Efficient Fine-Grained Guidance for Diffusion Model Based Symbolic Music Generation

Tingyu Zhu [*1] **Haoyu Liu** [*1] **Ziyu Wang** [2] **Zhimin Jiang** [3] **Zeyu Zheng** [1]

Abstract

Developing generative models to create or conditionally create symbolic music presents unique challenges due to the combination of limited data availability and the need for high precision in note pitch. To address these challenges, we introduce an efficient Fine-Grained Guidance (FGG) approach within diffusion models. FGG guides the diffusion models to generate music that aligns more closely with the control and intent of expert composers, which is critical to improve the accuracy, listenability, and quality of generated music. This approach empowers diffusion models to excel in advanced applications such as improvisation, and interactive music creation. We derive theoretical characterizations for both the challenges in symbolic music generation and the effects of the FGG approach. We provide numerical experiments and subjective evaluation to demonstrate the effectiveness of our approach. We have published a demo page [1] to showcase performances, which enables real-time interactive generation.

1. Introduction

Diffusion models (Ho et al., 2020) have consistently demonstrated effectiveness across a wide range of generative tasks, particularly in image and video generation (Rombach et al., 2022). Despite success, diffusion models face some limitations. (1) Imprecise detail generation: Diffusion models often struggle with accurately producing details, leading to artifacts or distortions in the generated content, such as noticeable inconsistencies or distortions in videos. (2) Limited controllability: Obtaining precise control over the generated content to align it with the intent of the user remains a significant challenge. For instance, correcting specific distortions in a generated video while keeping the rest of the scene unchanged is difficult with current diffusion model frameworks.

These limitations are exacerbated in situations where data is scarce, which is often the case in domains like symbolic music generation, where symbolic music data is limited due to copyright constraints and the effort needed to create data. Additionally, unlike image generation, where the inaccuracy of a single pixel may not significantly affect overall quality, symbolic music generation demands high precision, especially in terms of pitch. In many musical and tonal contexts, even a single incorrect or inconsistent note can be glaringly obvious and disturbing.

To provide more contexts, symbolic music generation is a subfield of music generation that focuses on creating music in symbolic form, typically represented as sequences of discrete events such as notes, pitches, rhythms, and durations. These representations are analogous to traditional sheet music or MIDI files, where the structure of the music is defined by explicit musical symbols rather than audio waveforms. Many recent works in symbolic music generation are based on diffusion models; see Min et al. (2023), Wang et al. (2024) and Huang et al. (2024) for example.

Following this branch of work, we address the precision and controllability challenges in diffusion-based symbolic music generation by incorporating fine-grained guidance into the training and sampling processes. While soft control schemes such as providing chord conditions may fail to ensure detailed pitch correctness, we propose to enhance chord conditioning with a hard control method integrated into the sampling process, which guarantees the desired tonal correctness in every generated sample.

Our results in this work are summarized as follows:

- **Motivation**: We theoretically and empirically characterize the challenge of precision in symbolic music generation

[*]Equal contribution [1]University of California, Berkeley, USA [2]New York University, New York, USA [3]Touka Technologies. Correspondence to: Haoyu Liu <haoyuliu@berkeley.edu>, Zeyu Zheng <zyzheng@berkeley.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

[1]The demo page is available at https://huajianduzhuo-code.github.io/FGG-diffusion-music/, we also release the complete source code at https://github.com/huajianduzhuo-code/FGG-music-code

- **Methodology**: We incorporate fine-grained harmonic and rhythmic guidance to symbolic music generation with diffusion models.
- **Functionality**: The developed model is capable of generating music with high accuracy in pitch and consistent rhythmic patterns that align closely with the user's intent.
- **Effectiveness**: We provide both theoretical and empirical evidence supporting the effectiveness of our approach.

1.1. Related Work

Symbolic Music Generation. Symbolic music generation literature can be classified based on the choice of data representation, among which the MIDI token-based representation adopts a sequential discrete data structure, and is often combined with sequential generative models such as Transformers and LSTMs.

To leverage well-developed generative models for symbolic music, Huang et al. (2018) introduced a Transformer-based model with a novel relative attention mechanism designed for symbolic music generation. Subsequent works have enhanced the controllability of symbolic music generation by incorporating input conditions. For instance, Huang & Yang (2020) integrated metrical structures to enhance rhythmic coherence, Ren et al. (2020) conditioned on melody and chord progressions for harmonically guided compositions, and Choi et al. (2020) encoded musical style to achieve nuanced harmonic control. These advancements have contributed to more interpretable and user-directed music generation control.

To better capture spatio-temporal harmonic structures in music, researchers have adopted diffusion models with various control mechanisms. Min et al. (2023) incorporated control signals tailored to diffusion inputs, enabling control over melody, chords, and texture. Wang et al. (2024) extended this by integrating hierarchical control for full-song generation. To further enhance control, Zhang et al. (2023) and Huang et al. (2024) leveraged the gradual denoising process to refine sampling. Building on these approaches, our work addresses the remaining challenge of precise control in real-time generation.

In parallel to diffusion-based approaches, a body of work on general symbolic music generation—using models such as RNNs, GANs, and VAEs—has also explored mechanisms for achieving precise user-controllable generation. Early work on symbolic generation already explored user-steerable conditioning. Meade et al. (2019) retrofitted an RNN method with human-interpretable controls such as note density and pitch range limits. Dong et al. (2017) proposed a method that conditions a GAN model on one track given by human to generate the remaining tracks based on temporal structure of that track. Wu & Yang (2021) utilized a Transformer based VAE model to realize fine-grained style transfer over full songs.

Image Inpainting. Image inpainting with diffusion models has advanced rapidly, offering valuable insights for our task. In our setting, harmonic conditions define constrained or masked regions, and the model must complete the rest—analogous to inpainting. Recent diffusion-based methods have enabled fine-grained control during both training and sampling. For instance, Lugmayr et al. (2022) introduced a post-conditioning strategy that adapts the reverse diffusion process to reconcile known and missing regions without retraining, albeit with increased inference time. Xie et al. (2023) combined shape and text prompts to enable precise, user-guided inpainting via joint training and sampling design. Corneanu et al. (2024) improved sampling efficiency by conditioning directly in the latent space, supporting faster and semantically coherent completions. Inspired by these works, we adapt the idea of context-aware, guided completion to symbolic music, enabling controllable generation over structured time-pitch domains.

Controlled Diffusion Models. Multiple works in controlled diffusion models are related to our work in terms of methodology. Specifically, we adopt the idea of classifier-free guidance in training and generation, see Ho & Salimans (2021). To control the sampling process, Chung et al. (2023), Song et al. (2023) and Novack et al. (2024) guide the intermediate sampling steps using the gradients of a loss function. In contrast, Dhariwal & Nichol (2021), Saharia et al. (2022), Lou & Ermon (2023) and Fishman et al. (2023) apply projection and reflection during the sampling process to straightforwardly incorporate data constraints. Different from these works, we design guidance for intermediate steps tailored to the unique characteristics of symbolic music data and generation. While the meaning of a specific pixel in an image is undefined until the entire image is generated, each position on a piano roll corresponds to a fixed time-pitch pair from the outset. This new context enables us to develop novel implementations and theoretical perspectives on the guidance approach.

2. Background: Diffusion Models for Piano Roll Generation

In this section, we introduce the data representation of piano roll. We then introduce the formulations of diffusion model, combined with an application on modeling the piano roll data.

Data Representation of Piano Rolls. Let $\mathbf{M} \in \{0,1\}^{L\times H}$ be a piano roll segment, where H is the pitch range and L is the number of time units in a frame. For example, H can be set as 128, representing a pitch range of $0-127$, and L as 64, representing a 4-bar segment with time signature 4/4 (4 beats per bar) and 16th-note resolution. Each element M_{lh} of $\mathbf{M}$ ($l \in [\![1, L]\!]$, $h \in [\![1, H]\!]$) takes value 0 or 1, where $\mathrm{M}_{lh} = 1/0$ represents the presence/absence of a note at time index l and pitch h.[2] Since standard diffusion models are based on Gaussian noise, the output of the diffusion model is a continuous random matrix $\mathbf{X} \in \mathbb{R}^{L\times H}$, which is then projected to the discrete piano roll $\mathbf{M}$ by $\mathrm{M}_{lh}(\mathbf{X}) = \mathbf{1}\{\mathrm{X}_{lh} \geq 1/2\}$, where $\mathbf{1}\{\cdot\}$ stands for the indicator function.

Formulation of the Diffusion Model. To model and generate the distribution of $\mathbf{M}$, denoted as $P_{\mathbf{M}}$, we use the the Denoising Diffusion Probabilistic Modeling (DDPM) formulation (Ho et al., 2020). The objective of DDPM training, with specific choices of parameters and reparameterizations, is given as

$$\mathbb{E}_{t\sim\mathcal{U}[\![1,T]\!],\mathbf{X}_0\sim P_{\mathbf{M}},\boldsymbol{\varepsilon}\sim\mathcal{N}(0,\mathbf{I})}[\lambda(t)\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, t)\|^2], \quad (1)$$

where $\boldsymbol{\varepsilon}_\theta$ is a deep neural network with parameter θ. Moreover, according to the connection between diffusion models and score matching (Song & Ermon, 2019), the deep neural network $\boldsymbol{\varepsilon}_\theta$ can be used to derive an estimator of the score function $\boldsymbol{s}_t(\mathbf{X}_t) = \nabla_{\mathbf{X}_t} \log p_t(\mathbf{X}_t)$. Specifically, $\boldsymbol{s}_t(\mathbf{X}_t)$ can be approximated by $-\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, t)/\sqrt{1-\bar{\alpha}_t}$.

With the trained noise prediction network $\boldsymbol{\varepsilon}_\theta$, the reverse sampling process can be formulated as (Song et al., 2021a):

$$\mathbf{X}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\mathbf{X}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, t)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, t) + \sigma_t\boldsymbol{\varepsilon}_t, \quad (2)$$

where σ_t are hyperparameters chosen corresponding to equation 1, and $\boldsymbol{\varepsilon}_t$ is standard Gaussian noise at each step. Going backward in time from $\mathbf{X}_T \sim \mathcal{N}(0, \mathbf{I})$, the process yields the final output $\mathbf{X}_0$, which can be converted into a piano roll $\mathbf{M}(\mathbf{X}_0)$.

According to Song et al. (2021b), the DDPM forward and backward processes can be regarded as discretizations of the following SDEs:

$$d\mathbf{X}_t = -\frac{1}{2}\beta(t)\mathbf{X}_t dt + \sqrt{\beta(t)}d\mathbf{W}_t, \quad (3)$$

$$d\mathbf{X}_t = -\left[\frac{1}{2}\beta(t)\mathbf{X}_t + \beta(t)\boldsymbol{s}_t(\mathbf{X}_t)\right]dt + \sqrt{\beta(t)}d\bar{\mathbf{W}}_t, \quad (4)$$

[2]This is a slightly simplified representation model for the purpose of theoretical analysis, the specified version with implementation details is provided in Section 5.1

3. Methodology: Fine-Grained Guidance

While generative models have achieved significant success in text, image, and audio generation, the effective modeling and generation of symbolic music remains a relatively unexplored area. One challenge of symbolic music generation involves the high-precision requirement in harmony. Unlike image generation, where a slightly misplaced pixel may not significantly affect the overall image quality, an "inaccurately" generated musical note can drastically disrupt the harmony, affecting the quality of a piece.

In this section, we present a control methodology that can precisely achieve the desired harmony. Specifically, we design a fine-grained conditioning and sampling control, altogether referred to as *Fine-Grained Guidance* (FGG) that leverage the characteristic of the piano roll data.

3.1. Fine-Grained Conditioning in Training

We first introduce fine-grained conditioning in training, which serves as the foundation of the more important sampling control in the next subsection 3.2.

We train a conditional diffusion model with fine-grained harmonic ($\mathcal{C}$, required) and rhythmic ($\mathcal{R}$, optional) conditions, which are provided to the diffusion models in the form of a piano roll $\boldsymbol{M}^{\text{cond}}$. We provide illustration of $\boldsymbol{M}^{\text{cond}}(\mathcal{C}, \mathcal{R})$ and $\boldsymbol{M}^{\text{cond}}(\mathcal{C})$ via examples in Figure 1 and Figure 2, respectively. The mathematical descriptions are provided in Appendix B.

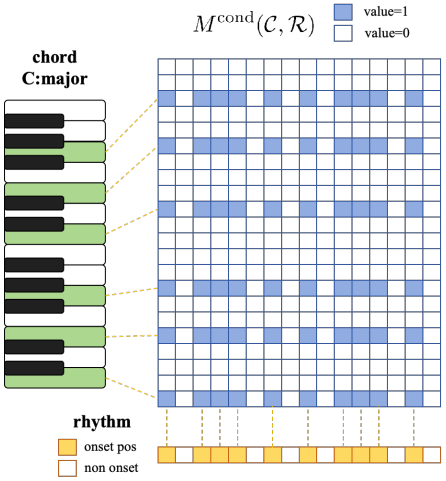


Figure 1. An illustrative example of $\boldsymbol{M}^{\text{cond}}(\mathcal{C}, \mathcal{R})$ with both conditions.

[4]To enable the model to handle both rhythm+chord and chord-only conditions, we use negative values to indicate the absence of rhythmic input when only harmonic conditions are provided, avoiding misinterpretation of 0s and 1s as active constraints. Empirically, removing this distinction (i.e., still using 0 and 1 when

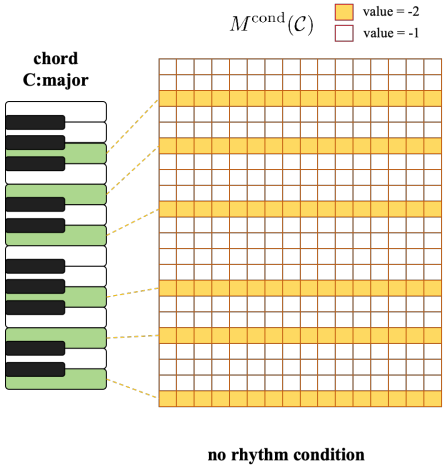


Figure 2. An illustrative example of $\boldsymbol{M}^{\text{cond}}(\mathcal{C})$ with harmonic conditions only.[4]

3.2. Fine-Grained Control in Sampling Process

We first provide a rough idea of the harmonic sampling control. To integrate harmonic constraints into our model, we employ temporary tonic key[5] signatures to establish the tonal center. Our sampling control mechanism guides the gradual denoising process to ensure that the final generated notes remain within a specified set of pitch classes. This control mechanism removes or replaces harmonically conflicting notes, maintaining alignment with the temporary tonic key.

Preliminaries. Recall that a piano roll segment $\mathbf{M} \in \{0,1\}^{L\times H}$, where $l \in [\![1, L]\!]$ is the time index, and $h \in [\![1, H]\!]$ is the pitch index. For given chord condition sequence $\mathcal{C}$, let $\mathcal{K}$ denote the corresponding key sequence. For example, when the C major chord appears as the chord condition at time index l, we would expect $\mathcal{K}(l)$ to contain the pitch classes of the C major scale[6]. We note that $\mathcal{C}$ is essentially different from $\mathcal{K}$, where $\mathcal{C}$ describes chord sequences and is provided as condition for generation, and $\mathcal{K}$ is a restriction of "allowed" pitch classes for sampling refinement.

Let $w(\boldsymbol{l}; \mathcal{K}) := \{l, w(l; \mathcal{K})\}_{l=1}^{L}$denote the undesired pitch positions on the piano roll $\mathbf{M}$. The generated piano roll $\widehat{\mathbf{M}}$ is expected to satisfy $\widehat{\mathbf{M}}_{lh} = 0$, for all $(l, h) \in w(\boldsymbol{l}, \mathcal{K})$. In other words, for $\widehat{\mathbf{X}}_0$ we need

$$\forall (l, h) \in w(\boldsymbol{l}, \mathcal{K}),\ P\left(\widehat{\mathrm{X}}_{0,lh} > 1/2\right) = 0. \tag{5}$$

Note that in the backward sampling equation 2 that derives $\boldsymbol{X}_{t-1}$ from $\boldsymbol{X}_t$, we have for the first term (Song et al., 2021a; Chung et al., 2023)

$$\begin{aligned}&\left(\frac{\boldsymbol{X}_t - \sqrt{1-\bar{\alpha}_t}\widehat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t)}{\sqrt{\bar{\alpha}_t}}\right) = \text{“predicted } \mathbf{X}_0\text{”} \\ &= \widehat{\mathbb{E}}[\mathbf{X}_0|\boldsymbol{X}_t], \quad t = T, T-1, \ldots, 1.\end{aligned} \tag{6}$$

Edit Intermediate-step Outputs of the Sampling Process. The primary cause of inaccurately generated notes is the estimation error of the probability density of $\mathbf{X}_0$, which in turn affects the corresponding score function $\widehat{\boldsymbol{s}}_t(\boldsymbol{X}_t)$. The equivalence $\widehat{\boldsymbol{s}}_t(\boldsymbol{X}_t) = -\widehat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t)/\sqrt{1-\bar{\alpha}_t}$ therefore inspires us to project $\widehat{\mathbb{E}}[\mathbf{X}_0|\boldsymbol{X}_t]$ to the $\mathcal{K}$-constrained domain $\mathbb{R}^{L\times H}\backslash\mathbb{W}_{\mathcal{K}}$ by adjusting the value of $\widehat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t)$. This adjustment is interpreted as an adjustment of the estimated score. Here $\mathbb{W}_{\mathcal{K}}$ is the set of matrices, connected to the set of positions (on the matrix) $w(\boldsymbol{l}, \mathcal{K})$ by

$$\mathbb{W}_{\mathcal{K}} = \left\{\boldsymbol{X} \in \mathbb{R}^{L\times H} |\ \exists (l, h) \in w(\boldsymbol{l}; \mathcal{K}), \boldsymbol{X}_{l,h} > 1/2\right\}.$$

Specifically, at each sampling step t, we replace the guided noise prediction $\widehat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t)$ with $\tilde{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t)$ such that

$$\begin{aligned}\tilde{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t) = &\arg\min_{\boldsymbol{\varepsilon}} \quad \|\boldsymbol{\varepsilon} - \widehat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t)\| \\ &\text{s.t.} \quad \left(\frac{\boldsymbol{X}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}}{\sqrt{\bar{\alpha}_t}}\right) \in \mathbb{R}^{L\times H}\backslash\mathbb{W}'_{\mathcal{K}}.\end{aligned} \tag{7}$$

The element-wise formulation of $\tilde{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t)$ is given as follows, with calculation details provided in Appendix A.1.

$$\begin{aligned}\tilde{\boldsymbol{\varepsilon}}_{\theta,lh}(\boldsymbol{X}_t, t) = &\mathbf{1}\{(l, h) \notin \omega(\boldsymbol{l}; \mathcal{K})\} \cdot \widehat{\boldsymbol{\varepsilon}}_{\theta,lh}(\boldsymbol{X}_t, t) \\ &+\mathbf{1}\{(l, h) \in \omega_{\mathcal{K}}(\boldsymbol{l})\} \cdot \\ &\max\left\{\widehat{\boldsymbol{\varepsilon}}_{\theta,lh}(\boldsymbol{X}_t, t), \frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(X_{t,lh} - \frac{\sqrt{\bar{\alpha}_t}}{2}\right)\right\}.\end{aligned} \tag{8}$$

Plugging the adjusted noise prediction $\tilde{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t)$ into equation 2, we derive the adjusted $\tilde{\mathbf{X}}_{t-1}$. The sampling process is therefore summarized as the following Algorithm 1.

Note that at the final step $t = 0$, the noise correction directly projects $\widehat{\mathbf{X}}_0$ to $\mathbb{R}^{L\times H}\backslash\mathbb{W}'_{\mathcal{K}}$, ensuring the probabilistic constraint 5.

Theoretical Property of the Sampling Control. A natural concern is that enforcing precise fine-grained control over generated samples may disrupt the learned local patterns. The following proposition 1, proved in A.2, provides an upper bound that quantifies this potential effect and address the concern.

rhythmic condition is not provided) led to a 8–15% drop in chord accuracy (e.g., direct chord accuracy drops from 0.485 to 0.421, and chord similarity drops from 0.767 to 0.705), highlighting the importance of explicitly encoding missing rhythmic information.

[5]As a clarification, instead of assigning one single key to a piece or a big section, here we refer to each key associated with the *temporary tonic*.

[6]We note that the correspondence between $\mathcal{C}$ and $\mathcal{K}$ is in fact flexible, and can be designed by the user of the model. More discussion is provided in the next section 4

Algorithm 1 DDPM sampling with fine-grained harmonic control

Input: Input parameters: forward process variances β_t, $\bar{\alpha}_t = \prod_{s=1}^{t} \beta_t$, backward noise scale σ_t, key signature guidance $\mathcal{K}$

Output: generated piano roll $\tilde{\mathbf{M}} \in \{0,1\}^{L \times H}$

$\mathbf{X}_T \sim \mathcal{N}(0, \boldsymbol{I})$;
for $t = T, T-1, \ldots, 1$ **do**
  Compute guided noise prediction $\widehat{\varepsilon}_\theta(\boldsymbol{X}_t, t)$
  Perform noise correction: derive $\tilde{\varepsilon}_\theta(\boldsymbol{X}_t, t)$ using equation 8
  Compute $\tilde{\mathbf{X}}_{t-1}$ by plugging the corrected noise $\tilde{\varepsilon}_\theta(\boldsymbol{X}_t, t)$ into equation 2
end
Convert $\tilde{\mathbf{X}}_0$ into piano roll $\tilde{\mathbf{M}}$
return *output*

Proposition 1. *Under the SDE formulation in equation 3 and equation 4, given an early-stopping time t_0[7], if*

$$\mathbb{E}_{\mathbf{X}_t \sim p_t}[\|\boldsymbol{\varepsilon}^*(\mathbf{X}_t, t) - \boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, t)\|^2] \leq \delta \tag{9}$$

for all t, where $\boldsymbol{\varepsilon}^*(\mathbf{X}_t, t)$ is the optimal solution of the DDPM training objective (1), then we have

$$KL(\tilde{p}_{t_0} | p_{t_0}) \leq \frac{\delta}{2} \int_{t_0}^{T} \frac{\beta(t)}{\sqrt{1 - e^{-\int_{t_0}^{t} \beta(s) ds}}} dt,$$

$$KL(\tilde{p}_{t_0} | \hat{p}_{t_0}) \leq \frac{\delta}{2} \int_{t_0}^{T} \frac{\beta(t)}{\sqrt{1 - e^{-\int_{t_0}^{t} \beta(s) ds}}} dt,$$

where p_{t_0} is the distribution of $\mathbf{X}_{t_0}$ in the forward process, $\hat{p}_{t_0}$ is the distribution of $\widehat{\mathbf{X}}_{t_0}$ generated by the diffusion sampling process without noise adjustment, and $\tilde{p}_{t_0}$ is the distribution of $\tilde{\mathbf{X}}_{t_0}$ generated by the fine-grained noise adjustment.

Proposition 1 provides upper bounds for the distance between the controlled distribution and the uncontrolled distribution, as well as between the controlled distribution and the ground truth. We remark that, our method can shape the output towards a specific tonal quality. This can be for example using the Dorian scale as the key signature sequence $\mathcal{K}$ to shape the generated music towards the Dorian mode (a tonal framework not present in the training data), where the generated distribution $\tilde{p}$ with fine-grained noise adjustment is fundamentally different from the ground truth distribution p. Nevertheless, Proposition 1 guarantees a substantial overlap between the two distributions $\tilde{p}$ and p, demonstrating a well-balanced interplay between external control and the model's internal learning from the training data, e.g., melodic lines. This theoretical insight aligns with our empirical observations, which is presented in the "Mode Change" section of the demo page.

[7]We adopt the early-stopping time to avoid the blow-up of score function, which is standard in many literature (Song & Ermon, 2020; Nichol & Dhariwal, 2021)

4. Challenges for Uncontrolled Symbolic Music Generation Models

In the previous section 3, we present our FGG method that guarantees the precision of generation. But why is it meaningful to provide such guarantee in the task of symbolic music generation? Why is it hard for models to self-ensure harmonic precision without having the hard sampling control? We use Section 4 to answer these questions. These discussions further motivate and justify the importance of the FGG method.

In the rest of this section, we focus our discussion to tonic-centric genres. Although not covering every aspect of music, it still spans a wide range of genres that are deeply embedded in everyday life, including tonic-centric New Age music, light classical music, and tonic-focused movie soundtracks. Such genres rely heavily on *harmony*, i.e., the simultaneous sound of different notes that form a cohesive entity in the mind of the listener (Müller, 2015).

Using the concept of temporary tonic key signatures we discussed in the previous section, we focus our discussion on the presence of out-of-key notes[8] in generated music. In the tonic-centric genres, out-of-key notes are uncommon, and produce noticeable dissonance, if not having a "resolution". We often notice that out-of-key notes are usually perceived merely as mistakes when appearing in generative model outputs, as demonstrated by some examples on our demo page.

We aim to explain why the existence of out-of-key notes is an issue for diffusion-based symbolic music generation models in the tonic-centric genres. Specifically, we explain the following phenomenon: Suppose $\mathcal{G}$ is a diffusion model trained to generate tonic-centric genres. In the target data distribution, out-of-key notes appear at a small rate $P(O) \gtrsim 0$. These out-of-key notes are carefully managed (by expert composers) in the training set. However, when out-of-key notes appear in the generated samples of $\mathcal{G}$, they often lack an appropriate resolution and are more likely to be perceived negatively. Why does the model often fail to learn this nuance?

We provide an intuitive explanation using statistical reasoning. Consider a piano roll segment, represented as a

[8]For instance, a G♮ note is considered as out-of-key in a G♭ major context. Admittedly the inference of temporary tonic key is even more vague than chord recognition, due to the flexibility of harmony. However, in the following discussion, we assume that the temporary tonic key is specified.

random variable $\mathbf{M}$. Suppose we are interested in whether this segment contains an out-of-key note (denoted as event $\{O\}$) and whether that note is eventually resolved within the segment (denoted as event $\{R, O\}$). In our training data, almost every out-of-key note is resolved, meaning the probability of unresolved out-of-key note is close to 0, i.e., $P(\bar{R}, O) \approx 0$.

Now, we examine the probability in the generated music. The key question then is whether the generative model also learns to keep $\widehat{P}(\bar{R}, O)$ small. The following proposition 2 leverages analysis of statistical errors to show that $\widehat{P}(\bar{R}, O)$ can decrease slowly as the dataset size n increases.

Proposition 2. *Consider generating piano roll* $\mathbf{M}$ *from a continuous random variable* $\mathbf{X}$*, i.e., given* n *i.i.d. data* $\{\mathbf{X}^i\}_{i=1}^n \sim p_{\mathbf{X}}$*, let* $\{\mathbf{M}^i\}_{i=1}^n$ *be given by* $\mathrm{M}_{lh}^i = \mathbf{1}\{\mathrm{X}_{lh}^i \geq 1/2\}$*. Denote the model for estimating the distribution of* $\mathbf{X}$ *as* $\widehat{p}_{\mathbf{X}}$*. We have* $\exists\, C > 0$ *such that* $\forall n$*,*

$$\inf_{\widehat{p}_{\mathbf{X}}} \sup_{p_{\mathbf{X}} \in \mathcal{P}_\delta} \mathbb{E}_{\{\mathbf{M}^i\}_{i=1}^n} \widehat{P}(\bar{R}, O) \geq C \cdot n^{-\frac{1}{LH+2}} - P(\bar{R}, O), \tag{10}$$

where $\widehat{P}$ *is the probability associated with the generated data* $\widehat{p}_{\mathbf{X}}$*.*

The proof of proposition 2 is provided in appendix A.3. The term $\sup_{p_{\mathbf{X}} \in \mathcal{P}_\delta}$ is the supremum taken over the search space of the continuous generative model[9], and $\inf_{\widehat{p}_{\mathbf{X}}}$ denotes the best possible realization of the model. The minimax formulation is standard in works that discuss statistical convergence of generative models (Fu et al., 2024).

The theoretical insights presented in proposition 2 demonstrate that the occurrence of unsolved out-of-key note is often unavoidable, and the decay rate of this error probability with respect to training set size n is slow $O(n^{-1/(LH)})$. Thus, relying on the model itself for precision is challenging for existing models, given the inherent scarcity of high-quality data and the slow decay rate of errors. There are two implications following this line: First, it would be immensely valuable to develop a model that enjoys the ability to implicitly learn contextually appropriate out-of-key notes (nevertheless, currently in our work we did not take this path). Second, with the fact that symbolic music generation requires an exceptional level of precision, it is worthwhile to develop methods that enable the model to function as a well-controlled collaborative tool to aid human composers.

[9]The exact formulation of P_δ is given in appendix A.3. While real life distribution classes associated with generative models are more complicated and difficult to analyze, P_δ essentially captures their characteristics, and is therefore comparable to them. This type of simplification while maintaining core characteristics appears to be standard in works that provide theoretical insights (Fu et al., 2024).

5. Experiments

In this section, we present experiments to demonstrate the effectiveness of our fine-grained guidance approach. We additionally create a demopage[10] for demonstration, which allows for fast and stable interactive music creation with user-specified input guidance[11], and even for generating music based on tonal frameworks absent from the training set.

5.1. Numerical Experiments

We present numerical experiments on accompaniment generation given both melody and chord generation, or symbolic music generation given only chord conditions. We focus on the former one as it provides a more effective basis for comparison. Due to page limits, we put the results and more detailed explanation of the latter one in Appendix D.3. For the accompaniment generation task, we compare with two state-of-the-art baselines: 1) WholeSongGen (Wang et al., 2024) and 2) GETMusic (Lv et al., 2023).

5.1.1. Data Representation and Model Architecture

The generation target $\boldsymbol{X}$ is represented by a piano-roll matrix of shape $2 \times L \times 128$ under the resolution of a 16th note, where L represents the total length of the music piece, and the two channels represent note onset and sustain, respectively. In our experiments, we set $L = 64$, corresponding to a 4-measure piece under time signature $4/4$. Longer pieces can be generated autoregressively using the inpainting method. The backbone of our model is a 2D UNet with spatial attention.

The condition matrix $\boldsymbol{M}^{\text{cond}}$ is also represented by a piano roll matrix of shape $2 \times L \times 128$, with the same resolution and length as that of the generation target $\boldsymbol{X}$. For the accompaniment generation experiments, we provide melody as an additional condition. Detailed construction of the condition matrices are provided in Appendix D.1.

5.1.2. Dataset

We use the POP909 dataset (Wang et al., 2020a) for training and evaluation. This dataset consists of 909 MIDI pieces of pop songs, each containing lead melodies, chord progression, and piano accompaniment tracks. We exclude 29 pieces that are in triple meter. 90% of the data are used to train our model, and the remaining 10% are used for

[10]See https://huajianduzhuo-code.github.io/FGG-diffusion-music/. We note that slow performance may result from Huggingface resource limitations and network latency.

[11]The format of user-specified input guidance is limited within the constrained piano roll format, as is demonstrated in the paper.

evaluation. In the training process, we split all the midi pieces into 4-measure non-overlapping segments (corresponding to $L = 64$ under the resolution of a 16th note), which in total generates 15761 segments in the entire training set. Training and sampling details are provided in Appendix D.2.

5.1.3. Task and Baseline Models

We consider accompaniment generation task based on melody and chord progression. We compare the performance of our model with two baseline models: 1) WholeSongGen (Wang et al., 2024) and 2) GETMusic (Lv et al., 2023). WholeSongGen is a hierarchical music generation framework that leverages cascaded diffusion models to generate full-length pop songs. It introduces a four-level computational music language, with the last level being accompaniment. The model for the last level can be directly used to generate accompaniment given music phrases, lead melody, and chord progression information. GETMusic is a versatile music generation framework that leverages a discrete diffusion model to generate tracks based on flexible source-target combinations. The model can also be directly applied to generate piano accompaniment conditioning on melody and chord. Since these baseline models do not support rhythm control, to ensure comparability, we will use the $\boldsymbol{M}^{\text{cond}}(\mathcal{C})$ without rhythm condition in our model.

5.1.4. Evaluation

We generate accompaniments for the 88 MIDI pieces in our evaluation dataset.[12] We introduce the following objective metrics to evaluate the generation quality of different methods:

(1) *Percentage of Out-of-Key Notes* First, for each method, we present the frequency of out-of-key notes by computing the percentage of steps in the generated sequences containing at least one out-of-key note, where each step corresponds to a 16th note. The results, presented in Table 1, indicate that frequency of out-of-key notes in the baselines is roughly 2%-4%, equating to about 1–3 occurrences in a 4-measure piece. In contrast, our sampling control method effectively eliminates such dissonant notes in the generated samples.

(2) *Direct Chord Accuracy* and *Chord Progression Similarity* We evaluate harmonic consistency by comparing the chord progressions of the generated and ground truth accompaniments. Chords are extracted using the rule-based recognition method from Dai et al. (2020). Direct chord accuracy is computed as the percentage of beats where the recognized chord of the generated output exactly matches that of the ground truth. However, since not all mismatches reflect equal harmonic deviation—for instance, C major is harmonically close to Cmaj7 but far from B major—direct accuracy may fail to reflect the nuanced similarity between chords.

To address this, we further assess chord progression similarity. We divide each accompaniment into non-overlapping 2-measure segments and encode them into a 256-dimensional latent space using a pre-trained disentangled VAE (Wang et al., 2020b). Cosine similarity is then computed between corresponding segments from the generated and ground truth progressions. Table 1 reports the average direct accuracy and average latent similarity, along with their 95% confidence intervals. The results demonstrate that our method significantly outperforms all baselines in chord accuracy.

(3) *Intersection over Union (IoU) Metrics.* We evaluate the similarity between the generated and ground truth accompaniments using two IoU-based metrics: *IoU of Chords* and *IoU of Piano Roll*. For *IoU of Chords*, we first apply the chord recognition method from Dai et al. (2020) to both the generated and ground truth accompaniments. Each chord is then represented as a 12-dimensional binary vector indicating the presence of pitch classes (C through B). We compute the IoU between the generated and ground truth pitch class sets at every 16th-note time step and report the average IoU across all time steps.

For *IoU of Piano Roll*, we represent each accompaniment as a binary piano roll. The IoU is computed at each 16th-note time step by comparing the sets of active pitches in the generated and ground truth piano rolls. We then report the average IoU across all time steps.[13] The results, presented in Table 1, show that our method consistently achieves higher IoU scores than the baselines, indicating closer alignment to the ground truth at both the harmonic and note level.

(4) Subjective Evaluation

To compare performance of our FGG method against the baselines (ground truth, WholeSongGen, and GETMusic), we prepared 6 sets of generated samples, with each set containing the melody paired with accompaniments generated by FGG, WholeSongGen, and GETMusic, along with the ground truth accompaniment. This yields a total of $6 \times 4 = 24$ samples. The samples are presented in a randomized order, and their sources are not disclosed

[12] The WholeSongGen model from Wang et al. (2024) is also trained on the POP909 dataset. Our evaluation set is a subset of their test set so there is no in-sample evaluation issue on their model.

[13] While exact agreement with the ground truth is not necessarily optimal—since a given melody may admit multiple valid accompaniments—the IoU still serves as a useful indicator of musical quality. A high-quality accompaniment is expected to align reasonably well with the expert-written ground truth, and thus should not deviate substantially.

Methods	% Out-of-Key Notes ↓	Direct Chord Accuracy ↑	Chord Similarity ↑	IoU (Chord) ↑	IoU (Piano Roll) ↑
FGG (Ours)	0.0%	$\mathbf{0.485 \pm 0.006}$	$\mathbf{0.767 \pm 0.007}$	$\mathbf{0.769 \pm 0.003}$	$\mathbf{0.281 \pm 0.005}$
WholeSongGen	2.1%	0.314 ± 0.006	0.611 ± 0.010	0.618 ± 0.004	0.107 ± 0.003
GETMusic	3.5%	0.153 ± 0.007	0.394 ± 0.012	0.412 ± 0.007	0.048 ± 0.003

Table 1. Evaluation of the similarity with ground truth for all methods.

to participants. Experienced listeners assess the quality of samples in 5 dimensions: creativity, harmony (whether the accompaniment is in harmony with the melody), melodiousness, naturalness and richness, together with an overall assessment. The results are shown in Figure 3. The bar height shows the mean rating, and the error bar shows the 95% confidence interval. FGG consistently outperforms the baselines in all dimensions. For details of our survey, please see Appendix F.

5.1.5. Ablation Study

In this section, we conduct ablation studies to better illustrate the effectiveness of our FGG method. We aim to demonstrate the effectiveness of both the fine-grained training condition (training control) and the sampling control. We also compare with simple rule-based post-sample editing[14]. The former leverages the structured gradual denoising process of diffusion models, ensuring a theoretical guarantee of preserving the distributional properties of the original learned distribution. In contrast, the latter employs a brute-force editing approach that disrupts the generated samples, affecting local melodic lines and rhythmic patterns. The numerical results further validate this analysis.

Moreover, we compare with a so-called "Inpainting" method, which treats the pixels where there is not supposed to be a note as 0 and inpaints the remaining pixels. This information is included by adding a mask channel in the training process. We still allow for the fine-grained conditioning in training.

Specifically, we include the following variants in our ablation study:

- **Training and Sampling Control**: our full method, which applies fine-grained conditioning during both training and sampling.
- **Inpainting**: out-of-key pixels are treated as 0, and the remaining positions are inpainted based on fixed context.
- **Training control + Round Notes Up/Down After Sampling**: training control is applied, and out-of-key notes are corrected post-sampling by rounding to the nearest in-key pitch.
- **Training control + Remove Wrong Notes After Sampling**: training control is applied, and out-of-key notes are corrected post-sampling by removing them.
- **Training Control Only**: only training control is used; no sampling-time controls are enforced.
- **No Control**: neither training control nor sampling control are applied.

We assess overall model performance using the same quantitative metrics as in the previous section. The results are shown in Table 2. In general, fine-grained conditioning (i.e., training control) leads to substantial improvements across all evaluation metrics, and adding sampling control further enhances performance. While rule-based post-sampling editing (e.g., removing or rounding out-of-key notes) yields moderate gains, it is consistently outperformed by our fine-grained sampling control method. Our approach fully leverages the structured, gradual denoising process of diffusion models, allowing the model to iteratively correct or replace errors while preserving the coherence of the original learned distribution.

Moreover, our method outperforms the inpainting baseline across all evaluation metrics. Unlike our approach, the inpainting method introduces additional architectural complexity by requiring the model to handle an auxiliary mask channel that indicates which pixels to regenerate. This not only increases implementation overhead but also adds computational burden during both training and inference.

5.2. Empirical Observations

Notably, harmonic control not only helps the model eliminate incorrect notes, but also guides it to replace them with correct ones. Such representative examples are presented in Appendix G. Moreover, samples generated from ablation conditions are available in Section 3 of our demo page[15]. Across all ablations, we observed occasional occurrences of excessively high-pitched notes and overly dense note clusters.

[14]Specifically, we compare with two rule-based post-sample editing methods: 1) Rounding wrong notes up/down and 2) Remove wrong notes.

[15]The demo page is available at https://huajianduzhuo-code.github.io/FGG-diffusion-music/

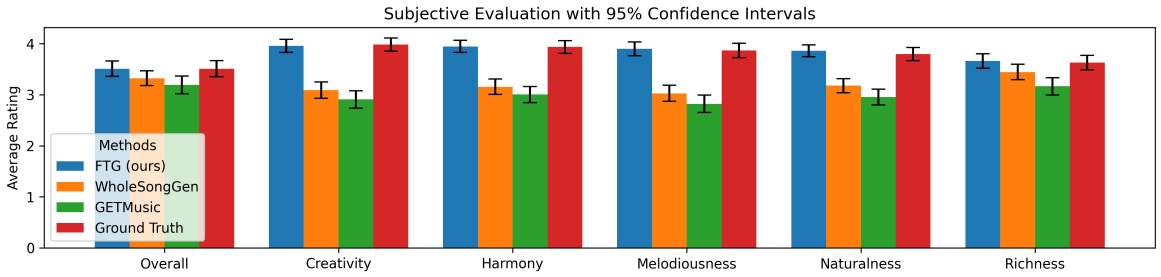


Figure 3. Subjective evaluation results on music quality.

Methods	% Out-of-Key Notes	Direct Chord Accuracy	Chord Similarity	IoU Chord	IoU Piano Roll
Training and Sampling Control	0.0%	**0.485 ±0.006**	**0.767 ±0.007**	**0.769 ±0.003**	**0.281 ±0.005**
Inpainting	0.0%	0.458 ±0.006	0.710 ±0.008	0.743 ±0.003	0.271 ±0.005
Training Control Round Notes Up/Down After Sampling	0.0%	0.472 ±0.006	0.756 ±0.007	0.763 ±0.003	0.272 ±0.005
Training Control Remove Wrong Notes After Sampling	0.0%	0.482 ±0.006	0.763 ±0.007	0.767 ±0.003	0.277 ±0.005
Only Training Control	3.7%	0.465 ±0.006	0.748 ±0.007	0.757 ±0.003	0.270 ±0.005
No Control	10.1%	0.112 ±0.004	0.378 ±0.007	0.378 ±0.004	0.072 ±0.002

Table 2. Ablation study.

6. Limitations and Future Work

While our method achieves strong performance across multiple metrics, several limitations remain. First, we adopt a 16th-note quantization scheme following Wang et al. (2024), which simplifies temporal representation but restricts rhythmic flexibility and precludes training on data without explicit beat annotations. A promising future direction is to integrate our pitch-class-based control mechanism with approaches such as Huang et al. (2024), which introduce a dynamic dimension and utilize finer 10ms time quantization to better capture expressive timing variations. Second, our method supports pitch class and rhythmic control in the piano roll representation, but does not accommodate more abstract forms or probabilistic control. Finally, we note that evaluation remains a broader challenge across the field of symbolic music generation. Since musical quality evaluation is inherently detailed and partly subjective, objective evaluation metrics such as rule-based and structural evaluation methods used in this work have inherent limitations in reflecting perceptual or creative aspects of music. This is a key reason why many recent studies supplement objective evaluation with subjective human listening evaluations. A valuable future direction is to develop improved automatic evaluation metrics that more faithfully align with human judgments of musicality and creativity.

7. Conclusion

In this work, we apply fine-grained textural guidance (FGG) on symbolic music generation models. We provide theoretical analysis and empirical evidence to highlight the need for fine-grained and precise control over the model output. We also provide theoretical analysis to quantify and upper bound the potential effect of fine-grained control on learned local patterns, and provide samples and numerical results for demonstrating the effectiveness of our approach. For the impact of our method, we note that the FGG method can be integrated with other diffusion-based symbolic music generation methods. With a moderate trade-off of flexibility, the FGG method prioritizes real-time generation stability and enables efficient generation with precise control.

Acknowledgements

The authors gratefully acknowledge Jinghai He, Ang Lv, Yifu Tang, Gus Xia, Yaodong Yu, Yufeng Zheng, anonymous reviewers, area chairs, and the anonymous evaluators of this work's demos.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are a range of potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M.-H., Murphy, K., Freeman, W. T., Rubinstein, M., et al. Muse: Text-to-image generation via masked generative transformers. In *International Conference on Machine Learning*, pp. 4055–4075, 2023.

Chen, S., Chewi, S., Li, J., Li, Y., Salim, A., and Zhang, A. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *International Conference on Learning Representations*, 2023.

Choi, K., Hawthorne, C., Simon, I., Dinculescu, M., and Engel, J. Encoding musical style with transformer autoencoders. In *International Conference on Machine Learning*, pp. 1899–1908. PMLR, 2020.

Chung, H., Kim, J., McCann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

Corneanu, C. A., Gadde, R., and Martínez, A. M. Latentpaint: Image inpainting in latent space with diffusion models. *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4322–4331, 2024.

Dai, S., Zhang, H., and Dannenberg, R. B. Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music. *arXiv preprint arXiv:2010.07518*, 2020.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Dong, H.-W., Hsiao, W.-Y., Yang, L.-C., and Yang, Y.-H. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *AAAI Conference on Artificial Intelligence*, 2017.

Fishman, N., Klarner, L., De Bortoli, V., Mathieu, E., and Hutchinson, M. J. Diffusion models for constrained domains. *Transactions on Machine Learning Research (TMLR)*, 2023.

Fu, H., Yang, Z., Wang, M., and Chen, M. Unveil conditional diffusion models with classifier-free guidance: A sharp statistical theory. *arXiv preprint arXiv:2403.11968*, 2024.

Gao, S., Zhou, P., Cheng, M.-M., and Yan, S. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23164–23173, 2023.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N. M., Simon, I., Hawthorne, C., Dai, A. M., Hoffman, M. D., Dinculescu, M., and Eck, D. Music transformer: Generating music with long-term structure. In *International Conference on Learning Representations*, 2018.

Huang, Y., Ghatare, A., Liu, Y., Hu, Z., Zhang, Q., Sastry, C. S., Gururani, S., Oore, S., and Yue, Y. Symbolic music generation with non-differentiable rule guided diffusion. In *International Conference on Machine Learning*, pp. 19772–19797. PMLR, 2024.

Huang, Y.-S. and Yang, Y.-H. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM international conference on multimedia*, pp. 1180–1188, 2020.

Karatzas, I. and Shreve, S. *Brownian motion and stochastic calculus*, volume 113. Springer Science & Business Media, 1991.

Lin, S., Liu, B., Li, J., and Yang, X. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 5404–5411, 2024.

Lou, A. and Ermon, S. Reflected diffusion models. In *International Conference on Machine Learning*, pp. 22675–22701. PMLR, 2023.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Gool, L. V. Repaint: Inpainting using denoising diffusion probabilistic models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461, 2022.

Lv, A., Tan, X., Lu, P., Ye, W., Zhang, S., Bian, J., and Yan, R. Getmusic: Generating any music tracks with a unified representation and diffusion framework. *arXiv preprint arXiv:2305.10841*, 2023.

Meade, N., Barreyre, N., Lowe, S. C., and Oore, S. Exploring conditioning for generative music systems with human-interpretable controls. In *Proceedings of the 10th International Conference on Computational Creativity (ICCC)*, Charlotte, North Carolina, 2019.

Min, L., Jiang, J., Xia, G., and Zhao, J. Polyffusion: A diffusion model for polyphonic score generation with internal and external controls. *Proceedings of 24th International Society for Music Information Retrieval Conference, ISMIR*, 2023.

Müller, M. *Fundamentals of music processing: Audio, analysis, algorithms, applications*, volume 5. Springer, 2015.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Novack, Z., McAuley, J., Berg-Kirkpatrick, T., and Bryan, N. J. Ditto: Diffusion inference-time t-optimization for music generation. In *International Conference on Machine Learning*, pp. 38426–38447. PMLR, 2024.

Ren, Y., He, J., Tan, X., Qin, T., Zhao, Z., and Liu, T.-Y. Popmag: Pop music accompaniment generation. In *Proceedings of the 28th ACM international conference on multimedia*, pp. 1198–1206, 2020.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Song, J., Zhang, Q., Yin, H., Mardani, M., Liu, M.-Y., Kautz, J., Chen, Y., and Vahdat, A. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, pp. 32483–32498. PMLR, 2023.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. *Advances in Neural Information Processing Systems*, 33:12438–12448, 2020.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

von Rütte, D., Biggio, L., Kilcher, Y., and Hofmann, T. Figaro: Controllable music generation using learned and expert features. In *International Conference on Learning Representations*, 2023.

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

Wang, Z., Wang, D., Zhang, Y., and Xia, G. Learning interpretable representation for controllable polyphonic music generation. *Proceedings of 21st International Society for Music Information Retrieval Conference, ISMIR*, 2020b.

Wang, Z., Min, L., and Xia, G. Whole-song hierarchical generation of symbolic music using cascaded diffusion models. In *International Conference on Learning Representations*, 2024.

Wu, S.-L. and Yang, Y.-H. Musemorphose: Full-song and fine-grained piano music style transfer with one transformer vae. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1953–1967, 2021.

Xie, S., Zhang, Z., Lin, Z., Hinz, T., and Zhang, K. Smartbrush: Text and shape guided object inpainting with diffusion model. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22428–22437, 2023.

Zhang, C., Ren, Y., Zhang, K., and Yan, S. Sdmuse: Stochastic differential music editing and generation via hybrid representation. *IEEE Transactions on Multimedia*, 2023.

A. Proof of propositions and calculation details

A.1. Calculation details in 3.2

Our goal is to find the optimal solution of problem (7). Since the constraint is an element-wise constraint on a linear function of $\boldsymbol{\varepsilon}$ and the objective is separable, we can find the optimal solution by element-wise optimization. Consider the (l,h)-element of $\boldsymbol{\varepsilon}$.

First, if $(l,h) \notin w(\boldsymbol{l};\mathcal{K})$, there is no constraint on $\boldsymbol{\varepsilon}_{lh}$. Therefore, the optimal solution of $\boldsymbol{\varepsilon}_{lh}$ is $\widehat{\boldsymbol{\varepsilon}}_{\theta,lh}(\boldsymbol{X}_t,t)$.

If $(l,h) \in w(\boldsymbol{l};\mathcal{K})$, the constraint on $\boldsymbol{\varepsilon}_{lh}$ is

$$X_{t,lh} - \frac{\sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}_{lh}}{\sqrt{\bar{\alpha}_t}} \leq \frac{1}{2},$$

which is equivalent to

$$\boldsymbol{\varepsilon}_{lh} \geq \frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(X_{t,lh} - \frac{\sqrt{\bar{\alpha}_t}}{2}\right).$$

The objective is to minimize $\|\boldsymbol{\varepsilon}_{lh} - \widehat{\boldsymbol{\varepsilon}}_{\theta,lh}(\boldsymbol{X}_t,t)\|$. Therefore, the optimal solution of $\boldsymbol{\varepsilon}_{lh}$ is

$$\boldsymbol{\varepsilon}_{lh} = \max\left\{\widehat{\boldsymbol{\varepsilon}}_{\theta,lh}(\boldsymbol{X}_t,t|\mathcal{C},\mathcal{R}), \frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(X_{t,lh} - \frac{\sqrt{\bar{\alpha}_t}}{2}\right)\right\}.$$

A.2. Proof of Proposition 1

Proof. Recall that According to Song et al. (2021b), the DDPM forward process $\mathbf{X}_t = \sqrt{\bar{\alpha}_t}\mathbf{X}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}$ can be regarded as a discretization of the following SDE:

$$d\mathbf{X}_t = -\frac{1}{2}\beta(t)\mathbf{X}_t dt + \sqrt{\beta(t)}d\mathbf{W}_t,$$

and the corresponding denoising process takes the form of a solution to the following stochastic differential equation (SDE):

$$d\mathbf{X}_t = -\left[\frac{1}{2}\beta(t)\mathbf{X}_t + \beta(t)\nabla_{\mathbf{X}_t}\log p_t(\mathbf{X}_t)\right]dt + \sqrt{\beta(t)}d\bar{\mathbf{W}}_t,$$

where $\beta(t/T) = T\beta_t$ as T goes to infinity, $\bar{\mathbf{W}}_t$ is the reverse time standard Wiener process, and $\bar{\alpha}_t$ term should be replaced by its continuous version $e^{-\int_0^t \beta(s)ds}$ (or $e^{-\int_{t_0}^t \beta(s)ds}$ when early-stopping time t_0 is adopted). The score function $\nabla_{\mathbf{X}_t}\log p_t(\mathbf{X}_t)$ can be approximated by $-\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t,t)/\sqrt{1-e^{-\int_0^t \beta(s)ds}}$.

Under the SDE formulation, the denoising process can take the form of a solution to stochastic differential equation (SDE):

$$d\mathbf{X}_t = -\left[\frac{1}{2}\beta(t)\mathbf{X}_t + \beta(t)\nabla_{\mathbf{X}_t}\log p_t(\mathbf{X}_t)\right]dt + \sqrt{\beta(t)}d\bar{\mathbf{W}}_t, \tag{11}$$

where $\beta(t/T) = T\beta_t$, $\bar{\mathbf{W}}_t$ is the reverse time standard Wiener process. According to Song et al. (2021b), as $T \to \infty$, the solution to the SDE converges to the real data distribution p_0.

In the diffusion model, $\nabla_{\mathbf{X}_t}\log p_t(\mathbf{X}_t)$ is approximated by $-\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t,t)/\sqrt{1-e^{-\int_{t_0}^t \beta(s)ds}}$. Therefore, the approximated reverse-SDE sampling process without harmonic guidance is

$$d\hat{\mathbf{X}}_t = -\left[\frac{1}{2}\beta(t)\hat{\mathbf{X}}_\mathrm{t} - \beta(\mathrm{t})\frac{\boldsymbol{\varepsilon}_\theta(\hat{\mathbf{X}}_\mathrm{t},\mathrm{t})}{\sqrt{1-\mathrm{e}^{-\int_{\mathrm{t}_0}^{\mathrm{t}}\beta(\mathrm{s})\mathrm{ds}}}}\right]dt + \sqrt{\beta(t)}d\bar{\mathbf{W}}_t. \tag{12}$$

Similarly, the sampling process with fine-grained harmonic guidance is

$$d\tilde{\mathbf{X}}_t = -\left[\frac{1}{2}\beta(t)\tilde{\mathbf{X}}_\mathrm{t} - \beta(\mathrm{t})\frac{\tilde{\boldsymbol{\varepsilon}}_\theta(\tilde{\mathbf{X}}_\mathrm{t},\mathrm{t})}{\sqrt{1-\mathrm{e}^{-\int_{\mathrm{t}_0}^{\mathrm{t}}\beta(\mathrm{s})\mathrm{ds}}}}\right]dt + \sqrt{\beta(t)}d\bar{\mathbf{W}}_t, \tag{13}$$

where $\tilde{\varepsilon}_\theta$ is defined as equation 7 and equation 8.

For simplicity, we denote the drift terms as follows:

$$f(\mathbf{X}_t, t) = -\left[\frac{1}{2}\beta(t)\mathbf{X}_t + \beta(t)\nabla_{\mathbf{X}_t}\log p_t(\mathbf{X}_t)\right]$$
$$\hat{f}(\hat{\mathbf{X}}_t, t) = -\left[\frac{1}{2}\beta(t)\hat{\mathbf{X}}_\mathrm{t} - \beta(\mathrm{t})\frac{\boldsymbol{\varepsilon}_\theta(\hat{\mathbf{X}}_\mathrm{t}, \mathrm{t})}{\sqrt{1-\mathrm{e}^{-\int_{\mathrm{t}_0}^{\mathrm{t}}\beta(\mathrm{s})\mathrm{ds}}}}\right],$$
$$\tilde{f}(\tilde{\mathbf{X}}_t, t) = -\left[\frac{1}{2}\beta(t)\tilde{\mathbf{X}}_\mathrm{t} - \beta(\mathrm{t})\frac{\tilde{\boldsymbol{\varepsilon}}_\theta(\tilde{\mathbf{X}}_\mathrm{t}, \mathrm{t})}{\sqrt{1-\mathrm{e}^{-\int_{\mathrm{t}_0}^{\mathrm{t}}\beta(\mathrm{s})\mathrm{ds}}}}\right].$$

Since

$$\mathbb{E}_{\mathbf{X}_t\sim p_t}[\|\boldsymbol{\varepsilon}^*(\mathbf{X}_t, t) - \boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, t)\|^2] \le \delta,$$

and

$$\boldsymbol{\varepsilon}^*(\mathbf{X}_t, t) = -\sqrt{1-e^{-\int_{t_0}^t \beta(s)ds}}\nabla_{\mathbf{X}_t}\log p_t(\mathbf{X}_t),$$

we have

$$\mathbb{E}_{\mathbf{X}\sim p_t}[\|f(\mathbf{X}, t) - \hat{f}(\mathbf{X}, t)\|] \le \frac{\beta(t)}{\sqrt{1-e^{-\int_{t_0}^t \beta(s)ds}}}\delta.$$

Now we consider $\tilde{\varepsilon}_\theta(\tilde{\mathbf{X}}_t, t)$, which is the solution of the optimization problem (7). In the continuous SDE case, the corresponding optimization problem becomes

$$\begin{aligned} \min_{\boldsymbol{\varepsilon}} \quad & \|\boldsymbol{\varepsilon} - \widehat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t|\mathcal{C}, \mathcal{R})\| \\ \text{s.t.} \quad & \left(\frac{\boldsymbol{X}_t - \sqrt{1-e^{-\int_{t_0}^t \beta(s)ds}}\boldsymbol{\varepsilon}}{e^{-\frac{1}{2}\int_{t_0}^t \beta(s)ds}}\right) \in \mathbb{R}^{L\times H}\backslash\mathbb{W}'_{\mathcal{K}}. \end{aligned} \tag{14}$$

According to Proposition 1 of Chung et al. (2023), the posterior mean of $\mathbf{X}_0$ conditioning on $\mathbf{X}_t$ is

$$\begin{aligned} \mathbb{E}[\mathbf{X}_0|\mathbf{X}_t] &= \frac{1}{e^{-\frac{1}{2}\int_{t_0}^t \beta(s)ds}}\left(\mathbf{X}_t + (1-e^{-\frac{1}{2}\int_{t_0}^t \beta(s)ds})\nabla_{\mathbf{X}_t}\log p_t(\mathbf{X}_t)\right) \\ &= \frac{1}{e^{-\frac{1}{2}\int_{t_0}^t \beta(s)ds}}\left(\mathbf{X}_t - \sqrt{1-e^{-\int_{t_0}^t \beta(s)ds}}\boldsymbol{\varepsilon}^*(\mathbf{X}_t, t)\right). \end{aligned}$$

Since the domain of $\mathbf{X}_0$ is $R^{L\times H}\backslash\mathbb{W}'_{\mathcal{K}}$, which is a convex set, we know that the posterior mean $\mathbb{E}[\mathbf{X}_0|\mathbf{X}_t]$ naturally belongs to its domain. Therefore, $\boldsymbol{\varepsilon}^*(\mathbf{X}_t, t)$ is feasible to the problem (14). Since the optimal solution of the problem is $\tilde{\varepsilon}_\theta(\mathbf{X}_t, t)$, we have

$$\|\tilde{\boldsymbol{\varepsilon}}_\theta(\mathbf{X}_t, t) - \boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, t)\| \le \|\boldsymbol{\varepsilon}^*(\mathbf{X}_t, t) - \boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, t)\|$$

for all $\mathbf{X}_t$ and t. This further leads to the result that

$$\mathbb{E}_{\mathbf{X}\sim p_t}[\|\tilde{f}(\mathbf{X}, t) - \hat{f}(\mathbf{X}, t)\|] \le \frac{\beta(t)}{\sqrt{1-e^{-\int_{t_0}^t \beta(s)ds}}}\delta. \tag{15}$$

Moreover, since $\tilde{\varepsilon}_\theta(\mathbf{X}_t, t)$ is essentially the projection of $\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, t)$ onto the convex set defined by the constraints in (14), and $\boldsymbol{\varepsilon}^*(\mathbf{X}_t, t)$ also belongs to the set, we know that the inner product of $\boldsymbol{\varepsilon}^*(\mathbf{X}_t, t) - \tilde{\boldsymbol{\varepsilon}}_\theta(\mathbf{X}_t, t)$ and $\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, t) - \tilde{\boldsymbol{\varepsilon}}_\theta(\mathbf{X}_t, t)$ is negative, which further leads to the result that

$$\|\tilde{\boldsymbol{\varepsilon}}_\theta(\mathbf{X}_t, t) - \boldsymbol{\varepsilon}^*(\mathbf{X}_t, t)\| \le \|\boldsymbol{\varepsilon}^*(\mathbf{X}_t, t) - \boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, t)\|, \tag{16}$$

which further implies

$$\mathbb{E}_{\mathbf{X}\sim p_t}[\|\tilde{f}(\mathbf{X},t)-f(\mathbf{X},t)\|] \leq \frac{\beta(t)}{\sqrt{1-e^{-\int_{t_0}^t \beta(s)ds}}}\delta. \tag{17}$$

The following Girsanov's Theorem (Karatzas & Shreve, 1991) will be used (together with equation 15 and equation 17) to prove the upper bounds for the KL-divergences in our Proposition 1:

Proposition 3. *Let p_0 be any probability distribution, and let $Z=(Z_t)_{t\in[0,T]}$, $Z'=(Z'_t)_{t\in[0,T]}$ be two different processes satisfying*

$$\begin{aligned} dZ_t &= b(Z_t,t)dt+\sigma(t)dB_t, \quad Z_0\sim p_0,\\ dZ'_t &= b'(Z'_t,t)dt+\sigma(t)dB_t, \quad Z'_0\sim p_0. \end{aligned}$$

We define the distributions of Z_t and Z'_t as p_t and p'_t, and the path measures of Z and Z' as $\mathbb{P}$ and $\mathbb{P}'$ respectively.

Suppose the following Novikov's condition:

$$\mathbb{E}_{\mathbb{P}}\left[\exp\left(\int_0^T \frac{1}{2}\int_x \sigma^{-2}(t)\|(b-b')(x,t)\|^2 dxdt\right)\right] < \infty. \tag{18}$$

Then, the Radon-Nikodym derivative of $\mathbb{P}$ with respect to $\mathbb{P}'$ is

$$\frac{d\mathbb{P}}{d\mathbb{P}'}(Z)=\exp\left\{-\frac{1}{2}\int_0^T \sigma(t)^{-2}\|(b-b')(Z_t,t)\|^2dt-\int_0^T \sigma(t)^{-1}(b-b')(Z_t,t)dB_t\right\},$$

and therefore we have that

$$KL(p_T\|p'_T)\leq KL(\mathbb{P}\|\mathbb{P}')=\int_0^T\frac{1}{2}\int_x p_t(x)\sigma(t)^{-2}\|(b-b')(x,t)\|^2dxdt.$$

Moreover, Chen et al. (2023) showed that if $\int_x p_t(x)\sigma^{-2}(t)\|(b-b')(x,t)\|^2dx\leq C$ holds for some constant C over all t, we have that

$$KL(p_T\|p'_T)\leq \int_0^T\frac{1}{2}\int_x p_t(x)\sigma(t)^{-2}\|(b-b')(x,t)\|^2dxdt,$$

even if the Novikov's condition equation 18 is not satisfied.

.

According to equation 15 and equation 17, we have

$$\int_x p_t(x)\beta(t)^{-1}\|\tilde{f}(\mathbf{X},t)-\hat{f}(\mathbf{X},t)\|dx \leq \frac{\beta(t)}{\sqrt{1-e^{-\int_{t_0}^t \beta(s)ds}}}\delta \leq \sup_{t\in[t_0,T]}\frac{\beta(t)}{\sqrt{1-e^{-\int_{t_0}^t \beta(s)ds}}}\delta, \tag{19}$$

$$\int_x p_t(x)\beta(t)^{-1}\|\tilde{f}(\mathbf{X},t)-f(\mathbf{X},t)\|dx \leq \frac{\beta(t)}{\sqrt{1-e^{-\int_{t_0}^t \beta(s)ds}}}\delta \leq \sup_{t\in[t_0,T]}\frac{\beta(t)}{\sqrt{1-e^{-\int_{t_0}^t \beta(s)ds}}}\delta. \tag{20}$$

Therefore, we can apply Proposition 3 to obtain upper bounds for the KL-divergences, which leads to

$$\begin{aligned} \mathrm{KL}(\tilde{p}_{t_0}|\hat{p}_{t_0}) &\leq \int_{t_0}^T\frac{1}{2}\int_x p_t(x)\beta(t)^{-1}\|\tilde{f}(\mathbf{X},t)-\hat{f}(\mathbf{X},t)\|dx\\ &\leq \delta\int_{t_0}^T\frac{1}{2}\frac{\beta(t)}{\sqrt{1-e^{-\int_{t_0}^t \beta(s)ds}}}dt \end{aligned} \tag{21}$$

and

$$\begin{aligned}\mathrm{KL}(\tilde{p}_{t_0}|p_{t_0}) &\leq \int_{t_0}^{T} \frac{1}{2} \int_x p_t(x)\beta(t)^{-1} \|\tilde{f}(\mathbf{X},t) - f(\mathbf{X},t)\| dx \\ &\leq \delta \int_{t_0}^{T} \frac{1}{2} \frac{\beta(t)}{\sqrt{1 - e^{-\int_{t_0}^{t} \beta(s) ds}}} dt.\end{aligned} \tag{22}$$

□

Remark 1. *Under the SDE formulation, the forward process terminates at a sufficiently large time T. Also, since the score functions blow up at $t \approx 0$, an early-stopping time t_0 is commonly adopted to avoid such issue (Song & Ermon, 2020; Nichol & Dhariwal, 2021). When t_0 is sufficiently small, the distribution of $\mathbf{X}_{t_0}$ in the forward process is close enough to the real data distribution.*

A.3. Proof of proposition 2

We first provide the following definition 1, which is adopted from Fu et al. (2024).

Definition 1. *Denote the space of density functions*

$$\mathcal{P}_0 = \left\{ p(\boldsymbol{X}) = f(\boldsymbol{X}) \exp(-C\|\boldsymbol{X}\|_2^2) : f \in \mathcal{L}(\mathbb{R}^{L\times H}, B), f(\boldsymbol{X}) \geq \alpha > 0 \right\},$$

where C and α can be any given constants, and $\mathcal{L}(\mathbb{R}^{L\times H}, B)$ denotes the class of Lipschitz continuous functions on $\mathbb{R}^{L\times H}$ with Lipschitz constant bounded by B.

Suppose that the density function of $\mathbf{X}$ belongs to the following space

$$\mathcal{P}_\delta = \left\{ p(\boldsymbol{X}) \in \mathcal{P}_0 | P(\bar{R}, O) = \delta \right\}, \tag{23}$$

where the distribution of $\mathbf{M}$ is defined from $\mathbf{X}$ by

$$\mathbf{M}_{lh} = \mathbf{1}\{\mathrm{X}_{lh} \geq 1/2\}.$$

Proposition 4. *Consider generating piano roll $\mathbf{M}$ from a continuous random variable $\mathbf{X}$, i.e., given n i.i.d. data $\{\mathbf{X}^i\}_{i=1}^n \sim p_{\mathbf{X}}$, let $\{\mathbf{M}^i\}_{i=1}^n$ be given by $\mathbf{M}_{lh}^i = \mathbf{1}\{\mathrm{X}_{lh}^i \geq 1/2\}$. Denote the model for estimating the distribution of $\mathbf{X}$ as $\widehat{p}_{\mathbf{X}}$. We have $\exists\, C > 0$ such that $\forall n$,*

$$\inf_{\widehat{p}_{\mathbf{X}}} \sup_{p_{\mathbf{X}} \in \mathcal{P}_\delta} \mathbb{E}_{\{\mathbf{M}^i\}_{i=1}^n} \widehat{P}(\bar{R}, O) \geq C \cdot n^{-\frac{1}{LH+2}} - P(\bar{R}, O), \tag{24}$$

where $\widehat{P}$ is the probability associated with the generated data $\widehat{p}_{\mathbf{X}}$.

Proof. We first restate a special case of proposition 4.3 of Fu et al. (2024) as the following lemma.

Lemma 1. *(Fu et al. (2024), proposition 4.3) Fix a constant $C_2 > 0$. Consider estimating a distribution $P(\mathbf{x})$ with a density function belonging to the space*

$$\mathcal{P} = \left\{ p(\mathbf{x}) = f(\mathbf{x}) \exp(-C_2\|\mathbf{x}\|_2^2) : f(\mathbf{x}) \in \mathcal{L}(\mathbb{R}^d, B), f(\mathbf{x}) \geq C > 0 \right\}.$$

Given n i.i.d. data $\{x_i\}_{i=1}^n$, we have

$$\inf_{\hat{\mu}} \sup_{p\in\mathcal{P}} \mathbb{E}_{\{x_i\}_{i=1}^n} [TV(\hat{\mu}, P)] \gtrsim n^{-\frac{1}{d+2}},$$

where the infimum is taken over all possible estimators $\hat{\mu}$ based on the data.

From lemma 1, since the space $\mathcal{P}_0$ that we define satisfies all the same conditions as the space $\mathcal{P}$ in lemma 1, we know from the conclusion of lemma 1 that

$$\inf_{\widehat{p}_{\mathbf{X}}} \sup_{p_{\mathbf{X}} \in \mathcal{P}_0} \mathbb{E}_{\{x_i\}_{i=1}^n} [\mathrm{TV}(\widehat{p}_{\mathbf{X}}, p_{\mathbf{X}})] \gtrsim n^{-\frac{1}{LH+2}}, \tag{25}$$

where by definition of total variation,

$$\mathrm{TV}(\widehat{p}_{\mathbf{X}}, p_{\mathbf{X}}) = \int_{\mathbb{R}^{L\times H}} |\widehat{p}_{\mathbf{X}}(\boldsymbol{X}) - p_{\mathbf{X}}(\boldsymbol{X})| d\boldsymbol{X}. \tag{26}$$

For simplicity, suppose event O denotes a note-out-of-key occurring at $(l, h) = (1, 1)$. We have

$$\begin{aligned} \widehat{P}(O) &= \int_{(\frac{1}{2},+\infty)} dX_{11} \int_{\mathbb{R}^{L\times H-1}} d\boldsymbol{Y}\ \widehat{p}_{\mathbf{X}}(X_{11}, \boldsymbol{Y}) \\ &\triangleq \int_{\Omega_O} \widehat{p}_{\mathbf{X}}(\boldsymbol{X}) d\boldsymbol{X}, \end{aligned} \tag{27}$$

where $\boldsymbol{Y}$ is a $(LH-1)$-dimensional variable denoting the elements in matrix $\boldsymbol{X}$ excluding X_{11}. Let $\mathbb{C}(O)$ denote the set of all possible realizations of piano roll $\boldsymbol{M}$ that contains (i) the note O as an out-of-key note, and (ii) a "resolution"[16] to accommodate it. For each $\boldsymbol{M} \in \mathbb{C}(O)$, let

$$\delta(\boldsymbol{M}) = \{(l, h) \in [\![1, L]\!] \times [\![1, H]\!] | M_{lh} = 1\}.$$

Therefore, we have

$$\begin{aligned} \widehat{P}(R, O) &= \sum_{\boldsymbol{M}\in\mathbb{C}(O)} \int_{(\frac{1}{2},+\infty)^{|\delta(\boldsymbol{M})|}} dX_{\delta(\boldsymbol{M})} \int_{(-\infty,\frac{1}{2})^{L\times H-|\delta(\boldsymbol{M})|}} d\boldsymbol{Y}\ \widehat{p}_{\mathbf{X}}(X_{\delta(\boldsymbol{M})}, X_{L\times H\setminus\delta(\boldsymbol{M})}) \\ &\triangleq \int_{\Omega_{\mathbb{C}(O)}} \widehat{p}_{\mathbf{X}}(\boldsymbol{X}) d\boldsymbol{X}, \end{aligned} \tag{28}$$

and note that $\Omega_{\mathbb{C}(O)} \subset \Omega_O$, we have

$$\widehat{P}(\bar{R}, O) = \widehat{P}(O) - \widehat{P}(R, O) = \int_{\Omega_O\setminus\Omega_{\mathbb{C}(O)}} \widehat{p}_{\mathbf{X}}(\boldsymbol{X}) d\boldsymbol{X} \tag{29}$$

To better explain and summarize equation 27, equation 28 and equation 29, the probabilities $\widehat{P}(\cdot)$ (the estimated probabilities of $O, \{R, O\}$ or $\{\bar{R}, O\}$) are always calculated from integrating $\widehat{p}_{\mathbf{X}}(\boldsymbol{X})$ on a corresponding domain, and the key of the 3 equations are all about finding the domain on which to integrate . Similarly, for the ground truth distributions and under definition 1 which provides $P_{\boldsymbol{M}}(\bar{R}, O) = \delta$, we have

$$P(\bar{R}, O) = \int_{\Omega_O\setminus\Omega_{\mathbb{C}(O)}} p_{\mathbf{X}}(\boldsymbol{X}) d\boldsymbol{X} \le \delta.$$

Therefore,

$$\begin{aligned} \widehat{P}(\bar{R}, O) &= \int_{\Omega_O\setminus\Omega_{\mathbb{C}(O)}} \widehat{p}_{\mathbf{X}}(\boldsymbol{X}) d\boldsymbol{X} \\ &\ge \int_{\Omega_O\setminus\Omega_{\mathbb{C}(O)}} |\widehat{p}_{\mathbf{X}}(\boldsymbol{X}) - p_{\mathbf{X}}(\boldsymbol{X})| - p_{\mathbf{X}}(\boldsymbol{X}) d\boldsymbol{X} \\ &\ge \int_{\Omega_O\setminus\Omega_{\mathbb{C}(O)}} |\widehat{p}_{\mathbf{X}}(\boldsymbol{X}) - p_{\mathbf{X}}(\boldsymbol{X})|\, d\boldsymbol{X} - \delta \end{aligned} \tag{30}$$

Therefore,

$$\widehat{P}(\bar{R}, O) = \mathrm{TV}|_{\Omega_O\setminus\Omega_{\mathbb{C}(O)}}(\widehat{p}_{\mathbf{X}}, p_{\mathbf{X}}) - \delta, \tag{31}$$

where $\mathrm{TV}|_{\Omega_O\setminus\Omega_{\mathbb{C}(O)}}$ is the total variation integral restricted on the domain $\Omega_O\setminus\Omega_{\mathbb{C}(O)}$.

[16] By definition, the resolution of an out-of-key note refers to how it is integrated into the surrounding harmonic and melodic structure to make it sound intentional rather than an error.

By construction of packing numbers provided in the proof of proposition 4.3 of Fu et al. (2024), we note that constraint $P_{\boldsymbol{M}}(\bar{R}, O) = \delta$ or restricting the integral of total variation on $\Omega_O \backslash \Omega_{\mathbb{C}(O)}$ does not change the order of the packing numbers, i.e., $\mathcal{P}_0$ and $\mathcal{P}_\delta$ have the same packing numbers. Let

$$\mathcal{P}_\delta^{\Omega_O \backslash \Omega_{\mathbb{C}(O)}} = \left\{ C(\Omega_O \backslash \Omega_{\mathbb{C}(O)}) \cdot p(\mathbf{X}) \mathbf{1}_{\mathbf{X} \in \Omega_O \backslash \Omega_{\mathbb{C}(O)}} \mid p(\mathbf{X}) \in \mathcal{P}_\delta \right\},$$

where the constant $C(\Omega_O \backslash \Omega_{\mathbb{C}(O)})$ is a scale factor to ensure that $C(\Omega_O \backslash \Omega_{\mathbb{C}(O)}) \cdot p(\mathbf{X}) \mathbf{1}_{\mathbf{X} \in \Omega_O \backslash \Omega_{\mathbb{C}(O)}}$ is a probability density function. For simplicity we use $\mathcal{P}(\delta, O)$ for short of $\mathcal{P}_\delta^{\Omega_O \backslash \Omega_{\mathbb{C}(O)}}$. Therefore, from the original lemma 1 of Fu et al. (2024) we have equation 25. Only changing the $\mathcal{P}_0$ into $\mathcal{P}(\delta, O)$ (all the arguments above are to justify why this change can be made), we have

$$\inf_{\widehat{p}_{\mathbf{X}}} \sup_{p \in \mathcal{P}(\delta, O)} \mathbb{E}_{\{\mathbf{X}_i\}_{i=1}^n} \mathrm{TV}(\widehat{p}_{\mathbf{X}}, p_{\mathbf{X}}) \gtrsim n^{-\frac{1}{LH+2}}. \tag{32}$$

Combining equation 32 with equation 31, and starting from our target $\mathbb{E}\widehat{P}(\bar{R}, O)$, we have

$$\begin{aligned} &\inf_{\widehat{p}_{\mathbf{X}}} \sup_{p \in \mathcal{P}_\delta} \mathbb{E}_{\{\mathbf{X}_i\}_{i=1}^n} \widehat{P}(\bar{R}, O) + \delta = \inf_{\widehat{p}_{\mathbf{X}}} \sup_{p \in \mathcal{P}_\delta} \mathrm{TV}|_{\Omega_O \backslash \Omega_{\mathbb{C}(O)}}(\widehat{p}_{\mathbf{X}}, p_{\mathbf{X}}) + \delta \\ =& \inf_{\widehat{p}_{\mathbf{X}}} \sup_{p \in \mathcal{P}(\delta, O)} \mathrm{TV}(\widehat{p}_{\mathbf{X}}, p_{\mathbf{X}}) + \delta \gtrsim n^{-\frac{1}{LH+2}}. \end{aligned}$$

Therefore, $\exists C > 0, \forall n$,

$$\inf_{\widehat{p}_{\mathbf{X}}} \sup_{p \in \mathcal{P}_\delta} \mathbb{E}_{\{\mathbf{X}_i\}_{i=1}^n} \widehat{P}(\bar{R}, O) \geq C \cdot n^{-\frac{1}{LH+2}} - P(\bar{R}, O).$$

which finishes the proof. □

B. Details of Conditioning and Algorithms

B.1. Mathematical formulation of textural conditions in section 3.1

Denote a chord progression by $\mathcal{C}$, where $\mathcal{C}(l)$ denotes the chord at time $l \in [\![1, L]\!]$. Let $\gamma_{\mathcal{C}}(l) \subset [\![1, H]\!]$ denote the set of pitch index h that belongs to the pitch classes included in chord $\mathcal{C}(l)$.[17], and let $\gamma_{\mathcal{R}} \subset [\![1, L]\!]$ denote the set of onset time indexes corresponding to rhythmic pattern $\mathcal{R}$. We define the following versions of representations for the condition:

- When harmonic ($\mathcal{C}$) and rhythmic ($\mathcal{R}$) conditions are both provided, the corresponding conditional piano roll $\boldsymbol{M}^{\text{cond}}(\mathcal{C}, \mathcal{R})$ is given element-wise by $M^{\text{cond}}{}_{lh}(\mathcal{C}, \mathcal{R}) = \mathbf{1}\{l \in \gamma_{\mathcal{R}}\}\mathbf{1}\{h \in \gamma_{\mathcal{C}}(l)\}$, meaning that the (l, h)-element is 1 if pitch index h belongs to chord $\mathcal{C}(l)$ and there is onset notes at time l, and 0 otherwise.
- When only harmonic ($\mathcal{C}$) condition is provided, the corresponding piano roll $\boldsymbol{M}^{\text{cond}}(\mathcal{C})$ is given element-wise by $M^{\text{cond}}{}_{lh}(\mathcal{C}) = -1 - \mathbf{1}\{h \in \gamma_{\mathcal{C}}(l)\}$, meaning that the (l, h)-element is -2 if pitch index h belongs to chord $\mathcal{C}(l)$, and -1 otherwise.

Figure 1 and Figure 2 provides illustrative examples of $\boldsymbol{M}^{\text{cond}}(\mathcal{C}, \mathcal{R})$ and $\boldsymbol{M}^{\text{cond}}(\mathcal{C})$. The use of -2 and -1 (rather than 1 and 0) in the latter case ensures that the model can fully capture the distinctions between the two scenarios, as a unified model will be trained on both types of conditions.

B.2. Classifier Free Guidance

To enable the model to generate under varying levels of conditioning, including unconditional generation, we implement the idea of classifier-free guidance, and randomly apply conditions with or without rhythmic pattern in the process of training. Namely, the training loss is modified from equation 1 and given as

$$\begin{aligned} &\mathbb{E}_{t, \boldsymbol{\varepsilon}, \mathbf{X}_0} \left[\lambda_1(t) \| \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, \mathbf{M}^{\text{cond}}(\mathcal{C}), t) \|^2 \right. \\ &\left. + \lambda_2(t) \| \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, \mathbf{M}^{\text{cond}}(\mathcal{C}, \mathcal{R}), t) \|^2 \right], \end{aligned} \tag{33}$$

[17] For example, when $\mathcal{C}(l) =$ C major (consisting of pitch classes C, E and G), $\gamma_{\mathcal{C}}$ includes all pitch values corresponding to the three pitch classes across all octaves.

where $\lambda_1(t)$ and $\lambda_2(t)$ are hyper-parameters. Note that both $\mathbf{M}^{\text{cond}}(\mathcal{C})$ and $\mathbf{M}^{\text{cond}}(\mathcal{C},\mathcal{R})$ are derived from $\mathbf{X}_0$ via pre-designed chord recognition and rhythmic identification algorithms.

The guided noise prediction at timestep t is then computed as

$$\begin{aligned}\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t,t|\mathcal{C},\mathcal{R}) =&\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t,\boldsymbol{M}^{\text{cond}}(\mathcal{C}),t)\\ &+w\cdot\left[\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t,\boldsymbol{M}^{\text{cond}}(\mathcal{C},\mathcal{R}),t)\right.\\ &\left.-\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t,\boldsymbol{M}^{\text{cond}}(\mathcal{C}),t)\right],\end{aligned}\tag{34}$$

where w is the weight parameter. Note that the general formulation $\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t,t|\mathcal{C},\mathcal{R})$ includes the case where rhythmic guidance is not provided ($\mathcal{R}=\emptyset$), and w in equation 34 is set as 0.

B.3. Additional algorithms in section 3.2

In this section, we provide the following algorithm: fine-grained sampling guidance additionally with rhythmic regularization, fine-grained sampling guidance combined with DDIM sampling.

Let $\mathcal{B}$ denote the rhythmic regularization. Specifically, we have the following types of regularization:

- $\mathcal{B}_1$: Requiring exactly N onset of a note at time position l, i.e., $\sum_{h\in[\![1,H]\!]}M_{lh}=N$
- $\mathcal{B}_2$: Requiring at least N onsets at time position l, i.e.,

  $$\exists\boldsymbol{h}\subset[\![1,H]\!],\ \text{or}\ \exists\boldsymbol{h}\subset[\![1,H]\!]\backslash\omega_{\mathcal{K}}(l)\ \text{if harmonic regularization is jointly included}$$

  such that $M_{l\boldsymbol{h}}=1$, and $|\boldsymbol{h}|\geq N$
- $\mathcal{B}_3$: Requiring no onset of notes at time position l, i.e., $\forall h\in[\![1,H]\!]$, $M_{lh}=0$

Let the set of $\boldsymbol{M}$ satisfying a specific regularization $\mathcal{B}$ be denoted as $\mathbb{M}_{\mathcal{B}}$, and the corresponding set of $\boldsymbol{X}$ be denoted as $\tilde{\mathbb{M}}_{\mathcal{B}}$, note that this includes the case where multiple requirements are satisfied, resulting in

$$\tilde{\mathbb{M}}_{\mathcal{B}}=\tilde{\mathbb{M}}_{\mathcal{B}_1,\mathcal{B}_2,\ldots}=\tilde{\mathbb{M}}_{\mathcal{B}_1}\cap\tilde{\mathbb{M}}_{\mathcal{B}_2}\cap\ldots.$$

The correction of predicted noise score is then formulated as

$$\begin{aligned}\tilde{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t,t|\mathcal{C},\mathcal{R})=\arg\min_{\boldsymbol{\varepsilon}}\quad&\|\boldsymbol{\varepsilon}-\widehat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t,t|\mathcal{C},\mathcal{R})\|\\ \text{s.t.}\quad&\left(\frac{\boldsymbol{X}_t-\sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}}{\sqrt{\bar{\alpha}_t}}\right)\in\tilde{\mathbb{M}}_{\mathcal{B}}.\end{aligned}\tag{35}$$

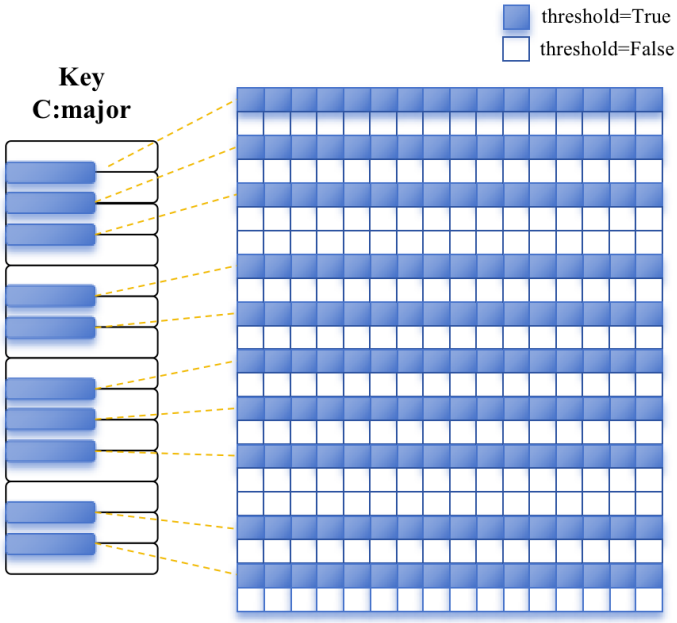
Further, we can perform predicted noise score correction with joint regularization on rhythm and harmony, resulting in the corrected noise score

$$\begin{aligned}\tilde{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t,t|\mathcal{C},\mathcal{R})=\arg\min_{\boldsymbol{\varepsilon}}\quad&\|\boldsymbol{\varepsilon}-\widehat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t,t|\mathcal{C},\mathcal{R})\|\\ \text{s.t.}\quad&\left(\frac{\boldsymbol{X}_t-\sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}}{\sqrt{\bar{\alpha}_t}}\right)\in(\mathbb{R}^{L\times H}\backslash\mathbb{W}'_{\mathcal{K}})\cap\tilde{\mathbb{M}}_{\mathcal{B}}.\end{aligned}\tag{36}$$

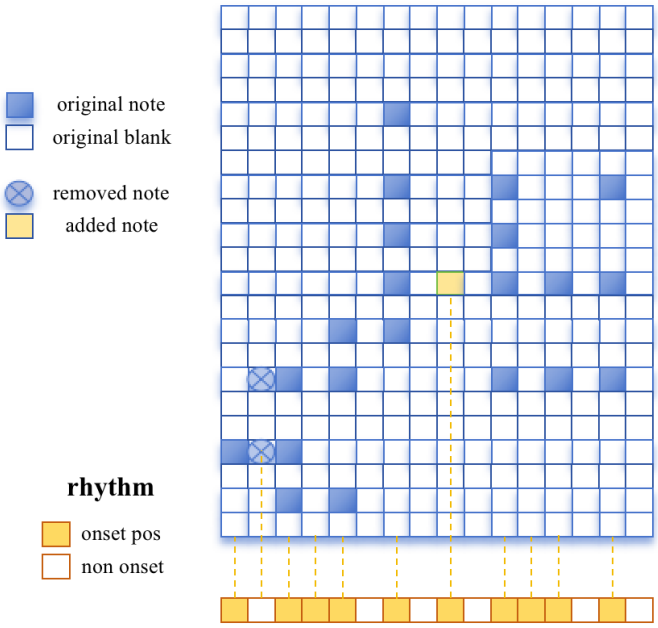
We for example provide a element-wise solution of $\tilde{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t,t|\mathcal{C},\mathcal{R})$ defined by problem (35). For given l, suppose $\mathcal{B}(l)$ takes the form of $\mathcal{B}_2$, for simplicity take $N=1$. This gives $\tilde{\boldsymbol{\varepsilon}}_{\theta,lh}=\widehat{\boldsymbol{\varepsilon}}_{\theta,lh}$ if $\max_h\mathbb{E}[\mathbf{X}_0|\boldsymbol{X}_t]_{hl}\geq\frac{1}{2}$ and $\mathbb{E}[\mathbf{X}_0|\boldsymbol{X}_t]_{hl}=\frac{1}{2}$, $h=\arg\max_h\mathbb{E}[\mathbf{X}_0|\boldsymbol{X}_t]_{hl}$, i.e.,

$$\tilde{\boldsymbol{\varepsilon}}_{\theta,lh}=\frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(X_{t,lh}-\frac{\sqrt{\bar{\alpha}_t}}{2}\right),$$

if $\max_h\mathbb{E}[\mathbf{X}_0|\boldsymbol{X}_t]_{hl}<\frac{1}{2}$. The correction applied to predicted $\mathbf{X}_0$ ($\mathbb{E}[\mathbf{X}_0|\boldsymbol{X}_t]$) is illustrated in the following figure 4.



(a) Fine-grained control for $\mathbb{E}[\mathbf{X}_0|\boldsymbol{X}_t] \in \mathbb{R}^{L\times H}\backslash\mathbb{W}'_{\mathcal{K}}$. The colored spots denote places that we require $\mathbb{E}[\mathbf{X}_0|\boldsymbol{X}_t]_{lh} \leq \frac{1}{2}$.



(b) Fine-grained control for $\mathbb{E}[\mathbf{X}_0|\boldsymbol{X}_t] \in \mathbb{W}'_{\mathcal{B}}$. Original notes are removed at l if $\mathcal{B}_3$ is applied. Otherwise if $\mathcal{B}_1$ is applied and currently no note exists, the "most likely notes" (i.e., at $h = \arg\max \mathbb{E}[\mathbf{X}_0|\boldsymbol{X}_t]_{lh}$) are added.

Figure 4. Illustration of fine-grained control on predicted $\mathbf{X}_0$.

Algorithm 2 DDPM sampling with fine-grained textural guidance

Input: Input parameters: forward process variances β_t, $\bar{\alpha}_t = \prod_{s=1}^{t} \beta_t$, backward noise scale σ_t, chord condition $\mathcal{C}$, key signature $\mathcal{K}$, rhythmic condition $\mathcal{R}$, rhythmic guidance $\mathcal{B}$
Output: generated piano roll $\tilde{\mathbf{M}} \in \{0,1\}^{L\times H}$
$\mathbf{X}_T \sim \mathcal{N}(0, \boldsymbol{I})$;
for $t = T, T-1, \ldots, 1$ **do**
  Compute guided noise prediction $\widehat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t|\mathcal{C}, \mathcal{R})$
  Perform noise correction: derive $\tilde{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t|\mathcal{C}, \mathcal{R})$ optimization equation 36
  Compute $\tilde{\mathbf{X}}_{t-1}$ by plugging the corrected noise $\tilde{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_t, t|\mathcal{C}, \mathcal{R})$ into equation 2
end
Convert $\tilde{\mathbf{X}}_0$ into piano roll $\tilde{\mathbf{M}}$
return *output*

We additionally remark that the fine-grained sampling guidance is empirically effective with the DDIM sampling scheme, which drastically improves the generation speed. Specifically, select subset $\{\tau_i\}_{i=1}^m \subset [\![1, T]\!]$, and denote

$$\mathbf{X}_{\tau_{i-1}} = \sqrt{\bar{\alpha}_{\tau_{i-1}}} \left(\frac{\mathbf{X}_t - \sqrt{1-\bar{\alpha}_{\tau_i}} \widehat{\boldsymbol{\varepsilon}}_\theta(\mathbf{X}_{\tau_i}, \tau_i)}{\sqrt{\bar{\alpha}_{\tau_i}}} \right) + \sqrt{1 - \bar{\alpha}_{\tau_{i-1}} - \sigma_{\tau_i}^2} \widehat{\boldsymbol{\varepsilon}}_\theta(\mathbf{X}_{\tau_i}, \tau_i) + \sigma_{\tau_i} \boldsymbol{\varepsilon}_{\tau_i},$$

we similarly perform the DDIM noise correction

$$\begin{aligned} \tilde{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_{\tau_i}, \tau_i|\mathcal{C}, \mathcal{R}) = \arg\min_{\boldsymbol{\varepsilon}} \quad & \|\boldsymbol{\varepsilon} - \widehat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{X}_{\tau_i}, \tau_i|\mathcal{C}, \mathcal{R})\| \\ \text{s.t.} \quad & \left(\frac{\boldsymbol{X}_t - \sqrt{1-\bar{\alpha}_{\tau_i}}\boldsymbol{\varepsilon}}{\sqrt{\bar{\alpha}_{\tau_i}}} \right) \in (\mathbb{R}^{L\times H} \backslash \mathbb{W}'_{\mathcal{K}}) \cap \tilde{\mathbb{M}}_{\mathcal{B}}. \end{aligned}$$

on each step i.

C. Comparison with Related Works

We provide a detailed comparison between our method and two related works in controlled diffusion models with constrained or guided intermediate sampling steps:

Comparison with reflected diffusion models In Lou & Ermon (2023), a bounded setting is used for both the forward and backward processes, ensuring that the bound applies to the training objective as well as the entire sampling process. In contrast, we do not adopt the framework of bounded Brownian motion, because we do not require the entire sampling process to be bounded within a given domain; instead, we only enforce that the final sample outcome aligns with the constraint. While Lou & Ermon (2023) enforces thresholding on $\mathbf{X}_t$ in both forward and backward processes, our approach is to perform a thresholding-like projection method on the predicted noise $\boldsymbol{\varepsilon}_\theta(\mathbf{X}_t, t)$, interpreted as noise correction.

Comparison with non-differentiable rule guided diffusion Huang et al. (2024) guides the output with musical rules by sampling multiple times at intermediate steps, and continuing with the sample that best fits the musical rule, producing high-quality, rule-guided music. Our work centers on a different aspect, prioritizing precise control to tackle the challenges of accuracy and regularization in symbolic music generation. Also, we place additional emphasis on sampling speed, ensuring stable generation of samples within seconds to facilitate interactive music creation and improvisation.

D. Numerical Experiment Details

D.1. Detailed Data Representation

The two-channel version of piano roll with with both harmonic and rhythm conditions ($\mathbf{M}^{\text{cond}}(\mathcal{C}, \mathcal{R})$) and with harmonic condition ($\mathbf{M}^{\text{cond}}(\mathcal{C})$) with onset and sustain are represented as:

- $\mathbf{M}^{\text{cond}}(\mathcal{C}, \mathcal{R})$: In the first channel, the (l, h)-element is 1 if there are onset notes at time l and pitch index h belongs to the chord $\mathcal{C}(l)$, and 0 otherwise. In the second channel, the (l, h)-element is 1 if pitch index h belongs to the chord $\mathcal{C}(l)$ and there is no onset note at time l.

- $\mathbf{M}^{\text{cond}}(\mathcal{C})$: In both channels, the (l, h)-element is 1 if pitch index h belongs to the chord $\mathcal{C}(l)$, and 0 otherwise.

In each diffusion step t, the model input is a concatenated 4-channel piano roll with shape $4 \times L \times 128$, where the first two channels correspond to the noisy target $\boldsymbol{X}_t$ and the last two channels correspond to the condition $\boldsymbol{M}^{\text{cond}}$ (either $\mathbf{M}^{\text{cond}}(\mathcal{C}, \mathcal{R})$ or $\mathbf{M}^{\text{cond}}(\mathcal{C})$). The output is the noise prediction $\hat{\varepsilon_\theta}$, which is a 2-channel piano roll with the same shape as $\boldsymbol{X}_t$. For the accompaniment generation experiments, we provide melody as an additional condition, which is also represented by a 2-channel piano roll with shape $2 \times L \times 128$, with the same resolution and length as $\boldsymbol{X}$. The melody condition is also concatenated with $\boldsymbol{X}_t$ and $\boldsymbol{M}^{\text{cond}}$ as model input, which results in a full 6-channel matrix with shape $6 \times L \times 128$.

D.2. Training and Sampling Details

We set diffusion timesteps $T = 1000$ with $\beta_0 = 8.5e{-}4$ and $\beta_T = 1.2e{-}2$. We use AdamW optimizer with a learning rate of $5e{-}5$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We applied data augmentation by transposing each 4-measure piece into all 12 keys. This involves uniformly shifting the pitch of all notes and adjusting the corresponding chords accordingly. This augmentation expands the dataset to 189,132 samples. Training is conducted with a batch size of 16, utilizing random sampling without replacement. Specifically, in each iteration, 16 samples are randomly selected without replacement until all samples are utilized, constituting one epoch. This procedure is repeated to ensure each sample was processed twice during training, resulting in a total of 23,642 iterations.

To speed up the sampling process, we select a sub-sequence of length 10 from $\{1, \cdots, T\}$ and apply the accelerated sampling process in Song et al. (2021a). It takes 0.4 seconds to generate the 4-measure accompaniment on a NVIDIA RTX 6000 Ada Generation GPU.

D.3. Experiments on Symbolic Music Generation Given only Chord Conditions

As mentioned in Section 5.1, we also run numerical experiments on symbolic music generation tasks given only chord condition. However, compared with the accompaniment generation task, we remark that this experiment does not have enough effective basis for comparison.

For the accompaniment generation task, we evaluate the cosine similarity of chord progression between the generated samples and the ground truth, as well as the IoU of chord and piano roll. The comparison with ground truth on those features make sense in the accompaniment generation task, because the leading melody inherently contains many constraints on the rhythm and pitch range of the accompaniment, ensuring coherence with the melody. Thus, similarity with ground truth on those metrics serves as an indicator of how well the generated samples adhere to the melody.

However, in symbolic music generation conditioned only on a chord sequence, while chord progression similarity remains comparable (as the chord sequence is provided), evaluating IoU of piano roll against ground truth is less informative. This is because multiple different pitch range and rhythm could appropriately align with a given chord progression, making deviations from the ground truth in these features less indicative of sample quality. Therefore, chord similarity emerges as the sole applicable metric in this context.

Additionally, WholeSongGen's architecture does not support music generation conditioned solely on chord progressions, as it utilizes a shared piano-roll for both chord and melody, rendering it unsuitable for comparison. Conversely, GETMusic facilitates the generation of both melody and piano accompaniment based on chord conditions, allowing for a viable comparison.

Consequently, we present results focusing on chord similarity between our model and GETMusic. For our model, we evaluate performance under two conditions: with both conditioning and control during training and sampling, and with conditioning during training but without control during sampling. The outcomes, summarized in Table 3, indicate that our fully controlled FGG method surpasses both the one without sampling control and GETMusic.

Methods	**FGG (Ours)**	FGG, only Training control	GETMusic
Chord Similarity	$\mathbf{0.676 \pm 0.007}$	0.645 ± 0.008	0.499 ± 0.013

Table 3. Evaluation of the similarity with ground truth, chord-conditioned music generation.

E. Demo Page Details

In this section, we briefly introduce how the Dorian mode and Chinese style clips are generated. We note that both styles are shaped not only by key-constraint $\mathcal{K}$, but also with designed chord progressions $\mathcal{C}$.

The key constraint for Dorian mode, example 1, is $\mathcal{K}_1 = \{A, B, C, D, E, F\#, G\}$ throughout the 4 bars, which means all generated notes have to be in the pitch classes in $\mathcal{K}_1$. The the chord progression for Dorian mode, example 1 is

$$\mathcal{C}_1 = \text{Am}(4) - \text{Em}(2) - \text{Am}(2) - \text{C}(2) - \text{D}(2) - \text{Am}(2) - \text{D}(2).$$

For example 2, $\mathcal{K}_2 = \{D, E, F, G, A, B, C\}$, and

$$\mathcal{C}_2 = \text{Dm}(4) - \text{G}(4) - \text{C}(4) - \text{F}(4).$$

The number in parentheses corresponds to the number of beats the chord lasts. For example, at the beginning of $\mathcal{C}_1$, the chord Am lasts 4 beats. Therefore, for the condition matrix under the 16th resolution, the positions corresponding to pitch classes A, C and E have value 1, where the rest have value 0, for $t = 0, 1, 2, \ldots, 15$. The condition is passed to the diffusion model as generation condition. Then $\mathcal{K}_1$ is applied as sampling control to shape and refine the tonal quality.

Similarly, for Chinese mode, we have $\mathcal{K}_1 = \{C, D, E, G, A\}$ and

$$\mathcal{C}_1 = \text{G}(2) - \text{Am}(2) - \text{C}(2) - \text{G}(2) - \text{Em}(2) - \text{G}(2) - \text{D}(4).$$

For the second example, $\mathcal{K}_2 = \{D, E, \#F, A, B\}$, and

$$\mathcal{C}_2 = \text{A}(4) - \text{Bm}(2)\text{Fm}(2) - \text{Bm}(2) - \text{A}(2) - \text{Fm}(2) - \text{A}(2).$$

F. Subjective Evaluation

To compare performance of our FGG method against the baselines (WholeSongGen and GETMusic), we prepared 6 sets of generated samples, with each set containing the melody paired with accompaniments generated by FGG, WholeSongGen, and GETMusic, along with the ground truth accompaniment. This yields a total of $6 \times 4 = 24$ samples. The samples are presented in a randomized order, and their sources are not disclosed to participants. Experienced listeners assess the quality of samples in 5 dimensions: creativity, harmony (whether the accompaniment is in harmony with the melody), melodiousness, naturalness and richness, together with an overall assessment.

F.1. Background of Participants

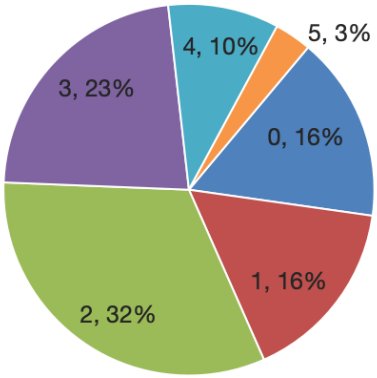
To evaluate the musical background of the participants, we first present the following questions:

- How many instruments (including vocal) are you playing or have you played?
- Please list all instruments (including vocal) that you are playing or have played.
- What is the instrument (including vocal) you have played the longest, and how many years have you been playing it? (e.g., piano, 3 years)

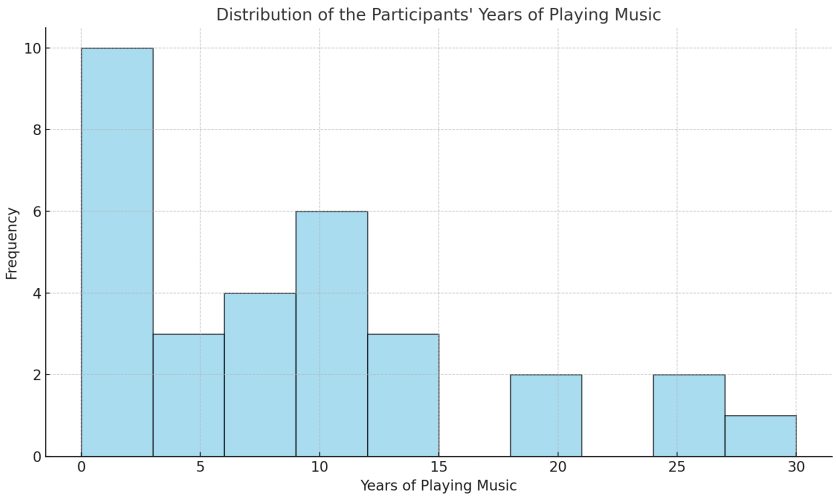
We recruited 31 participants with substantial musical experience for our survey. The number of instruments these participants play range from 0 to 5, with an average value of 2.03, and a standard deviation of 1.31. Examples of instrument played include piano, violin, vocal, guitar, saxphone, Dizi, Yangqin and Guzheng. The average years of playing has an average of 8.61 and standard deviation of 8.08. Specifically, the percentage of participants with ≥ 3 years of playing music is 67.74%, and the percentage of participants with ≥ 10 years of playing music is 45.16%. The distributions are given in the following figure 5.

F.2. Evaluation Questions

Thank you for taking the time to participate in this experiment. You will be presented with 6 sets of clips, each containing 4 clips. The first clip in each set features the melody alone, while the remaining three include the melody accompanied by different accompaniments. After listening to each clip, please evaluate the accompaniments in the following dimensions based on your own experience.



(a) Number of instruments played by the participants.



(b) Distribution of the participants' years of playing instruments.

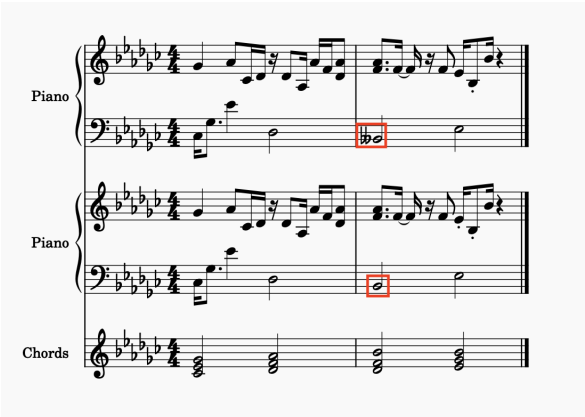
Figure 5. Information of the musical background of the participants in the subjective evaluation.

- Does the accompaniment sound pleasant to you?
- How would you rate the richness (i.e., the complexity, fullness, and expressive depth) of the accompaniment?
- Does the accompaniment sound natural to you?
- Do you think the accompaniment aligns well with the melody?
- Does the accompaniment sound creative to you?
- Please give an overall score for the clip.

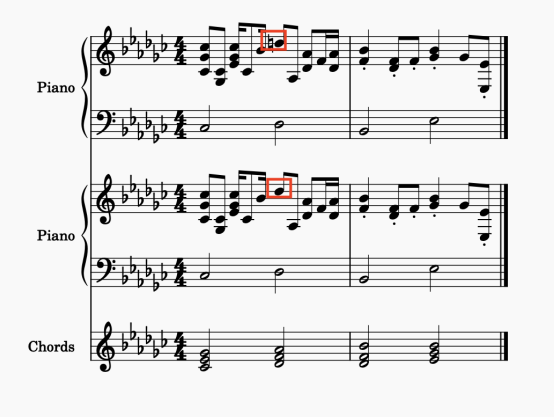
For each question, participants are provided with a Likert scale ranging from 1 to 5, where 1 represents "very poor" and 5 represents "very good."

G. Representative Examples of Sampling Control

In this section, we provide empirical examples of how model output is reshaped by fine-grained correction in Figure 6. Notably, harmonic control not only helps the model eliminate incorrect notes, but also guides it to replace them with correct ones.



(a) An example of replacing an out-of-key note B𝄫 with the in-key note B♭.



(b) An example of replacing an out-of-key note D♮ with the in-key note D♭.

Figure 6. Examples resulting from symbolic music generation with FGG. The first track is generated without key-signature control in sampling, the second track is generated with key-signature sampling control. The third track presents the chord condition. In each subfigure, the tracks are generated with the same conditions and the same set of noise.

H. The Effect of Guidance Weight for Classifier-free Guidance

In Section 3.1, we discussed the implementation of classifier-free guidance for rhythmic patterns, designed to enable the model to generate outputs under varying levels of conditioning. Specifically, we randomly apply conditions with or without rhythmic patter in the process of training. This approach ensures that the model can function effectively with both chord and rhythmic conditions or with chord conditions alone. Following Ho & Salimans (2021), when generating with both chord and rhythmic conditions, the guided noise prediction at timestep t is computed as:

$$\begin{aligned}\varepsilon_\theta(\mathbf{X}_t, t|\mathcal{C}, \mathcal{R}) =& \varepsilon_\theta(\mathbf{X}_t, \boldsymbol{M}^{\text{cond}}(\mathcal{C}), t) \\ &+ w \cdot \left[\varepsilon_\theta(\mathbf{X}_t, \boldsymbol{M}^{\text{cond}}(\mathcal{C}, \mathcal{R}), t) - \varepsilon_\theta(\mathbf{X}_t, \boldsymbol{M}^{\text{cond}}(\mathcal{C}), t)\right],\end{aligned}$$

where $\varepsilon_\theta(\mathbf{X}_t, \boldsymbol{M}^{\text{cond}}(\mathcal{C}, \mathcal{R}), t)$ is the model's predicted noise without rhythmic condition, and $\varepsilon_\theta(\mathbf{X}_t, \boldsymbol{M}^{\text{cond}}(\mathcal{C}, \mathcal{R}), t)$ is the model's predicted noise with rhythmic condition, and w is the guidance weight.

The literature has consistently demonstrated that the guidance weight w plays a pivotal role in balancing diversity and stability in generation tasks (Ho & Salimans, 2021; Chang et al., 2023; Gao et al., 2023; Lin et al., 2024). In general, a lower weight w enhances sample diversity and quality, but this may come at the cost of deviation from the provided conditions. Conversely, higher values of w promote closer adherence to the conditioning input, but excessively high w can degrade output quality by over-constraining the model, resulting in less natural or lower-quality samples.

In this section, we hope to investigate the effect of the guidance weight w on our music generation task. We focus on the same accompaniment generation task as mentioned in Section 5. To measure the samples' adherence to rhythmic controls, we use the rhythm of the ground truth as the rhythmic condition and assess the overlapping area (OA) of note duration and note density between the generated and ground-truth samples. Specifically, we split both the generated accompaniments and the ground truth into non-overlapping 2-measure segments. Following (von Rütte et al., 2023), for each feature f ($f \in \{\text{note duration}, \text{note density}\}$), we calculate the macro overlapping area (MOA) in segment-level feature distributions so that the metric also considers the temporal order of the features. MOA is defined as

$$MOA(f) = \frac{1}{N}\sum_{i=1}^{N} \text{overlap}(\pi_i^{\text{gen}}(f), \pi_i^{\text{gt}}(f)),$$

where $\pi_i^{\text{gen}}(f)$ is the distribution of feature f in the i-th generated segment, and $\pi_i^{\text{gt}}(f))$ is the distribution of feature f in the i-th ground truth segment. Additionally, we measured the percentage of out-of-key notes as a proxy for sample quality.

In these experiments, we only use the fine-grained control in training, but do not insert any sampling control so that we can

evaluate the inherent performance of the models themselves. The experiments were conducted across a range of guidance weights (w from 0.5 to 10), and he results are summarized in Table 4.

Values of w	% Out-of-Key Notes	OA (duration)	OA (note density)
0.5	1.3%	0.592 $\pm$0.005	0.803 $\pm$0.004
1.0	1.4%	0.617 $\pm$0.005	0.830 $\pm$0.003
3.0	1.7%	0.644 $\pm$0.003	0.848 $\pm$0.003
5.0	2.6%	0.638 $\pm$0.005	0.846 $\pm$0.003
7.5	6.0%	0.643 $\pm$0.005	0.829 $\pm$0.004
10.0	14.3%	0.630 $\pm$0.005	0.779 $\pm$0.005

Table 4. Comparison of the results with and without control in the sampling process.

The findings indicate that as the guidance weight w increases, the percentage of out-of-key notes rises, suggesting that lower w values yield higher-quality samples. Meanwhile, the OA of duration and note density improves as w increases from 0.5 to 3.0, indicating better alignment with rhythmic conditions. However, when w exceeds 5.0, a notable decline is observed in both the OA metrics and the percentage of out-of-key notes. This degradation is likely due to a significant drop in sample quality at excessively high w values, where unnatural outputs undermine adherence to the rhythmic conditions. These observations are coherent with the existing results about the trade-off between sample quality and adherence to conditions in literature.

I. Discussion

The role of generative AI in music and art remains an intriguing question. While AI has demonstrated remarkable performance in fields such as image generation and language processing, these domains possess two characteristics that symbolic music lacks: an abundance of training data and well-designed objective metrics for evaluating quality. In contrast, for music, it is even unclear whether it is necessary to set the goal as generating compositions that closely resemble[18] some "ground truth".

In this work, we apply fine-grained sampling control to eliminate out-of-key notes, ensuring that generated music adheres to the most common harmonies and chromatic progressions. This approach allows the model to consistently and efficiently produce music that is (in some ways) "pleasing to the ear". While suitable for the task of quickly creating large amounts of mediocre pieces, such models have a limited capability of replicating the artistry of a real composer, of creating sparkles with unexpected "wrong" keys by themselves.

[18] or, in what sense?