NotaGen: Advancing Musicality in Symbolic Music Generation with Large Language Model Training Paradigms

Yashan Wang[1], **Shangda Wu**[1], **Jianhuai Hu**[1], **Xingjian Du**[2], **Yueqi Peng**[3], **Yongxin Huang**[4], **Shuai Fan**[5], **Xiaobing Li**[1], **Feng Yu**[1], **Maosong Sun**[1,6]
[1]Central Conservatory of Music, China,
[2]University of Rochester, USA,
[3]Beijing Flowingtech Ltd., China,
[4]Independent Researcher,
[5]Beihang University, China,
[6]Tsinghua University, China
{alexis_wang, shangda, hujianhuai}@mail.ccom.edu.cn, sms@tsinghua.edu.cn

Abstract

We introduce NotaGen, a symbolic music generation model aiming to explore the potential of producing high-quality classical sheet music. Inspired by the success of Large Language Models (LLMs), NotaGen adopts pre-training, fine-tuning, and reinforcement learning paradigms (henceforth referred to as the LLM training paradigms). It is pre-trained on 1.6M pieces of music in ABC notation, and then fine-tuned on approximately 9K high-quality classical compositions conditioned on "period-composer-instrumentation" prompts. For reinforcement learning, we propose the CLaMP-DPO method, which further enhances generation quality and controllability without requiring human annotations or predefined rewards. Our experiments demonstrate the efficacy of CLaMP-DPO in symbolic music generation models with different architectures and encoding schemes. Furthermore, subjective A/B tests show that NotaGen outperforms baseline models against human compositions, greatly advancing musical aesthetics in symbolic music generation.

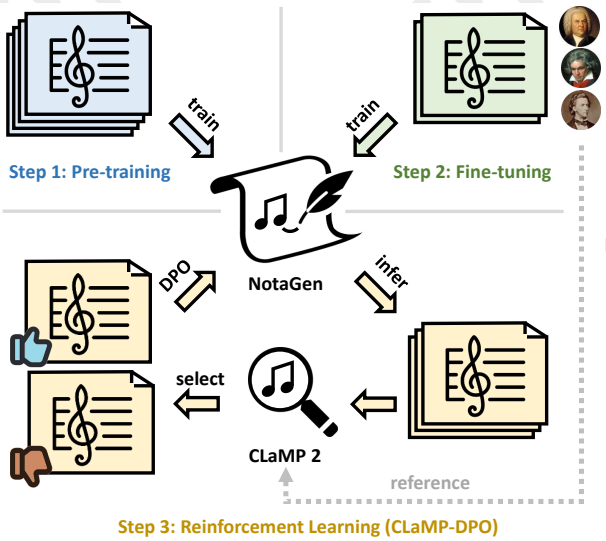


Figure 1: An overview of NotaGen's training paradigms.

1 Introduction

The pursuit of musicality is a core objective in music generation research, as it fundamentally shapes how we perceive and experience musical compositions. Symbolic music abstracts music into discrete symbols such as notes and beats, with performance signals (i.e., MIDI) and sheet music (e.g., ABC notation, MusicXML) being the two predominant modalities. Both of them can efficiently model melody, harmony, instrumentation, etc., all of which are crucial for musicality.

Training tokenized representations with language model architectures, such as Transformers [Vaswani *et al.*, 2017], has emerged as a powerful paradigm for symbolic music generation [Huang *et al.*, 2018; Casini and Sturm, 2022]. However, several challenges persist. First, the scarcity of high-quality music data [Hentschel *et al.*, 2023] hinders the ability of deep learning models to generate sophisticated compositions. Second, when optimizing a language model's loss function, the focus typically lies in minimizing the discrepancy between the predicted and the ground-truth next tokens, potentially neglecting holistic musical aspects like music structure and stylistic coherence.

Insights from the Natural Language Processing (NLP) domain provide a promising approach to overcoming the challenges inherent in symbolic music generation. The success of Large Language Models (LLMs) [Dubey *et al.*, 2024] has established the paradigm of pre-training, fine-tuning, and reinforcement learning as a widely acknowledged framework to improve the quality of text generation and align output with human preferences. These techniques have been successfully adapted for music generation. To overcome the scarcity of high-quality data, large-scale pre-training followed by fine-tuning on smaller, task-specific datasets has been employed

effectively [Donahue *et al.*, 2019; Wu *et al.*, 2024a]. Reinforcement Learning from Human Feedback (RLHF) [Stiennon *et al.*, 2020], transcending next-token prediction approaches, has also shown promising results in music generation [Cideron *et al.*, 2024]. However, to the best of our knowledge, the complete pipeline of LLM training paradigms has not been fully implemented in symbolic music generation. Furthermore, the high cost of RLHF for human annotation highlights the necessity for more efficient and automated solutions.

In this work, we introduce NotaGen (Musical **Nota**tion **Gen**eration), a symbolic music generation model focused on classical sheet music. Compared to MIDI generation, sheet music generation not only aims to produce artistically refined music, but also emphasizes proper voice arrangement and notation to create well-formatted sheets for performance and analysis. Furthermore, the challenge of sheet music generation is exacerbated by the diverse instrumentation and rich musicality inherent in classical music. The success of LLMs has motivated us to apply the training paradigms to sheet music generation. NotaGen is pre-trained on a corpus of over 1.6M sheets in ABC notation, and fine-tuned on a collection of approximately 9K high-quality classical pieces from 152 composers with "period-composer-instrumentation" (e.g."Baroque-Bach, Johann Sebastian-Keyboard") prompts guiding conditional generation. During reinforcement learning, we introduce the CLaMP-DPO method to further optimize NotaGen's musicality and controllability using the Direct Preference Optimization (DPO) [Rafailov *et al.*, 2024] algorithm. In this approach, CLaMP 2 [Wu *et al.*, 2024b], a multimodal symbolic music information retrieval model, assigns generated samples as "chosen" or "rejected" based on references from the fine-tuning dataset. Our contributions are two-fold:

- We introduce NotaGen, a symbolic music generation model implementing LLM training paradigms, which significantly outperforms baseline models in subjective A/B tests against human compositions.
- We propose CLaMP-DPO, a reinforcement learning approach that integrates the DPO algorithm with CLaMP 2 feedback, enhancing musicality and controllability of symbolic music generation without relying on human annotation or predefined rewards. This potential is showcased across symbolic music generation models with varying architectures and encoding schemes.

2 Related Works

2.1 Sheet Music Generation

Sheet music generation has been widely studied, with a focus on encoding methods and composition modeling. Score Transformer [Suzuki, 2021] introduces a tokenized representation for sheet music and applies it to piano music generation. Measure by Measure [Yan and Duan, 2024] models sheet music as grids of part-wise bars and employs hierarchical architectures for generation. Compared to the intricate representations used by the models above, ABC notation, a comprehensive text-based sheet music representation, simplifies encoding and facilitates composition modeling, gaining increasing adoption in recent research. The following models utilize the ABC notation: FolkRNN [Sturm *et al.*, 2016], Tradformer [Casini and Sturm, 2022], and Tunesformer [Wu *et al.*, 2023a], specializing in folk melody generation; MuPT [Qu *et al.*, 2024], a large-scale pre-trained model for sheet music, which explores multitrack symbolic music generation.

2.2 Pre-training in Symbolic Music Generation

The success of pre-training in NLP has inspired the application of this technique in symbolic music generation. LakhNES [Donahue *et al.*, 2019] enhances chiptune music generation by pre-training on the Lakh MIDI Dataset [Raffel, 2016]. MuseBERT [Wang and Xia, 2021] adopts masked language modeling [Devlin *et al.*, 2019], while MelodyGLM [Wu *et al.*, 2023c] implements auto-regressive blank infilling [Du *et al.*, 2021] for generation. MelodyT5 [Wu *et al.*, 2024a] leverages multi-task learning [Raffel *et al.*, 2020]. These studies highlight the effectiveness of pre-training in enhancing music generation performance.

2.3 Reinforcement Learning in Music Generation

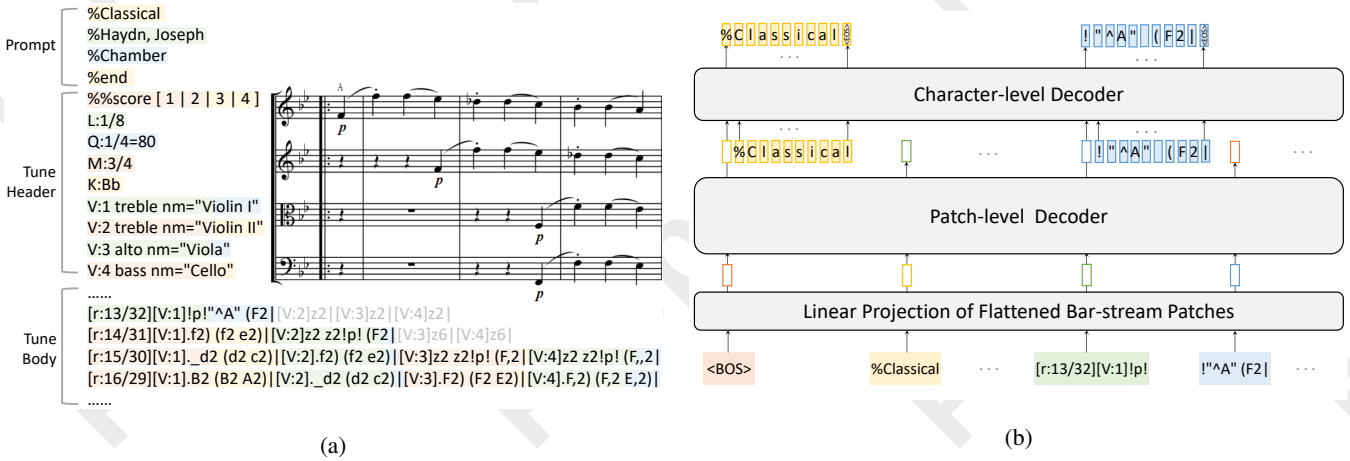
Reinforcement learning has long been recognized as a promising approach for enhancing the musicality of music generation models. It has been successfully applied in RL Tuner [Jaques *et al.*, 2017] for melody generation, RL-Duet [Jiang *et al.*, 2020] for online duet accompaniment, RL-Chord [Ji *et al.*, 2023] for melody harmonization, and [Guo *et al.*, 2022] for multi-track music generation. However, these methods either base their rewards on music theory, which limits its flexibility, or tailor them to specific music styles, hindering their generalization to a broader range of music generation tasks. To tackle this problem, MusicRL [Cideron *et al.*, 2024] adopts the RLHF method with extensive human feedback to align the generated compositions with human preference.

3 NotaGen

3.1 Data Representation

ABC notation sheets consist of two parts: the tune header, which contains metadata such as tempo, time signature, key, and instrumentation; the tune body, where the musical content for each voice is recorded. We adopt a modified version—interleaved ABC notation [Wu *et al.*, 2024b; Qu *et al.*, 2024]. In this format, different voices of the same bar are rearranged into a single line and differentiated using voice indicators "[V:]". This ensures alignment of duration and musical content across voices. Furthermore, we remove bars with full rests (containing only "z" or "x" notes), reducing the length to 80.7% on average, while increasing information density.

We employ stream-based training and inference methods to enable long musical piece generation. We annotate the current and countdown bar indices before each tune body line using the label "[r:]". During training, we randomly segment the tune body and concatenated it with the tune header for longer pieces; during inference, we enforce the generation to start from scratch using the bar annotations. If the piece is incomplete within the current context length, we concatenate the generated tune header with the second half of the tune body and continue generating until the final bar.



(a) (b)

Figure 2: Data representation and model architecture of NotaGen. (a) An example of data representation for an excerpt from *String Quartet in B-flat major, Hob.III:1* by Joseph Haydn using interleaved ABC notation. Bar annotations "[r:]" denote current/countdown bar indices, with gray bars representing omitted rests. Colored backgrounds delineate bar-stream patch boundaries. (b) The model architecture of NotaGen. After passing through the linear projection, bar-stream patches are processed by the patch-level decoder to generate features for a character-level decoder, which performs auto-regressive character prediction.

3.2 Model Architecture

NotaGen utilizes bar-stream patching [Wang *et al.*, 2024] and the Tunesformer architecture [Wu *et al.*, 2023a]. Building upon bar patching [Wu *et al.*, 2023b], bar-stream patching divides the tune header lines and bars into fixed-length patches (padded when necessary), striking a balance between musicality of generation and computational efficiency among sheet music tokenization methods.

NotaGen consists of two hierarchical GPT-2 decoders [Radford *et al.*, 2019]: a patch-level decoder and a character-level decoder. Each patch is flattened by concatenating one-hot character vectors and then passed through a linear layer to obtain the patch embedding. The patch-level decoder captures the temporal relationships among patches, and its final hidden states are passed to the character-level decoder, which auto-regressively predicts the characters of the next patch. The data representation and model architecture are illustrated in Figure 2.

3.3 Training Paradigms

Pre-training

Pre-training enables NotaGen to capture fundamental musical structures and patterns through next-token prediction on a large, diverse dataset spanning various genres and instrumentations.

The pre-training stage utilized a carefully curated internal-use dataset comprising 1.6M ABC notation sheets. We also preprocessed the text annotations, retaining music-related content such as tempo and expression hints, while removing irrelevant content like lyrics and background information.

All music sheets were transposed to 15 keys (including F♯, G♭, C♯, C♭) for data augmentation. During training, a randomly selected key was used for each piece in every epoch.

Fine-tuning

NotaGen was fine-tuned on high-quality classical music sheet data to further enhance musicality in generation. Spanning from the intricate contrapuntal orchestra suites of the Baroque period to the melodious and harmonically nuanced piano pieces of the Romantic era, classical music encompasses a diverse array of compositional styles and instrumentations, all characterized by exceptional musicality.

Thus, we curated a fine-tuning dataset comprising 8,948 classical music sheets, from DCML corpora [Neuwirth *et al.*, 2018; Hentschel *et al.*, 2021a; Hentschel *et al.*, 2021b; Hentschel *et al.*, 2023], OpenScore String Quartet Corpus [Gotham *et al.*, 2023], OpenScore Lieder Corpus [Gotham and Jonas, 2022], ATEPP [Zhang *et al.*, 2022], KernScores [Sapp, 2005], and internal resources, as listed in Table 1. Sheets with more than 16 staves were excluded due to generation complexity. Each work was assigned with three labels: period, composer and instrumentation. The data distribution is provided in supplementary materials, and the details of each label are explained as follows:

- **Period**:
  - **Baroque** (1600s-1750s): e.g., Bach, Vivaldi.
  - **Classical** (1750s-1810s): e.g., Mozart, Beethoven.
  - **Romantic** (1810s-1950s): e.g., Chopin, Liszt.
- **Composer**: The official names of a total of 152 composers, as listed on IMSLP[1], were included.
- **Instrumentation**:
  - **Keyboard**: piano and organ works.
  - **Chamber**: instrumental music typically for a small group of performers, each playing a unique part.
  - **Orchestral**: instrumental music for orchestra.

[1]https://imslp.org/

Data Sources	Amount
DCML Corpora	560
OpenScore String Quartet Corpus	342
OpenScore Lieder Corpus	1,334
ATEPP	55
KernScores	221
Internal Sources	6,436
Total	8,948

Table 1: Data sources and the respective amounts for fine-tuning.

- **Art Song**: vocal music typically for solo or duet voices with piano accompaniment.
- **Choral**: vocal music for a choir.
- **Vocal-Orchestral**: works involving both vocal and orchestral elements, including Cantata, Oratorio, and Opera.

In fine-tuning, a "period-composer-instrumentation" prompt was prepended to each piece for conditional generation. This approach challenges NotaGen to produce high-quality compositions, imitate the styles of composers across different periods, and conform to specified instrumentation requirements.

To facilitate NotaGen's learning of appropriate pitch ranges for each instrument while optimizing data utilization, data augmentation during fine-tuning was restricted to the six nearest key transpositions of the original. Keys farther from the original were selected with decreasing probability.

Reinforcement Learning

To refine both the musicality and the prompt controllability of the fine-tuned NotaGen, we present CLaMP-DPO. This method builds upon the principles of Reinforcement Learning from AI Feedback (RLAIF) [Lee *et al.*, 2024] and implements Direct Preference Optimization (DPO) [Rafailov *et al.*, 2024]. In CLaMP-DPO, CLaMP 2 serves as the evaluator within the DPO framework, distinguishing between chosen and rejected musical outputs to optimize NotaGen.

CLaMP 2 is a multimodal symbolic music information retrieval model supporting both ABC notation and MIDI formats. Leveraging contrastive learning, CLaMP 2 extracts semantic features that encapsulate global musical properties. These features encompass comprehensive musical information, including style, instrumentation, and compositional complexity. Meanwhile, they are consistent with human subjective perceptions, as validated by [Retkowski *et al.*, 2024]. In the context of music generation, the objective is to produce pieces which closely resemble the ground truth. Accordingly, it is critical to ensure the alignment of the semantic features between the generated pieces and authentic references.

We introduce the CLaMP 2 Score to quantify the similarity among pieces. To elaborate, we denote P as the set of prompts for NotaGen. For each prompt $p \in P$, Y_p represents the corresponding set of ground truth with an average semantic feature $\bar{z}_p$. Similarly, each prompt p has a generated set X_p, where each piece x_p is associated with a semantic feature z_{x_p}.

The CLaMP 2 Score c for a generated piece x_p is defined as the cosine similarity between z_{x_p} and $\bar{z}_p$:

$$c_{x_p} = \frac{z_{x_p} \cdot \bar{z_p}}{\|z_{x_p}\| \|\bar{z_p}\|}. \tag{1}$$

Our goal is to maximize the average, $\bar{c}_{x_p}$ over X_p, thereby ensuring the music generated for prompt p aligns semantically with the ground truth. It is achieved by employing the DPO algorithm to improve $\bar{c}_{x_p}$.

The DPO algorithm optimizes a language model based on preference data, which consists of paired chosen and rejected examples under the same prompts. It eliminates the need of explicit reward modeling. In the proposed CLaMP-DPO algorithm, the fine-tuned model first generates data across the prompt set P. For each generated set X_p, the pieces $x_p \in X_p$ are sorted according to c_{x_p}, with the top 10% selected as chosen set X_{pw} and the bottom 10% as rejected set X_{pl}. Additional criteria, such as syntax error checks or the exclusion of ground-truth plagiarism, can be applied to refine these sets. Finally, the chosen and rejected pairs (x_{pw}, x_{pl}) are randomly selected and combined into preference data for optimization.

Given a prompt p, an auto-regressive language model predicts the next token based on its policy π_θ, where θ represents the model parameters. The probability of generating a chosen data x_{pw} is $\pi_\theta(x_{pw}|p)$, and that of generating a rejected data x_{pl} is $\pi_\theta(x_{pl}|p)$. To prevent excessive drift from the initial model that generates the preference data and ensure diversity in the generated content, the initial model policy, i.e., the reference model policy π_{ref} is introduced and kept frozen during optimization. The objective function to be minimized in DPO is given by:

$$\begin{aligned}\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(p,x_{pw},x_{pl})\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\log\frac{\pi_\theta(x_{pw}|p)}{\pi_{\text{ref}}(x_{pw}|p)} \\ - \beta\log\frac{\pi_\theta(x_{pl}|p)}{\pi_{\text{ref}}(x_{pl}|p)}\Big)\Big],\end{aligned} \tag{2}$$

where σ is the sigmoid function, $\mathcal{D}$ is the preference dataset, and β is the hyperparameter that controls the deviation between π_θ and π_{ref}.

The optimization process increases the relative log probability of chosen data over rejected data. However, we observed a decrease in $\pi_\theta(x_{pw}|p)$, leading to degraded outputs. To mitigate this issue, we adopt the DPO-Positive (DPOP) objective function [Pal *et al.*, 2024], which incorporates a penalty term to stabilize $\pi_\theta(x_{pw}|p)$:

$$\begin{aligned}\mathcal{L}_{\text{DPOP}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(p,x_{pw},x_{pl})\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\log\frac{\pi_\theta(x_{pw}|p)}{\pi_{\text{ref}}(x_{pw}|p)} \\ - \beta\log\frac{\pi_\theta(x_{pl}|p)}{\pi_{\text{ref}}(x_{pl}|p)} \\ - \beta\lambda\cdot\max\Big(0, \log\frac{\pi_{\text{ref}}(x_{pw}|p)}{\pi_\theta(x_{pw}|p)}\Big)\Big)\Big],\end{aligned} \tag{3}$$

where the hyperparameter λ controls the impact of penalty.

Algorithm 1: Iterative CLaMP-DPO

Input: Fine-tuned policy π_θ^0, CLaMP 2 model C, prompt set P, fine-tuning dataset Y
Parameter: Iterations K, DPO hyperparameter β, DPOP hyperparameter λ, optimization steps N, learning rate η
Output: Optimized policy π_θ^K

```
# Initialize ground-truth features
```
foreach prompt $p \in P$ **do**
  $\bar{z}_p \leftarrow \text{Avg}(C(y_p)), \quad \forall y_p \in Y_p$
end
```
# Iterative Optimization
```
for $k \leftarrow 1$ **to** K **do**
  ```
  # Construct preference data
  ```
  foreach prompt $p \in P$ **do**
    $X_p^{k-1} \leftarrow \pi_\theta^{k-1}(p)$ `# Generate on p`
    foreach piece $x_p^{k-1} \in X_P^{k-1}$ **do**
      $z_{x_p^{k-1}} \leftarrow C(x_p^{k-1})$
      $c_{x_p^{k-1}} \leftarrow$ Eq. (1)$(z_{x_p^{k-1}}, \bar{z}_p)$
    end
    $X_{pw}^{k-1}, X_{pl}^{k-1} \leftarrow \text{Select}(X_p^{k-1}, \text{Sort}(c_{x_p^{k-1}}))$
  end
  ```
  # Optimize using DPO
  ```
  $\pi_{\text{ref}} \leftarrow \pi_\theta^{k-1}$
  for $i \leftarrow 1$ **to** N **do**
    Sample prompt $p \sim P$
    Sample pairs $(x_{pw}, x_{pl}) \sim (X_{pw}^{k-1}, X_{pl}^{k-1})$
    $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{DPOP}}(\pi_\theta, \pi_{\text{ref}}, x_{pw}, x_{pl}, p, \beta, \lambda)$
  end
  $\pi_\theta^k \leftarrow \pi_\theta$
end
return π_θ^K

The fine-tuned model is optimized by minimizing $\mathcal{L}_{\text{DPOP}}$ in Eq.(3) for a specified number of steps, completing the process of CLaMP-DPO algorithm. Notably, CLaMP-DPO supports iterative optimization. After the first round, the model generates a new set X'_p. Using CLaMP 2, we construct new chosen and rejected sets, X'_{pl} and X'_{pw}, allowing the model to undergo further optimization via Eq.(3).

4 Experiments

4.1 Settings

The experiments are divided into two parts. The first part assesses CLaMP-DPO's ability to improve the controllability and musicality of symbolic music models. The second part compares the musicality of NotaGen with baseline models. Along with the pre-trained NotaGen, we selected two pre-trained symbolic music generation models as baselines: MuPT [Qu *et al.*, 2024] and Music Event Transformer (MET) [2][SkyTNT, 2024]. All models adopt language model architectures and are trained auto-regressively. A brief overview of their architectures and pre-training procedures follows:

- **NotaGen** features a 20-layer patch-level decoder and a 6-layer character-level decoder, with a context length of 1024 and a hidden size of 1280, totaling 516M parameters. It was pre-trained on 1.6M ABC notation sheets, augmented to 15 key transpositions. The AdamW optimizer [Loshchilov and Hutter, 2019] was utilized with a learning rate of 1e-4 and a 1,000-step warm-up phase. The pre-training was performed on 8 NVIDIA H800 GPUs, with a batch size of 4 per GPU.
- **MuPT** utilizes Synchronized Multi-Track ABC notation (SMT-ABC) as data representation. SMT-ABC is equivalent to interleaved ABC notation, as both merge multi-track voices into a single sequence. Byte Pair Encoding (BPE) is used for tokenization. MuPT-v1-8192-550M, the chosen baseline model, consists of a 16-layer Transformer decoder with a hidden size of 1024 and a context length of 8192, totaling 505M parameters. MuPT was pre-trained on a corpus of 33.6B tokens.
- **MET** encodes MIDI events into token sequences and uses hierarchical Transformer decoders for generation, including a event-level decoder and a token-level decoder. Details on the encoding and model architecture are provided in the supplementary materials. MET consists of a 12-layer event-level decoder and a 3-layer token-level decoder, with a context length of 4096 and a hidden size of 1024, totaling 234M parameters. It was pre-trained on three MIDI datasets: Los Angeles MIDI Dataset [Lev, 2024], Monster MIDI Dataset [3], and SymphonyNet Dataset [Liu *et al.*, 2022].

We applied fine-tuning and reinforcement learning to these models using their pre-trained weights.

Fine-tuning. The fine-tuning dataset for NotaGen and MuPT comprises 8,948 classical music pieces, referred to as the sheet ground truth set (sheet-GT). All data were formatted to match the pre-training data of different models, each preceded by a "period-composer-instrumentation" prompt. Due to the challenges in converting between MIDI and ABC notation, only the keyboard subset, consisting of 3,104 pieces was used for fine-tuning MET, referred to as the MIDI ground truth set (MIDI-GT). Each piece was preceded by a "period-composer" prompt.

Reinforcement learning. Considering that the accuracy of prompt semantic feature $\bar{z}_p$ in CLaMP-DPO relies on a sufficient amount of ground truth data in Y_p, we defined the prompt set P to only include prompts p that appear more than ten times in the fine-tuning dataset ($Y_p > 10$). The detailed list of P can be referred in supplementary materials. For NotaGen and MuPT, P contained 112 prompts, covering 86.4% of the data; for MET, P contained 29 prompts, covering 90.5%. The number of iterations K was set to 3, with approximately 100 pieces generated per prompt as X_p in each iteration. The chosen and rejected sets were constructed based on CLaMP 2 Scores. Sheets where staves for the same

[2]https://huggingface.co/skytnt/midi-model-tv2o-medium

[3]https://huggingface.co/datasets/projectlosangeles/Monster-MIDI-Dataset

instrument were not grouped together were excluded from the chosen set. The hyperparameters $\beta = 0.1$ and $\lambda = 10$ were used, with $N = 10,000$ optimization steps. The learning rate was fixed at 1e-6 for NotaGen and MET, and 1e-7 for MuPT, yielding stable CLaMP-DPO performance.

Given the challenge of establishing objective metrics that fully capture musicality, we conducted subjective A/B tests in both experiments to evaluate different models and settings. For each question, two pieces were generated using identical prompts; videos were rendered from sheet music using Sibelius and MIDI files using MIDIVisualizer[4]. Participants were instructed to evaluate musicality from multiple perspectives and select the piece they found more musically appealing, or indicate no preference if they perceived no differences. The evaluation criteria included melodic appeal, harmonic fluency, orchestral balance, counterpoint correctness, and structural coherence, and, for sheet music, notation formatting quality. A total of 92 participants from music colleges took part in the assessment. At least 35 valid responses were recorded for each test group, ensuring statistical reliability.

4.2 Ablation Studies on CLaMP-DPO

This experiment evaluates the impact of the proposed CLaMP-DPO algorithm in enhancing the controllability and musicality of generated music for NotaGen, MuPT, and MET. In the objective assessment, we selected several metrics for both the fine-tuned models (denoted as $K = 0$) and the models after K iterations of CLaMP-DPO optimization. We also assessed a subset of these metrics on the fine-tuning datasets (sheet-GT and MIDI-GT) for reference. The metrics are as follows:

- **Average CLaMP 2 Score (ACS)**: The average CLaMP 2 Score across generated outputs. For sheet-GT and MIDI-GT, the score is computed over the corpus.
- **Label Accuracy (LA)**: The alignment with specified period (per.) and instrumentation (inst.) prompts. We extracted features from the fine-tuning dataset via a multimodal symbolic music encoder—M3[Wu *et al.*, 2024b], then trained two linear classifiers to predict the period and instrumentation labels. LA is defined as the classification accuracy, where for the fine-tuning dataset, it measures the accuracy on the test set, and for generated outputs, it reflects the match between predicted labels and prompt labels.
- **Bar Alignment Error (BAE)**: The proportion of bars where duration is misaligned, occurring in either the generated output or the fine-tuning corpus. This metric applies only to sheet data.
- **Perplexity (PPL)**: A language model metric, where lower PPL indicates better prediction capability.

We conducted subjective A/B tests on each model before and after three optimization iterations with CLaMP-DPO to appraise its efficacy in enhancing the musical quality of generated outputs. The results of the objective and subjective tests are presented in Table 2 and Figure 3, respectively.

[4]https://github.com/kosua20/MIDIVisualizer

Models & Data	*K*	ACS	LA (%)		BAE (%)	PPL
			Per.	Inst.		
sheet-GT	-	0.792	96.1	95.5	0.377	-
NotaGen	0	0.570	84.7	78.5	0.269	**1.2151**
	1	0.674	92.1	87.8	0.175	1.2341
	2	0.708	**93.3**	92.9	**0.158**	1.2614
	3	**0.730**	93.0	**94.6**	0.176	1.2880
MuPT	0	0.515	76.3	78.6	**0.824**	**1.4159**
	1	0.596	78.8	86.2	1.520	1.4476
	2	0.631	80.3	**89.2**	2.601	1.5214
	3	**0.674**	**82.1**	87.6	4.676	1.6121
MIDI-GT	-	0.812	92.9	-	-	-
MET	0	0.565	30.0	-	-	**1.2251**
	1	0.609	34.6	-	-	1.2261
	2	0.637	36.7	-	-	1.2255
	3	**0.655**	**38.2**	-	-	1.2290

Table 2: Objective metrics on fine-tuned models and the models after each iteration of CLaMP-DPO optimization. Some of metrics were also assessed on the fine-tuning dataset for reference.

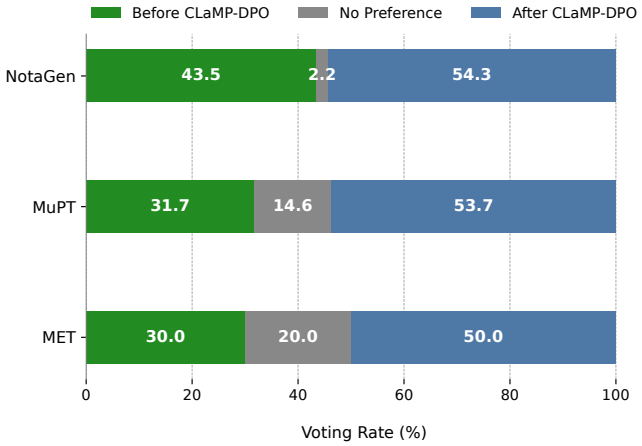


Figure 3: Subjective A/B tests on musicality of generated outputs before and after CLaMP-DPO optimization. All models exhibited improvement in human-perveiced musicality after applying the CLaMP-DPO algorithm.

The ACS, as the primary optimization goal, exhibited a monotonic increase across iterations of its DPO-based process. Though significant improvements were observed in early iterations, subsequent gains exhibited diminishing returns.

LA for period and instrumentation classification exceeded 90% on the test set of fine-tuning data, validating the reliability of the label assignments and the performance of the classifiers. Following the CLaMP-DPO method, all models demonstrated a noticeable improvement in LA, indicating enhanced prompt controllability and better alignment with the intended musical styles. NotaGen exhibited the highest controllability among the models, further confirming its superior adaptability to specified prompts.

Regarding BAE, NotaGen maintained a relatively low error rate throughout optimization, indicating its character-level

prediction is more robust at managing duration consistency. In contrast, MuPT's increased error rate is likely due to the use of BPE tokenization, which may merge duration with other musical elements, such as pitch, into single tokens. It may lead to inaccuracies in duration prediction after CLaMP-DPO adjusts token probabilities.

Subjective A/B tests showed that all models exhibited improvement in musicality after applying the CLaMP-DPO algorithm, with post-optimization outputs receiving more votes than their pre-optimization counterparts. However, it is noteworthy that PPL increased after optimization. It suggests that PPL may not be a suitable indicator for model performance in symbolic music generation, highlighting the limitations of traditional language model metrics in assessing musical quality.

In summary, the CLaMP-DPO algorithm efficiently enhanced both the controllability and the musicality across three models, irrespective of their data modalities, encoding schemes, or model architectures. This underscores CLaMP-DPO's broad applicability and potential for auto-regressively trained symbolic music generation models.

4.3 Comparative Evaluations

This experiment compares the musicality of three models after the LLM training paradigm. For baseline comparison, we constructed the reference set using human-authored musical pieces from the fine-tuning dataset, which represent professional compositional standards. The subjective A/B tests were organized into three groups, each containing the generated results of a model and the ground truth. For comparison involving MET, all data were converted to MIDI to eliminate format-based bias. The results are shown in Figure 4.

Human compositions consistently outperformed all model-generated outputs in voting due to their exceptional musicality. Nevertheless, NotaGen achieved the highest voting rate against the ground truth among the three models, suggesting its superior perceived musicality relative to other systems in human evaluations.

Overall, NotaGen outperformed the baseline models. The superior performance of NotaGen compared to MuPT is attributed to well-designed data representation and tokenization. Despite its architectural similarities to MET, NotaGen achieved better musicality, benefiting from the efficiency and structural integrity of sheet music representation compared to MIDI.

5 Limitations and Challenges

While NotaGen shows promising advancements in symbolic music generation, limitations and challenges still warrant discussion.

We once introduced a post-training stage between pre-training and fine-tuning, refining the model with classical-style subset of the pre-training dataset. While it accelerated the fine-tuning convergence and improved ACS for NotaGen, the impact was less pronounced on MuPT and MET.

Furthermore, the prerequisite for evaluating generated results using CLaMP 2 Score is that the model has been well trained and is capable of generating reasonable compositions.

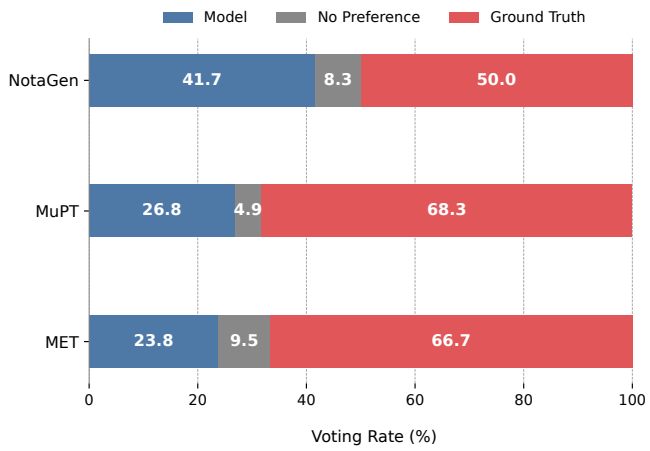


Figure 4: Subjective A/B test between model outputs and ground truth. NotaGen achieved the highest voting rate against the ground truth among the three models.

For corrupted or syntactically flawed pieces, the CLaMP 2 Score may not reliably indicate the true musical similarity.

Finally, we found that modeling orchestral music presents greater challenges compared to smaller ensembles (e.g. solo piano or string quartets). While rest-bar omission during data pre-processing addresses the degeneration due to excessive blank bars in ensemble writing, NotaGen's performance in orchestral music generation still lags behind. More effective methods are expected for generating large ensemble compositions.

6 Conclusions

In this work, we present NotaGen, a symbolic music generation model designed to advance the musicality of generated outputs through a comprehensive LLM-inspired training paradigm. By integrating pre-training, fine-tuning, and reinforcement learning with the proposed CLaMP-DPO algorithm, NotaGen demonstrates superior performance in generating compositions that align with both the music style specified by prompts and human-perceived musicality. Our experiments validate two key findings: (1) CLaMP-DPO efficaciously enhances controllability and musicality across diverse symbolic music models, regardless of their modality, architectures, or encoding schemes, without requiring human annotations or predefined rewards; (2) NotaGen outperforms baseline models in subjective evaluations, achieving the highest voting rate against human-composed ground truth.

NotaGen establishes the viability of adapting LLM training paradigms to symbolic music generation, while addressing domain-specific challenges, including data scarcity and demand for high-quality music outputs. Future work could extend this framework with CLaMP-DPO to broader musical genres such as jazz, pop, and ethnic music; as well as exploring its compatibility with emerging music generation models.

Acknowledgments

We would like to express our sincere gratitude to SkyTNT for his valuable support on this project. We also acknowledge Yuling Yang, Xinran Zhang, Jiafeng Liu, Yuqing Cheng, and Yuhao Ding from Central Conservatory of Music for their support, especially on subjective tests and paper writing.

This work was supported by the following funding sources: Special Program of National Natural Science Foundation of China (Grant No. T2341003), Advanced Discipline Construction Project of Beijing Universities, Major Program of National Social Science Fund of China (Grant No. 21ZD19), and the National Culture and Tourism Technological Innovation Engineering Project (Research and Application of 3D Music).

Contribution Statement

Yashan Wang, Shangda Wu, and Jianhuai Hu are co-first authors with equal contribution. Maosong Sun is the corresponding author.

References

[Casini and Sturm, 2022] Luca Casini and Bob Sturm. Tradformer: A transformer model of traditional music transcriptions. In *International Joint Conference on Artificial Intelligence IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 4915–4920, 2022.

[Cideron *et al.*, 2024] Geoffrey Cideron, Sertan Girgin, Mauro Verzetti, Damien Vincent, Matej Kastelic, Zalán Borsos, Brian McWilliams, Victor Ungureanu, Olivier Bachem, Olivier Pietquin, et al. Musicrl: Aligning music generation to human preferences. *arXiv preprint arXiv:2402.04229*, 2024.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2. Minneapolis, Minnesota, 2019.

[Donahue *et al.*, 2019] Chris Donahue, Huanru Henry Mao, Yiting Ethan Li, Garrison W Cottrell, and Julian McAuley. Lakhnes: Improving multi-instrumental music generation with cross-domain pre-training. *arXiv preprint arXiv:1907.04868*, 2019.

[Du *et al.*, 2021] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360*, 2021.

[Dubey *et al.*, 2024] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[Gotham and Jonas, 2022] Mark Robert Haigh Gotham and Peter Jonas. The openscore lieder corpus. In *Music Encoding Conference Proceedings 2021, 19–22 July, 2021 University of Alicante (Spain): Onsite & Online*, pages 131–136. Universidad de Alicante/Universitat d'Alacant, 2022.

[Gotham *et al.*, 2023] Mark Gotham, Maureen Redbond, Bruno Bower, and Peter Jonas. The "openscore string quartet" corpus. In *Proceedings of the 10th International Conference on Digital Libraries for Musicology*, pages 49–57, 2023.

[Guo *et al.*, 2022] Xuefei Guo, Hongguang Xu, and Ke Xu. Fine-tuning music generation with reinforcement learning based on transformer. In *2022 IEEE 16th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pages 1–5. IEEE, 2022.

[Hentschel *et al.*, 2021a] Johannes Hentschel, Fabian Claude Moss, Markus Neuwirth, and Martin Rohrmeier. A semi-automated workflow paradigm for the distributed creation and curation of expert annotations. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 2021.

[Hentschel *et al.*, 2021b] Johannes Hentschel, Markus Neuwirth, and Martin Rohrmeier. The annotated mozart sonatas: Score, harmony, and cadence. *Transactions of the International Society for Music Information Retrieval*, 4(1):67–80, 2021.

[Hentschel *et al.*, 2023] Johannes Hentschel, Yannis Rammos, Fabian C Moss, Markus Neuwirth, and Martin Rohrmeier. An annotated corpus of tonal piano music from the long 19th century. *Empirical Musicology Review*, 18(1):84–95, 2023.

[Huang *et al.*, 2018] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer. *arXiv preprint arXiv:1809.04281*, 2018.

[Jaques *et al.*, 2017] Natasha Jaques, Shixiang Gu, Richard E Turner, and Douglas Eck. Tuning recurrent neural networks with reinforcement learning. 2017.

[Ji *et al.*, 2023] Shulei Ji, Xinyu Yang, Jing Luo, and Juan Li. Rl-chord: Clstm-based melody harmonization using deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[Jiang *et al.*, 2020] Nan Jiang, Sheng Jin, Zhiyao Duan, and Changshui Zhang. Rl-duet: Online music accompaniment generation using deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 710–718, 2020.

[Lee *et al.*, 2024] Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: scaling reinforcement learning from human feedback with AI feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[Lev, 2024] Aleksandr Lev. Los angeles midi dataset: Sota kilo-scale midi dataset for mir and music ai purposes. In *GitHub*, 2024.

[Liu *et al.*, 2022] Jiafeng Liu, Yuanliang Dong, Zehua Cheng, Xinran Zhang, Xiaobing Li, Feng Yu, and Maosong Sun. Symphony generation with permutation invariant language model. *arXiv preprint arXiv:2205.05448*, 2022.

[Loshchilov and Hutter, 2019] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2019.

[Neuwirth *et al.*, 2018] Markus Neuwirth, Daniel Harasim, Fabian C Moss, and Martin Rohrmeier. The annotated beethoven corpus (abc): A dataset of harmonic analyses of all beethoven string quartets. *Frontiers in Digital Humanities*, 5:379513, 2018.

[Pal *et al.*, 2024] Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

[Qu *et al.*, 2024] Xingwei Qu, Yuelin Bai, Yinghao Ma, Ziya Zhou, Ka Man Lo, Jiaheng Liu, Ruibin Yuan, Lejun Min, Xueling Liu, Tianyu Zhang, et al. Mupt: A generative symbolic music pretrained transformer. *arXiv preprint arXiv:2404.06393*, 2024.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[Rafailov *et al.*, 2024] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[Raffel, 2016] Colin Raffel. *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*. Columbia University, 2016.

[Retkowski *et al.*, 2024] Jan Retkowski, Jakub Stępniak, and Mateusz Modrzejewski. Frechet music distance: A metric for generative symbolic music evaluation. *arXiv preprint arXiv:2412.07948*, 2024.

[Sapp, 2005] Craig Stuart Sapp. Online database of scores in the humdrum file format. In *ISMIR*, pages 664–665, 2005.

[SkyTNT, 2024] SkyTNT. Midi model: Midi event transformer for symbolic music generation. https://github.com/SkyTNT/midi-model, 2024.

[Stiennon *et al.*, 2020] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[Sturm *et al.*, 2016] Bob L Sturm, Joao Felipe Santos, Oded Ben-Tal, and Iryna Korshunova. Music transcription modelling and composition using deep learning. *arXiv preprint arXiv:1604.08723*, 2016.

[Suzuki, 2021] Masahiro Suzuki. Score transformer: Generating musical score from note-level representation. In *Proceedings of the 3rd ACM International Conference on Multimedia in Asia*, pages 1–7, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Lukasz Kaiser Jones, Aidan Nating, and Illia Gomez, & Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008. Curran Associates, Inc., 2017.

[Wang and Xia, 2021] Ziyu Wang and Gus Xia. Musebert: Pre-training music representation for music understanding and controllable generation. In *ISMIR*, pages 722–729, 2021.

[Wang *et al.*, 2024] Yashan Wang, Shangda Wu, Xingjian Du, and Maosong Sun. Exploring tokenization methods for multitrack sheet music generation. *arXiv preprint arXiv:2410.17584*, 2024.

[Wu *et al.*, 2023a] Shangda Wu, Xiaobing Li, Feng Yu, and Maosong Sun. Tunesformer: Forming irish tunes with control codes by bar patching. *arXiv preprint arXiv:2301.02884*, 2023.

[Wu *et al.*, 2023b] Shangda Wu, Dingyao Yu, Xu Tan, and Maosong Sun. Clamp: Contrastive language-music pretraining for cross-modal symbolic music information retrieval. *arXiv preprint arXiv:2304.11029*, 2023.

[Wu *et al.*, 2023c] Xinda Wu, Zhijie Huang, Kejun Zhang, Jiaxing Yu, Xu Tan, Tieyao Zhang, Zihao Wang, and Lingyun Sun. Melodyglm: multi-task pre-training for symbolic melody generation. *arXiv preprint arXiv:2309.10738*, 2023.

[Wu *et al.*, 2024a] Shangda Wu, Yashan Wang, Xiaobing Li, Feng Yu, and Maosong Sun. Melodyt5: A unified score-to-score transformer for symbolic music processing. *arXiv preprint arXiv:2407.02277*, 2024.

[Wu *et al.*, 2024b] Shangda Wu, Yashan Wang, Ruibin Yuan, Zhancheng Guo, Xu Tan, Ge Zhang, Monan Zhou, Jing Chen, Xuefeng Mu, Yuejie Gao, et al. Clamp 2: Multimodal music information retrieval across 101 languages using large language models. *arXiv preprint arXiv:2410.13267*, 2024.

[Yan and Duan, 2024] Yujia Yan and Zhiyao Duan. Measure by measure: Measure-based automatic music composition with modern staff notation. *Transactions of the International Society for Music Information Retrieval*, Nov 2024.

[Zhang *et al.*, 2022] Huan Zhang, Jingjing Tang, Syed RM Rafee, Simon Dixon, George Fazekas, and Geraint A Wiggins. Atepp: A dataset of automatically transcribed expressive piano performance. In *Ismir 2022 Hybrid Conference*, 2022.