 2022. doi: 10.1109/ICASSP43922.2022.9747817. URL `https://doi.org/10.1109/ICASSP43922.2022.9747817`.

Ruihan Yang, Dingsu Wang, Ziyu Wang, Tianyao Chen, Junyan Jiang, and Gus Xia. Deep music analogy via latent representation disentanglement. In Arthur Flexer, Geoffroy Peeters, Julián Urbano, and Anja Volk (eds.), *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*, pp. 596–603, 2019. URL `http://archives.ismir.net/ismir2019/paper/000072.pdf`.

Yixiao Zhang, Ziyu Wang, Dingsu Wang, and Gus Xia. BUTTER: A representation learning framework for bi-directional music-sentence retrieval and generation. In *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, pp. 54–58, Online, 16 October 2020. Association for Computational Linguistics. URL `https://aclanthology.org/2020.nlp4musa-1.11`.

A APPENDIX

In the appendix, we provide detailed explanations of certain parts of our methodology and offer more examples of generated results. Specifically, we elaborate on the Tonal Reduction Algorithm (TRA) for the language *Counterpoint* in A.1; the detailed mapping rules of phrase types for language *Form* in A.2; model architecture and training details in A.3; some more generated examples in A.4; an evaluation on whether the generated examples have plagiarism issue in A.5; and an evaluation on external control efficacy in A.6.

A.1 TONAL REDUCTION ALGORITHM

The Tonal Reduction Algorithm (TRA) is designed for the extraction of the second-level music language *Counterpoint* (defined in section 3.1) containing melody reduction and simplified chord progression. Here, the extraction of simplified chord progression is relatively trivial and can be approximated by downsampling the chord progression to the ideal resolution. The most challenging aspect of TRA is to find melody reduction for a music phrase. The melody reduction should demonstrate a more fundamental music flow with respect to the harmony and remove local melodic contours, passing tones, motifs, and other music stylistic properties. In the following, we focus on the melody reduction part of the TRA algorithm.

Assume $(x_1, \dots, x_N)$ is a note sequence of a melody phrase. The sequence is sorted by note onsets and each note has properties of its onset, pitch, duration, and underlying chord progression. We consider the input music phrase under the graph representation $\mathcal{G}(V, E)$ where $V := \{x_1, ..., x_N\}$ are the vertices and $E := \{(x_i, x_{i+1}) | 1 \leq i < N\}$ are the edges showing temporal order. The key point of the proposed method is the discovery of skip edges $(x_i, x_j), i < j$, meaning the motion (x_i, x_j) is more structurally important than $(x'_i, x'_i + 1), \forall i \leq i' < j$. The algorithm defines cost of all possible edges based on music domain knowledge and find the shortest path from x_1 to x_N as the reduction of the input music phrase.

The cost of an edge $c(i, j), i < j$ has two components. First, based on the conventional belief in tonal music that stepwise and harmonic motion are more fundamental than local patterns (Cadwallader et al., 1998), we define a binary *progression cost* $c_0(x_i, x_j)$ based on the edge types:

- Prolongational edges: the pitches of x_i and x_j are identical and x_i and x_j are at most K measures apart, meaning that notes in-between are elaborations of x_j. In this case $c_0(x_i, x_j) = 0.1$.
- Imaginary prolongational edges: the pitch class of x_i and x_j are identical and x_i and x_j are at most K measures apart. In this case $c_0(x_i, x_j) = 1$.
- Linear edges: the interval between x_i and x_j is a major or minor second and x_i and x_j are at most K measures apart. In this case $c_0(x_i, x_j) = 0.3$.

- Imaginary linear edges: the pitch-class interval between x_i and x_j is a major or minor second and x_i and x_j are at most K measures apart. In this case $c_0(x_i, x_j) = 1.3$.
- Arpeggiation: the interval between x_i and x_j is larger than second and x_i and x_j are both chord tones belonging to the same chord. In this case $c_0(x_i, x_j) = 1.5$.
- Others: other (x_i, x_j) that does not belong to the above five categories and $j = i + 1$. In this case $c_0(x_i, x_j) = 3$.

Note that under this definition, the existence of at least one path from x_1 to x_N is ensured.

Additionally, we give the algorithm a preference of reduction resolution by a *distance cost* $c_2(x_i, x_j) := (j - i)^\alpha$. Hence, we extend the edges to E' and define the cost $(x_i, x_j) \in E'$ to be:

$$c(x_i, x_j) = c_1(x_i, x_j) + c_3(x_i, x_j). \tag{7}$$

We use the shortest-path algorithm to find the shortest path from x_1 to x_N. In practice, we find $K = 2$ and $\alpha = 1.6$ has the best performance.

After running the shortest path algorithm, we put these notes to the correct chord positions on the score and assign a fixed rhythmic pattern under the resolution of quarter notes.

Table 3: Definition of phrase type

Phrase Type	**Channel ID**	**Meaning**
`"A"`	0	Verse section phrases
`"B"`	1	Chorus section phrases
`"X"`	2	Other phrases with lead melody
`"i"`	3	Intro section phrases
`"o"`	4	Outro section phrases
`"b"`	5	Bridge section phrases

A.2 ADDITIONAL EXPLANATION OF DATA REPRESENTATION

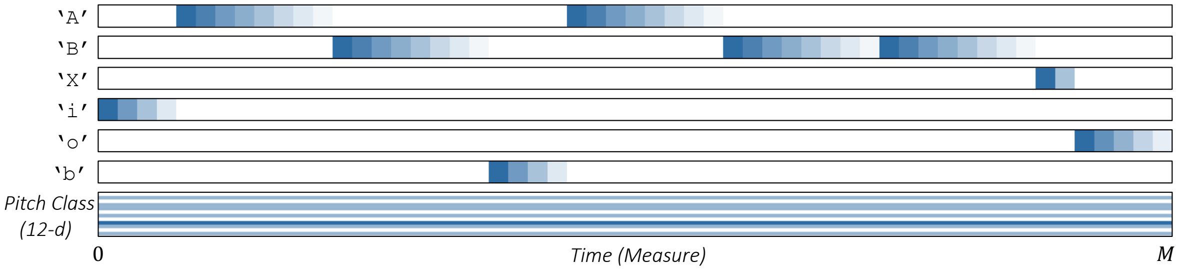
The language of *Form* $\boldsymbol{X}^1$ consists of phrase division $\boldsymbol{P} \in \mathbb{R}^{6 \times M \times 1}$ and key $\boldsymbol{K} \in \mathbb{R}^{2 \times M \times 12}$. In this section, we first present the six phrase types considered in this paper in Table 3. Such representation is originated from the string representations in Dai et al. (2020). Each phrase type is mapped to a channel in $\boldsymbol{P}$ following Table 3. We also provide an example data representation of all four levels of hierarchical language in Figure 4.

A.3 MODEL ARCHITECTURE AND TRAINING DETAILS

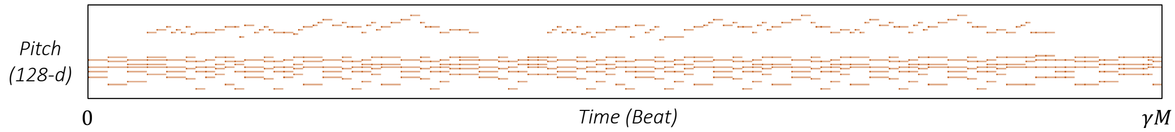
In this section, we provide more detail on the model architecture and training method. We first discuss the three conditioning methods introduced in section 3.3 in more detail and summarize the information in Table 4.

Detail on background condition. At each level $k > 1$, the background condition $\boldsymbol{X}^{<k}_{t:t+b_k}$ is represented by an image having the same width and height as the diffusion output. Thus, the background condition can be concatenated with the input along channel axis at each step of the diffusion process. The background condition will be set to all -1.0 under the probability $p_{\text{uncond}} = 0.2$, following classifier-free guidance (Baykal et al., 2023).

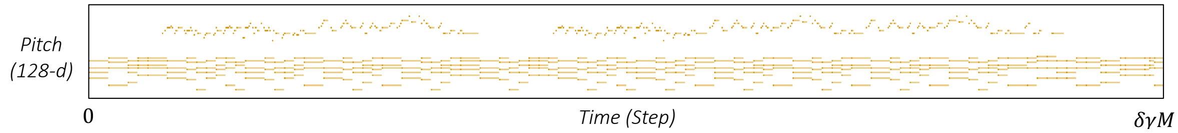
Detail on autoregressive condition. At each level $k > 1$, the generation of $\boldsymbol{X}^k_{t:t+b_k}$ is dependent on past generation $\boldsymbol{X}^{\le k}_{<t}$. Here we select top-$S_k$ past segments of length b'_k based on their *phrase type similarity* to the current segment. (Recall that phrase type is encoded in $\boldsymbol{X}^1$.) These music segments are embedded using 3 layers of 2d-convolution and fed to the backbone diffusion models by cross-attention mechanism. In our implementation, we set $S_2 = 3$, $b'_2 = 32$, $S_3 = S_4 = 1$, and $b'_3 = b'_4 = 96$. The autoregressive condition will be set to all -1.0 under the probability $p_{\text{uncond}} = 0.1$.



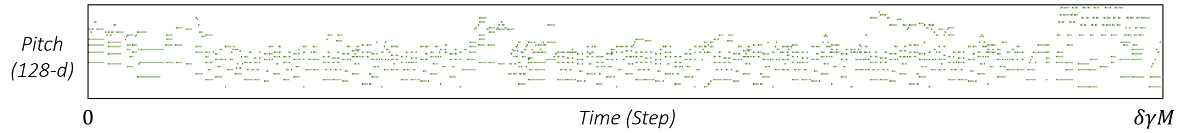
(a) First-level music language: *Form*.



(b) Second-level language: *Counterpoint*.



(c) Third-level language: *Lead Sheet*.



(d) Fourth-level language: *Accompaniment*.

Figure 4: An illustration of our proposed hierarchical music languages.

Table 4: Details about the conditioning methods in four stages of diffusion models.

Generation stages	***Form***	***Counterpoint***	***Lead Sheet***	***Accompaniment***
Time scope	256 measures	128 beats	128 steps	128 steps
Output shape	$(2, 256, 12)$	$(2, 128, 128)$	$(2, 128, 128)$	$(2, 128, 128)$
Background cond: shape	N/A	$(6, 128, 128)$	$(8, 128, 128)$	$(10, 128, 128)$
Background cond: p_{uncond}	N/A	0.2	0.2	0.2
Autoreg. cond: # of segments	N/A	3	1	1
Autoreg. cond: shape	N/A	$(8, 32, 128)$	$(10, 96, 128)$	$(12, 96, 128)$
Autoreg. cond: p_{uncond}	N/A	0.1	0.1	0.1
External cond: # of latent codes	N/A	4	4	4
External cond: latent dimension	N/A	512	128	256
External cond: p_{uncond}	N/A	0.2	0.2	0.2

Detail on external condition. The condition for *Counterpoint* is four 8-measure latent codes of chord progression encoded from Wang et al. (2020b). The condition for *Lead Sheet* is four latent codes of rhythmic pattern encoded from Yang et al. (2019). The condition for *Accompaniment* is four latent codes of accompaniment texture encoded from Wang et al. (2020b). These latent codes are fed to the backbone diffusion models by cross-attention mechanism and will be set to all -1.0 under the probability $p_{\text{uncond}} = 0.2$.

The diffusion models for all four stages use the same noise schedule and training methods. Similar to Min et al. (2023), the backbone model is a 2D-UNet model, the encoder and decoder of which contain 4 layers of 2d-convolution with spatial attention at the third and forth layers. We summarize these common details across all four stages in Table 5.

Table 5: Details about the backbone UNet model and training method. These features are the same across all four stages.

Attributes	**Value**
Diffusion Steps (N)	1000
Noise Schedule	Linear from 1 to 1e-4
UNet Channels	64
UNet Channel Multipliers	1,2,4,4
Batch Size	16
Attention Levels	3,4
Number of Heads	4
Learning Rate	5e-5

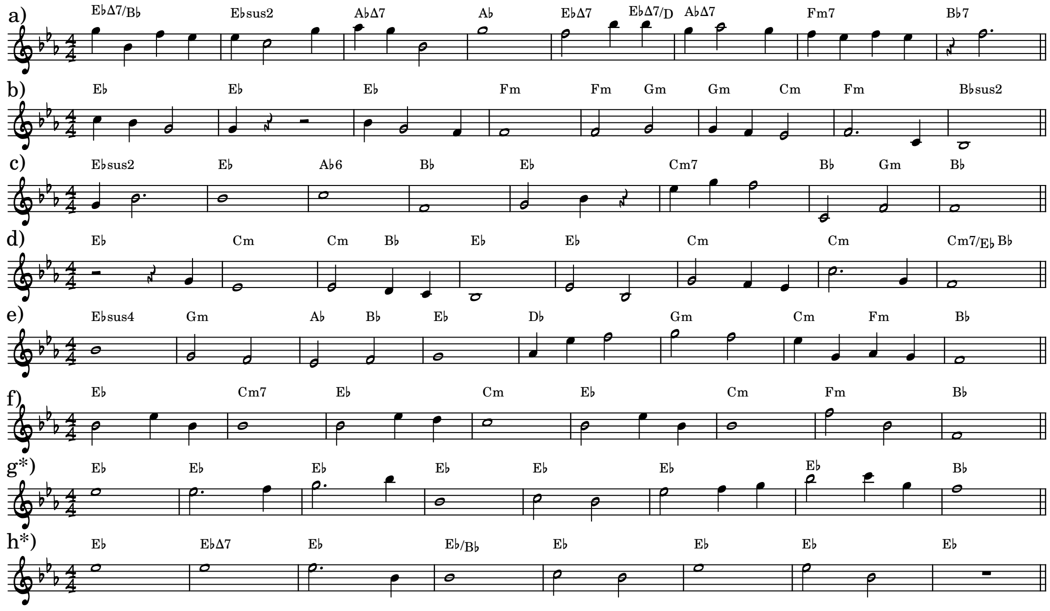


Figure 5: Examples of generated *Counterpoint* of `"A8"` phrase in E♭ major. The samples marked with * are controlled by external conditions of “Unchanging chord progression”.

A.4 MORE EXAMPLES ON STRUCTURAL GENERATION

In this section, we break down each level of the hierarchical language and show more generation examples. For each level, we fix the upper level, and demonstrate a variety of generation results under the upper level control. We also introduce generation samples that are controlled by external conditions.

***Form* generation.** Below shows examples of *Form* generated by our model:

```
(i8)(A8B16A8B16)(b6)(B14)(o2)
(i12)(A4A4B12)(b4b4)(A4A4B12)(b4b4(B16)o4o1)
(i4)(A4A4B4X5)(b4)(A4b5B4X4X5)(o2)
(i4)(A8B9A8B9X18)(o2o1)
```

Here, we use parentheses to group music sections for better readability. The results show the model captures verse-chorus form of pop songs: the composition usually starts with intro and ends with outro; verse and chorus appears multiple times with bridge phrases in between. Phrases are usually 4 or 8 measures long, similar to real music samples.

***Counterpoint* generation with external harmony control.** Figure 5(a)-(f) show examples of 8-measure generation of the *Counterpoint* level. The results are all controlled by the same *Form*: an

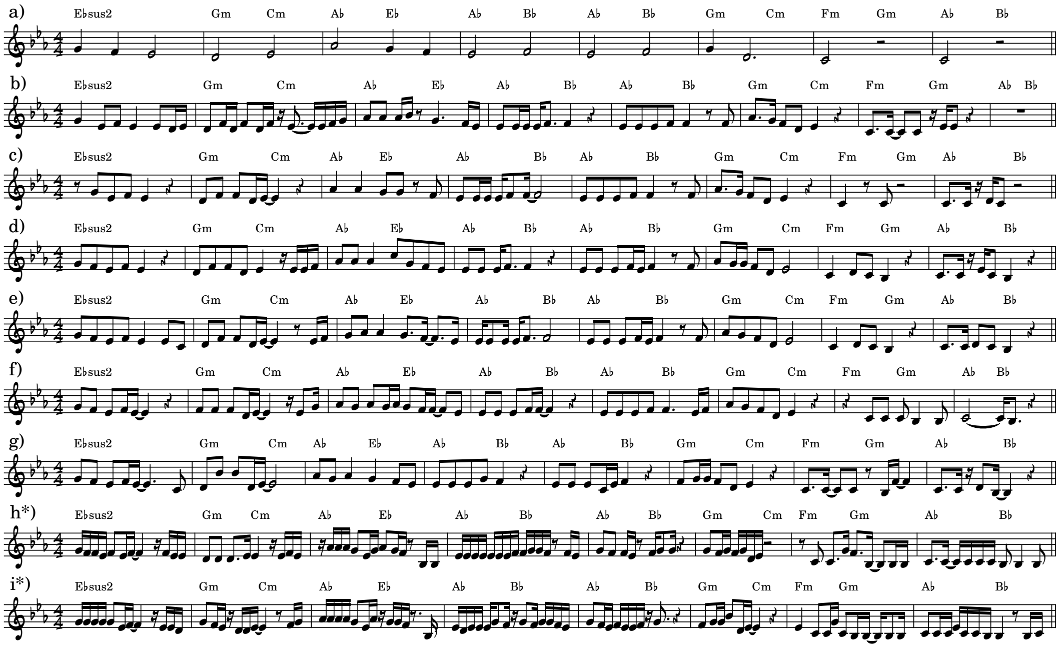

a) E♭sus2 Gm Cm A♭ E♭ A♭ B♭ A♭ B♭ Gm Cm Fm Gm A♭ B♭
b) E♭sus2 Gm Cm A♭ E♭ A♭ B♭ A♭ B♭ Gm Cm Fm Gm A♭ B♭
c) E♭sus2 Gm Cm A♭ E♭ A♭ B♭ A♭ B♭ Gm Cm Fm Gm A♭ B♭
d) E♭sus2 Gm Cm A♭ E♭ A♭ B♭ A♭ B♭ Gm Cm Fm Gm A♭ B♭
e) E♭sus2 Gm Cm A♭ E♭ A♭ B♭ A♭ B♭ Gm Cm Fm Gm A♭ B♭
f) E♭sus2 Gm Cm A♭ E♭ A♭ B♭ A♭ B♭ Gm Cm Fm Gm A♭ B♭
g) E♭sus2 Gm Cm A♭ E♭ A♭ B♭ A♭ B♭ Gm Cm Fm Gm A♭ B♭
h*) E♭sus2 Gm Cm A♭ E♭ A♭ B♭ A♭ B♭ Gm Cm Fm Gm A♭ B♭
i*) E♭sus2 Gm Cm A♭ E♭ A♭ B♭ A♭ B♭ Gm Cm Fm Gm A♭ B♭

Figure 6: Examples of generated *Lead Sheet* of `"A8"` phrase in E♭ major given the upper-level *Counterpoint*. The samples with marked with * are controlled by additional latent codes.

8-measure verse phrase in E♭ major. The generated samples show many possible ways to develop the melody (different contour and melodic climax positions) and the harmony (different chord types and harmonic rhythm). Moreover, each of the samples has a consistent style and usually ends in a tonic or dominant chord indicating the ending of a phrase. Moreover, we also use a latent chord representation (encoded from Wang et al. (2020b)) of unchanging chord to control the *Counterpoint* generation process and the results in shown in Figure 5(g)-(h). The results have fewer changes in harmony and the melody reduction alters accordingly.

***Lead Sheet* generation with external rhythm control.** Figure 6(b)-(g) show examples of 8-measure generation of the *Lead Sheet* level. The results are all controlled by the same *Counterpoint*, shown in Figure 6(a). The generated samples follow the pitch contour in the melody reduction and different in local pitch and rhythm patterns. At this level, we also use latent control of rhythm (encoded from Yang et al. (2019)) to control the melody generation using sixteenth notes. The generation examples shown in Figure 5(h)-(i) shows melody realization with more frequent onsets accordingly.

***Accompaniment* generation with external texture control.** Figure 7(b)-(d) show examples of 8-measure generation of the *Accompaniment* level controlled by the same *Lead Sheet*, shown in Figure 7(a). The generated samples mainly use arpeggios but are different in the exact patterns. Some of the generation has a “fill” in the fourth and eighth measures to indicate phrasing. At this level, we also use latent control of texture (encoded from Wang et al. (2020b)) to control the accompaniment generation. Here the control is a texture of “Alberti bass” figure and Figure 5(e) shows the generation results. The generation adopts the Alberti bass pattern and makes variations throughout the piece.

A.5 EVALUATION OF GENERATION PLAGIARISM

In generative modeling, a critical consideration is whether the model overfits the training data, resulting in generation plagiarism. This section includes a quantitative evaluation focused on the similarity between generated segments and the entire training set. We primarily focus on *melody* similarity, a most recognizable aspect of music composition.

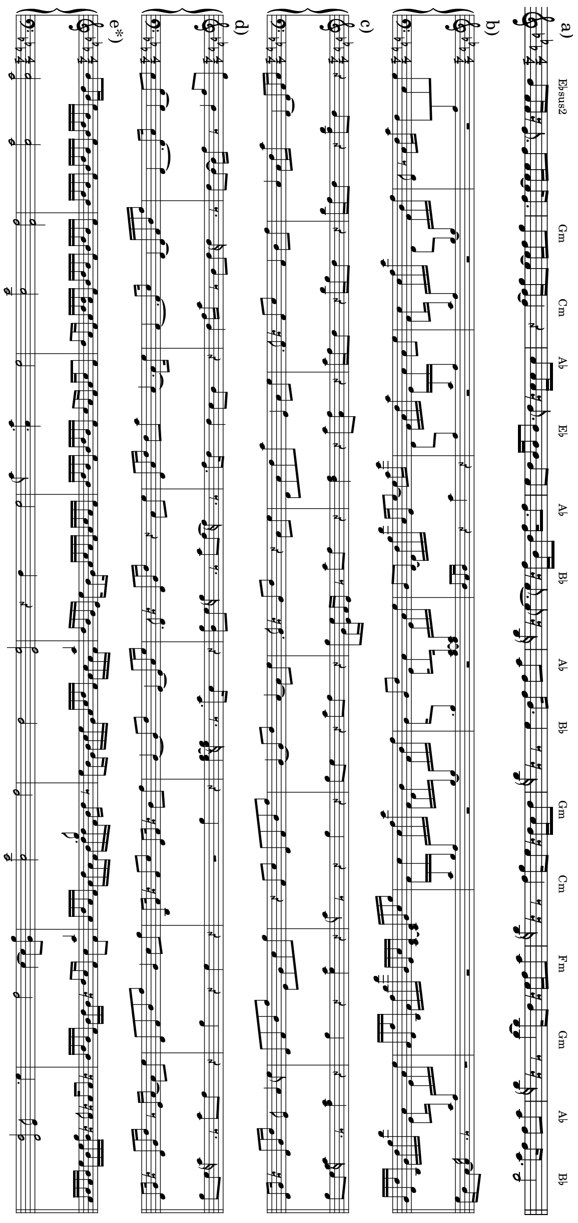


Figure 7: Examples of generated *Accompaniment* of `"A8"` phrase in E♭ major given the upper-level *Melody*. The samples with marked with * are controlled by additional latent codes.

Our goal is to measure the *Degree of Plagiarism (DoP)* with respect to a set of generated samples, or a specific sample from this set. Let $\boldsymbol{x}$ be a two-measure melody segment from a generated piece or a set of pieces, we define *Similarity to the Training Set* of segment $\boldsymbol{x}$ as:

$$S(\boldsymbol{x}) := \max_{\boldsymbol{x}' \in \mathcal{T}} \operatorname{sim}(\boldsymbol{x}, \boldsymbol{x}'), \tag{8}$$

where $\mathcal{T}$ denotes the training set, $\boldsymbol{x}'$ is a two-measure segment from the training set, and $\operatorname{sim}(\boldsymbol{x}, \boldsymbol{x}')$ computes the similarity between $\boldsymbol{x}$ and $\boldsymbol{x}'$. Here $S(\boldsymbol{x}) \in [0, 1]$, and a larger $S(\boldsymbol{x})$ shows a higher degree of plagiarism. The DoP of a piece or a set of pieces can be represented by the histogram of $S(\boldsymbol{x})$. We typically report the mean and standard deviation of the histogram in the following experiments.

We define a rule-based similarity metric and a latent similarity metric as follows:

Rule-based similarity metric. We compute the note-wise similarity between two segments by matching the exact onsets and pitches. Let $n_{\boldsymbol{x}\cap\boldsymbol{x}'}$ denote the number of notes that appear in both $\boldsymbol{x}$ and $\boldsymbol{x}'$ with the same *pitch class* and *onset*, and let $n_{\boldsymbol{x}}$ and $n_{\boldsymbol{x}'}$ denote the number of notes in $\boldsymbol{x}$ and $\boldsymbol{x}'$, respectively. The rule-based similarity metric is defined as:

$$\mathrm{sim}^{\mathrm{rb}}(\boldsymbol{x}, \boldsymbol{x}') := \frac{2n_{\boldsymbol{x}\cap\boldsymbol{x}'}}{n_{\boldsymbol{x}} + n_{\boldsymbol{x}'}}.$$

Latent similarity metric. We also measure the melodic similarity in the latent space because rule-based methods cannot detect *indirect plagiarism* (e.g., same pitch contour or rhythm). We leverage the pre-trained EC^2-VAE (Yang et al., 2019), which learns a semantic meaningful and disentangled latent space of pitch contour and rhythmic pattern. We extract the latent code of pitch (denoted as $\boldsymbol{z}_{\mathrm{p}}^{\boldsymbol{x}}$) and rhythm (denoted as $\boldsymbol{z}_{\mathrm{r}}^{\boldsymbol{x}}$) of melody segments and compute the cosine similarity in terms of both pitch and rhythm:

$$\mathrm{sim}_{\mathrm{p}}^{\mathrm{lt}}(\boldsymbol{x}, \boldsymbol{x}') := \frac{\langle \boldsymbol{z}_{\mathrm{p}}^{\boldsymbol{x}}, \boldsymbol{z}_{\mathrm{p}}^{\boldsymbol{x}'} \rangle}{||\boldsymbol{z}_{\mathrm{p}}^{\boldsymbol{x}}|| \cdot ||\boldsymbol{z}_{\mathrm{p}}^{\boldsymbol{x}'}||}, \tag{9}$$

$$\mathrm{sim}_{\mathrm{r}}^{\mathrm{lt}}(\boldsymbol{x}, \boldsymbol{x}') := \frac{\langle \boldsymbol{z}_{\mathrm{r}}^{\boldsymbol{x}}, \boldsymbol{z}_{\mathrm{r}}^{\boldsymbol{x}'} \rangle}{||\boldsymbol{z}_{\mathrm{r}}^{\boldsymbol{x}}|| \cdot ||\boldsymbol{z}_{\mathrm{r}}^{\boldsymbol{x}'}||}. \tag{10}$$

For both of the metrics, the samples in the training sets are transposed to 12 keys to account for relative pitch similarity. Segments that only contain rests are discarded beforehand.

Table 6: Plagiarism evaluation of the models and the references. The highlighted data in red indicate potential plagiarism.

Sample Source	**Sample Size**	**Similarity Metric**		
		$\mathrm{sim}^{\mathrm{rb}} \downarrow$	$\mathrm{sim}_{\mathrm{p}}^{\mathrm{lt}} \downarrow$	$\mathrm{sim}_{\mathrm{r}}^{\mathrm{lt}} \downarrow$
Test set *(no plag.)*	88 pieces	0.6567 ± 0.1141	0.8637 ± 0.0486	0.8320 ± 0.0680
Copy-bot 1 *(plag.)*	128 pieces	0.7108 ± 0.1159	0.8616 ± 0.0526	0.8276 ± 0.0699
Copy-bot 2 *(plag.)*	128 pieces	0.6888 ± 0.1628	0.9086 ± 0.0340	0.8555 ± 0.0411
Cas.Diff. (ours)	128 pieces	0.6530 ± 0.1321	0.8743 ± 0.0491	0.8180 ± 0.0710
Polyff. + ph.l.	128 pieces	0.6117 ± 0.1162	0.8639 ± 0.0487	0.8424 ± 0.0622
TFxl(REMI) + ph.l.	128 pieces	0.6088 ± 0.1053	0.8599 ± 0.0446	0.8154 ± 0.0642

In Table 6, we show the mean and standard deviation of $S(\boldsymbol{x})$ on the data samples generated using our proposed methods and other baselines used for whole-song generation. We compute the statistics of the test set of POP909 as a reference for *plagiarism-free*, since no song in the training set (or their cover-song versions) appears in the test set. We also design two copy-bots as references for *potential plagiarism*. The first copy-bot copies different part of the training set at each measure, which emulates a *direct plagiarism* behavior. The second copy-bot encodes the melodies from the training set and adds noise to the latent representation before reconstruction, which emulates an *indirect plagiarism* behavior. *Experimental results show that our proposed method (as well as the baseline whole-song generation methods) have similar DoPs compared to the plagiarism-free DoP of the test set. Also, the proposed metrics successfully detect both direct and indirect plagiarism behaviors as the DoPs of copy-bots are noticeably higher. Thus, we conclude that our model has a very low risk of plagiarism.*

In Table 7, we show the mean and standard deviation of $S(\boldsymbol{x})$ on each generated samples in the demo page. Experimental results show that all samples are plagiarism-free except the third sample in the section “More Examples of Whole-song Generation” of the demo page.

A.6 EVALUATION OF EXTERNAL CONTROL EFFICACY

In this section, we evaluate the efficacy of external controls. These controls are achieved by feeding pre-trained representations as external condition to each layer of the cascaded diffusion model (introduced in section 3.3). Specifically, we evaluate three scenario: (1) chord control in *Counterpoint* generation (Stage two), (2) rhythm control in *Lead Sheet* generation (Stage three), and (3) texture

Table 7: Plagiarism evaluation of the samples in the demo page. The highlighted data in red indicate potential plagiarism.

Generated sample	**Similarity Metric**		
	$\text{sim}^{\text{rb}} \downarrow$	$\text{sim}_{\text{p}}^{\text{lt}} \downarrow$	$\text{sim}_{\text{r}}^{\text{lt}} \downarrow$
Main demo (Figure 2)	0.5846 ± 0.0917	0.8943 ± 0.0336	0.7684 ± 0.0634
More demo 1	0.5770 ± 0.1262	0.8856 ± 0.0385	0.7919 ± 0.0504
More demo 2	0.5462 ± 0.0880	0.8861 ± 0.0342	0.7786 ± 0.0425
More demo 3	0.7867 ± 0.1996	0.8950 ± 0.0831	0.8926 ± 0.1000
More demo 4	0.6476 ± 0.0817	0.8861 ± 0.0342	0.7786 ± 0.0425
More demo 5	0.5949 ± 0.0434	0.8861 ± 0.0342	0.7786 ± 0.0425
More demo 6	0.6050 ± 0.0976	0.8348 ± 0.0477	0.8322 ± 0.0497

control in *Accompaniment* generation (Stage four). In this section, we let $\boldsymbol{z}^{\text{ext}}$ denote the external control in one of the three scenario and let $\boldsymbol{x}^{\text{ext}}$ denote the actual observation from which $\boldsymbol{z}^{\text{ext}}$ is encoded from. Let $\boldsymbol{x}^{\text{out}}$ denote the conditional generation results.

Efficacy of control can be evaluated by computing the similarity between the input control and the generation result. We propose a *rule-based metric* and a *latent metric*. The rule-based metric directly computes the distance between $\boldsymbol{x}^{\text{out}}$ and $\boldsymbol{x}^{\text{ext}}$ in terms of the corresponding features. In particular, for chord control, we compute the ℓ_2 distance between the given chord condition and the generated chord at each time step; and for rhythm or texture control, we compute the ℓ_2 distance of note onsets between the given control and the generated lead sheets or accompaniments. Such distance-based metric has previously been used to evaluate control efficacy in Ren et al. (2020) and Min et al. (2023). In the latent metric, we encode the generation $\boldsymbol{x}^{\text{out}}$ back to the latent code $\boldsymbol{z}^{\text{out}}$ using the same pre-trained encoders and measure the cosine similarity between $\boldsymbol{z}^{\text{out}}$ and $\boldsymbol{z}^{\text{ext}}$.

There are two reference methods for comparison. First, we use the unconditional mode of our method to serve as a baseline where control is ineffective. Second, we generate samples by sampling from the Variational Autoencoders (VAEs) that the pre-trained encoders belong to. Specifically, we sample $\boldsymbol{z}^{\text{ext}}$ from the VAE posterior distribution and sample the rest of the latent codes from Gaussian prior to decode results. By the well-disentangled property shown in the original paper (Yang et al., 2019; Wang et al., 2020b), this reference method indicates the maximum attainable level of controllability.

For each of the three scenario, we randomly selected 32 versions of external control from the test set and generate 128 music segments for each methods. In Table 8, we show the rule-based distance (denoted by dis^{rb}) and latent similarity (denoted by sim^{lt}) for the three generation stages. Experimental results show that the use of external condition significant yields controllability for all three scenario.

Table 8: Objective evaluation of external control efficacy of chord, rhythm and texture in the three diffusion stages. dis^{rb} denotes the rule-based distance-based metric and sim^{lt} denotes the latent similarity-based metric.

	Stage 1: Chord		**Stage 2: Rhythm**		**Stage 3: Texture**	
	$\text{dis}^{\text{rb}} \downarrow$	$\text{sim}^{\text{lt}} \uparrow$	$\text{dis}^{\text{rb}} \downarrow$	$\text{sim}^{\text{lt}} \uparrow$	$\text{dis}^{\text{rb}} \downarrow$	$\text{sim}^{\text{lt}} \uparrow$
Cas.Diff. (uncond)	2.09 ± 0.80	0.37 ± 0.09	2.27 ± 0.53	0.14 ± 0.23	3.94 ± 1.46	0.02 ± 0.11
VAE Sampling	0.19 ± 0.47	0.97 ± 0.07	0.14 ± 0.42	0.96 ± 0.04	0.33 ± 0.59	0.90 ± 0.06
Cas.Diff. (cond)	1.73 ± 1.02	0.48 ± 0.14	1.10 ± 0.74	0.75 ± 0.16	0.87 ± 0.80	0.89 ± 0.06