Compose with Me: Collaborative Music Inpainter for Symbolic Music Infilling

Zhejing Hu[1], Yan Liu[1]*, Gong Chen[1], Bruce X.B. Yu[2],

[1] Department of Computing, The Hong Kong Polytechnic University
[2] Zhejiang University-University of Illinois Urbana-Champaign Institute
zhejing.hu@connect.polyu.hk, yan.liu@polyu.edu.hk, gong-cg.chen@polyu.edu.hk, xinboyu@intl.zju.edu.cn

Abstract

The field of music generation has seen a surge of interest from both academia and industry, with innovative platforms such as Suno, Udio, and SkyMusic earning widespread recognition. However, the challenge of music infilling—modifying specific music segments without reconstructing the entire piece—remains a significant hurdle for both audio-based and symbolic-based models, limiting their adaptability and practicality. In this paper, we address symbolic music infilling by introducing the Collaborative Music Inpainter (CMI), an advanced human-in-the-loop (HITL) model for music infilling. The CMI features the Joint Embedding Predictive Autoregressive Generative Architecture (JEP-AGA), which learns the high-level predictive representations of the masked part that needs to be infilled during the autoregressive generative process, akin to how humans perceive and interpret music. The newly developed Dynamic Interaction Learner (DIL) achieves HITL by iteratively refining the infilled output based on user interactions alone, significantly reducing the interaction cost without requiring further input. Experimental results confirm CMI's superior performance in music infilling, demonstrating its efficiency in producing high-quality music.

Code — https://github.com/hu-music/compose_with_me

Introduction

Music generation, a pivotal aspect of artificial intelligence, has gained immense significance with the rise of deep generative models (Ji and Yang 2024; Copet et al. 2024; Lam et al. 2024; Hu et al. 2022, 2023; Yu et al. 2022). Recent music generation products like Suno, Udio and SkyMusic[1] have captured widespread attention beyond research circles, showcasing AI's prowess in music generation. These models have significantly lowered the barrier to music creation and found application across various domains such as film, gaming, and advertising.

Despite these advancements, current music generation products and models, including both audio-based and symbolic-based generation, lack flexibility and struggle to accommodate user demands after generation. For example, users often face challenges in modifying specific parts of generated music without regenerating the entire piece. Addressing this challenge involves refining parts that fail to meet user satisfaction, a task known as *music infilling* or *inpainting*. This process bridges the gaps between existing past and future musical contexts (Chang, Lee, and Yang 2021).

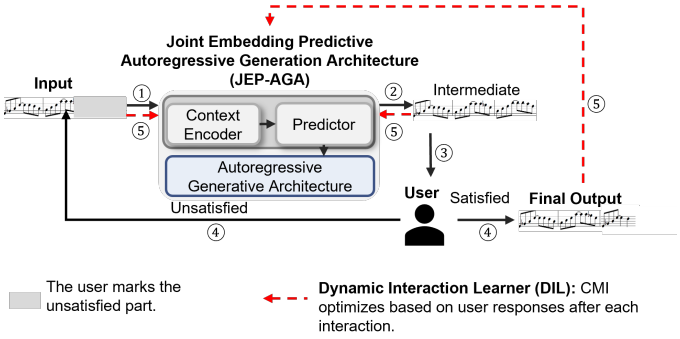

Figure 1: **Human-in-the-loop (HITL) in CMI**: it establishes a user-model loop where the model generates an intermediate output based on the user's input. The user then marks any unsatisfactory parts, which the model regenerates until satisfaction is achieved. We introduce the novel Joint Embedding Predictive Autoregressive Generative Architecture (JEP-AGA), which learns high-level predictive representations of the masked part during the autoregressive generative process. The Dynamic Interaction Learner (DIL) dynamically updates the encoder-predictor structure of JEP-AGA after each user interaction, enhancing the quality of music infilling, as illustrated by the red arrow.

Within the task of music infilling, researchers have made various efforts and achieved significant improvements in both audio-based (Lin et al. 2024) and symbolic-based music infilling (Hadjeres, Pachet, and Nielsen 2017; Ippolito et al. 2018; Huang et al. 2017; Chen et al. 2020; Wei et al. 2022; Min et al. 2023). Symbolic-based music infilling, which allows for modifications from note-level to song-level by users and machines, is more straightforward than audio-based approaches that use waveforms. As a result, many works have focused on symbolic music infilling, and this work also falls within the scope of symbolic music infilling. Recently, with the de-

*Corresponding author.

[1]www.suno.com; www.udio.com; music.tiangong.cn

velopment of large language models, various models have been proposed to infill music and have achieved good performance. For example, Chang et al. (Chang, Lee, and Yang 2021) introduced an XLNet-based model capable of generating polyphonic pop music up to 768 tokens (128 notes). The MMM (Ens and Pasquier 2020) is a GPT2-based model that can infill up to 2048 tokens. Additionally, Malandro (Malandro 2023) proposed a T5-based model that handles song inputs of up to 512 or 1650 tokens. However, these models generally overlook the crucial role of continuous human interaction in the music infilling task. Music, as an art form deeply intertwined with human emotion and expression, requires the ability to continuously manipulate specific elements based on user responses. The absence of a mechanism that adapts based on user responses might lead to the generation of random results after each interaction, imposing greater human effort and reducing overall usability and controllability.

To address this challenge, this paper proposes a novel Collaborative Music Inpainter (CMI) model that utilizes human-in-the-loop (HITL) technology for symbolic music infilling. As shown in Figure 1, the user's prompt is input into the model to generate intermediate outputs. If users are not satisfied and wish to change several bars, they can mark the locations of these unsatisfactory bars. CMI features two novel mechanisms to ensure high-quality infilling. Firstly, we introduce the Joint Embedding Predictive Autoregressive Generative Architecture (JEP-AGA), which learns high-level predictive representations of the masked part (the part that needs to be infilled) during the autoregressive generative process. Unlike traditional autoregressive methods that predict in token space, JEP-AGA mimics how humans perceive and interpret music by understanding the missing part in context and filling it note by note, potentially eliminating unnecessary token-level details and preventing monotonous, repetitive notes. This high-level predictive representation, inspired by LeCun's Joint Embedding Predictive Architecture (JEPA) (LeCun 2022), predicts the representation of the missing part of an input based on the representations of other parts of the same input (Assran et al. 2023), unlike traditional feature extraction models that extract static, independent representations. Secondly, we propose the Dynamic Interaction Learner (DIL) that achieves HITL by enabling fine-tuning the joint embedding predictive structure in JEP-AGA based solely on user interactions. These interactions, which involve selecting and refining parts of the musical output, are treated as direct responses. This approach allows the system to adaptively improve and align with the user's interaction without additional data inputs.

Related Work

Human-in-the-loop (HITL) involves integrating human intelligence into the AI workflow, enabling continuous training and validation of models (Monarch 2021). The interaction between humans and machine learning algorithms has become increasingly vital, not only for enhancing the accuracy of machine learning models and achieving desired outcomes more swiftly, but also for augmenting human efficiency and effectiveness. HITL methodologies have been effectively applied to various tasks, such as text classification (Arous et al. 2021), text summarization (Stiennon et al. 2020), and video object segmentation (Oh et al. 2019). Despite its widespread application, the music generation domain remains a relatively unexplored territory for HITL. Initial efforts in this field have attempted to tackle specific musical attributes such as chord progression (Matsumoto et al. 2022), drum loops (Alain et al. 2020), and melody (Zhou et al. 2020) using HITL approaches. While current works have made some primary research progress by involving humans in achieving single-attribute learning, our work takes a significant step forward by enabling users to control specific music pieces by learning multiple musical attributes, thereby enhancing user interaction experiences. Additionally, our approach provides users with granular control by allowing them to select unsatisfactory parts for optimization instead of providing binary judgments (like or dislike) for the entire generated piece, thus acknowledging the nuanced nature of music evaluation.

Collaborative Music Inpainter

Overview

In this section, we introduce the Collaborative Music Inpainter (CMI), designed to achieve HITL symbolic music infilling. The user provides a marked music piece with unsatisfactory parts to CMI, which then generates a revised version that addresses these issues. CMI continues to learn from each interaction behind the scenes until the user is satisfied with the outcome.

The system infills the unsatisfactory part using the Joint Embedding Predictive Autoregressive Generative Architecture (JEP-AGA). Firstly, the system utilizes a pre-trained context encoder and predictor to understand high-level predictive representations of the masked (unsatisfied) infilling part based on its context. It then uses these high-level predictive representations to infill the masked infilling part, token by token, following the autoregressive generative process. Additionally, the Dynamic Interaction Learner (DIL) uniquely incorporates responses directly through user interactions, whereby the selection of musical segments to refine is automatically considered responses, finetuning the encoder-predictor structure to better align with the user's musical taste by considering both the unsatisfied and neutral parts of the music.

Joint Embedding Predictive Autoregressive Generative Architecture

To alleviate the situation that autoregressive generative models try to fill in every bit of missing information and tend to generate unnecessary token-level details, we propose JEP-AGA, which builds upon an autoregressive generative architecture and a pre-trained encoder-predictor structure to learn high-level predictive representation of the masked infilling part, as illustrated in Figure 2 (a). The theoretical foundation of JEP-AGA lies in the fact that when humans perceive a piece of music, a sentence, or an image, they infer the missing part based on its context and the relationship between its context and the missing part, rather than inferring token by token or pixel by pixel. Additionally, prior work, such as JEPA, has demonstrated its ability to learn high-level predictive

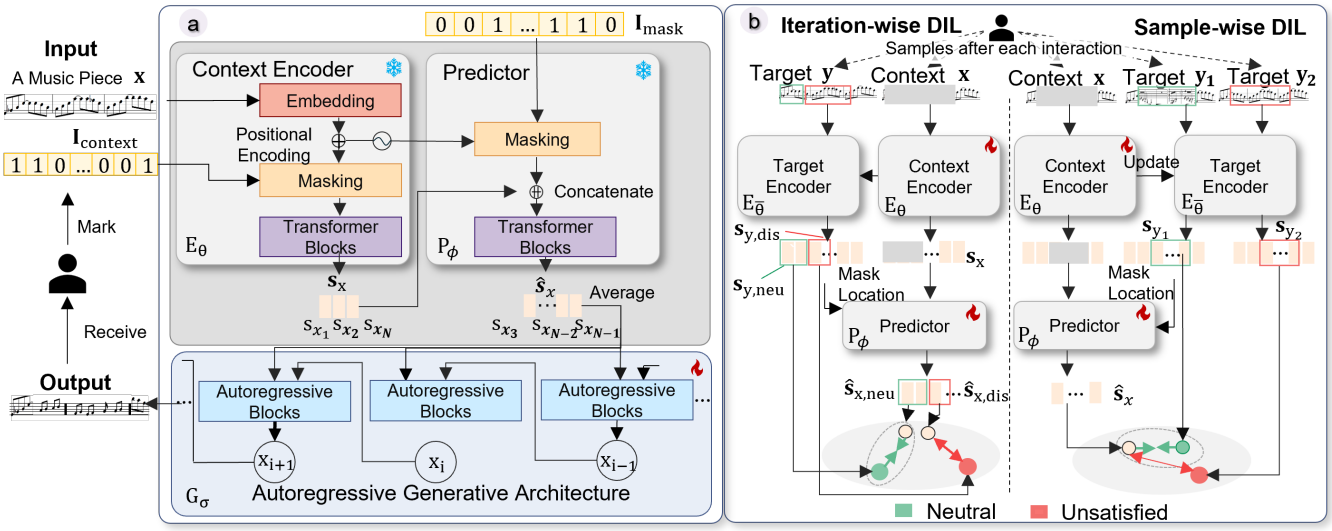

Figure 2: (a) Joint Embedding Predictive Autoregressive Generative Architecture (JEP-AGA): It infills the music based on an autoregressive generative architecture and a context encoder-predictor structure. (b) Dynamic Interaction Learner (DIL): It learns from user interactions through iteration-wise and sample-wise DIL using contrastive learning. The context encoder and predictor predict the representations of the user's unsatisfied or neutral part, and the target encoder learns the representations of the same location from the target/ground truth piece. Fire: trainable parameters. Snowflake: frozen parameters.

representation of masked tokens based on context (Assran et al. 2023). This high-level predictive information assists the generative model in filling blanks of the input music while preventing it from generating meaningless repetitions and unclear music structures.

Specifically, JEP-AGA includes a pre-trained context encoder $E_\theta(\cdot)$, a pre-trained predictor $P_\phi(\cdot,\cdot)$, and an autoregessive generation model $G_\sigma(\cdot)$. The context encoder is tasked with extracting high-level context representation (surrounding information that is not masked by the user) from the musical sequence. The predictor predicts high-level predictive representations of the missing part by interpreting context representations alongside the user's unsatisfied location information. In the generative architecture pipeline, the model operates in an autoregressive manner, predicting the token at time t based on previous information.

Context Encoder The context encoder is designed to learn high-level context representation of the user's unsatisfied part. Given the original music $\mathbf{x} = \{x_1, x_2, \cdots, x_N\}$, where N is the length of the music, and the context indicator $\mathbf{I}_{\text{context}} = \{0,1\}^N$, where 0 indicates the user's unsatisfied part that has been masked and 1 indicates the unmasked part, the context encoder $E_\theta(\cdot)$ outputs the context representation: $\mathbf{s}_x = \{s_{x_j} \mid j \in \mathcal{J}\}, \quad \mathcal{J} = \{j : \mathbf{I}_{\text{context}}[j] = 1\}$. Specifically, the masking module in the context encoder operates to selectively extract specific patches from $\mathbf{x}$ where the indicator is set to 1, ensuring that only context representations not masked by the user are chosen.

Predictor The predictor $P_\phi(\cdot,\cdot)$, receiving inputs $\mathbf{s}_x$ from the context encoder, further utilizes the positional information of the user's unsatisfied part to predict its final representation, where $\mathbf{I}_{\text{mask}}$ is denoted as: $\mathbf{I}_{\text{mask}}[j] = 1 - \mathbf{I}_{\text{context}}[j], \quad \forall j \in \{1, \ldots, N\}$. Thus, $P_\phi(\cdot,\cdot)$ takes the context representation $\mathbf{s}_x$ and the user unsatisfied tokens location $\mathbf{I}_{\text{mask}}$, and outputs the predicted representation $\hat{\mathbf{s}}_x = \{\hat{s}_{x_j} \mid j \in \mathcal{J}\}, \quad \mathcal{J} = \{j : \mathbf{I}_{\text{mask}}[j] = 1\}$.

Generator In the autoregressive generative model, we construct the sequence for symbolic music infilling tasks based on ILM (Donahue, Lee, and Liang 2020): $\{x_1, x_2, \text{<M>}, x_N, \text{<S>}, x_3, \cdots, x_{N-1}, \text{<A>}\}$, where <M> represents the mask token, <S> represents the separation token, <A> represents the answer token, and N is the number of tokens. The autoregressive generative model $G_\sigma(\cdot)$ receives the inputs from the embedding from time 1 to $t-1$ and the predictor representations from encoder-predictor structure by concatenating them at each time t: $[\text{Embed}(x_i), \hat{\mathbf{s}}_{x,\text{norm}}]$, where $[.,.]$ denotes the concatenation operation and $\hat{\mathbf{s}}_{x,\text{norm}}$ is defined as $\dfrac{\hat{\mathbf{s}}_x}{\sum_{j=1}^{N} \mathbf{I}_{\text{mask}}[j]}$. The training objective of the generative architecture can be denoted as:

$$-\sum_{0<t\leq N} \log p(x_t \mid x_{t-1}, \ldots, x_2, x_1, \hat{\mathbf{s}}_{x,\text{norm}}) \qquad (1)$$

During training, the goal is to learn σ^* that optimizes the objective based on the input data and the pre-trained E_θ and P_ϕ. During generation, the goal is to generate the infilled part $\{x_3, \cdots, x_{N-1}, \text{<A>}\}$ given the condition $\{x_1, x_2, \text{<M>}, x_N, \text{<S>}\}$ and the pre-trained E_θ and P_ϕ.

Dynamic Interaction Learner

To ensure the HITL process, the context encoder and predictor structure in JEP-AGA should learn personalized high-level predictive representations instead of a universal pre-

trained structure. Each user interaction, which involves choosing parts of the music to adjust, directly informs the fine-tuning of the context encoder and predictor. This process is illustrated in Figure 2 (b).

Through both iteration-wise and sample-wise contrastive learning, DIL minimizes discrepancies between the user's expectations and the system's outputs by adjusting its predictions via parameter updating, with adjustments occurring iteratively or upon achieving satisfactory outcomes in each example.

Iteration-wise DIL Iteration-wise DIL ensures that the model's infilling results differ from the user's previous unsatisfactory results. This step is essential to inform the model of what the user does not like and to avoid generating similar unsatisfactory results.

To begin with, the user indicates the unsatisfied part, which we categorize as the unsatisfied part $\mathbf{I}_{\text{dis}}$ (shown in a red rectangle). Then, the user-neutral part is randomly marked as a counterpart of the unsatisfied part and designated as the neutral part $\mathbf{I}_{\text{neu}}$ (shown in a green rectangle). The input without any grey masks is denoted as the target or ground truth $\mathbf{y}$, the same input but with the user's unsatisfied and neutral parts masked is denoted as context $\mathbf{x}$. The representation of the neutral and unsatisfied part after the target encoder $E_{\bar{\theta}}(\cdot)$ is denoted by $\mathbf{s}_{y,\text{neu}} = \{s_{y_j} \mid j \in \mathcal{J}\}, \quad \mathcal{J} = \{j : \mathbf{I}_{\text{neu}}[j] = 1\}$, and $\mathbf{s}_{y,\text{dis}} = \{s_{y_j} \mid j \in \mathcal{J}\}, \quad \mathcal{J} = \{j : \mathbf{I}_{\text{dis}}[j] = 1\}$. The predictive representation for the neutral and unsatisfied parts after context encoder and predictor is then $\hat{\mathbf{s}}_{x,\text{neu}}$ and $\hat{\mathbf{s}}_{x,\text{dis}}$, respectively.

We formulate the objective to maximize the similarity between positive pairs $(\hat{\mathbf{s}}_{x,\text{neu}}, \mathbf{s}_{y,\text{neu}})$ and minimize the similarity between negative pairs $(\hat{\mathbf{s}}_{x,\text{dis}}, \mathbf{s}_{y,\text{dis}})$. The negative pair similarity for iteration-wise learning, denoted as Neg_{ite}, is calculated as $Neg_{\text{ite}} = Sim(\mathbf{s}_{y,\text{dis}}, \hat{\mathbf{s}}_{x,\text{dis}})$. $Sim(a, b)$ is the cosine similarity between two features: $\dfrac{a \cdot b}{|a||b|}$, where $a \cdot b$ is the dot product of a and b, $|a|$ and $|b|$ are the norms of a and b respectively.

The positive pair similarity for iteration-wise learning, denoted as Pos_{ite}, is calculated as $Pos_{\text{ite}} = Sim(\mathbf{s}_{y,\text{neu}}, \hat{\mathbf{s}}_{x,\text{neu}})$. The loss for iteration-wise learning is defined as:

$$L_{\text{ite}} = \max(0, Neg_{\text{ite}} - Pos_{\text{ite}} + \alpha). \tag{2}$$

Here, α is a predefined margin designed to ensure that the positive similarity is greater than the negative similarity by at least this margin.

Target Encoder We obtain the output representations of the target encoder, rather than the input, for further similarity and difference comparison to ensure that the target representations are at a high semantic level (Assran et al. 2023). The target encoder has the same structure as the context encoder, with the only difference being that the target encoder processes the complete music, while the context encoder processes only the context. The input to the target encoder can be a piece from the last iteration, a piece satisfying user feedback, or a piece reflecting user dissatisfaction. In iteration-wise DIL, the input to the target encoder is a piece reflecting user dissatisfaction. For a given target music piece $\{y_1, y_2, \cdots, y_N\}$, the target encoder produces the representation $\mathbf{s}_y = \{s_{y_1}, s_{y_2}, \cdots, s_{y_N}\}$, where s_{y_i} represents the representation of the i-th token.

Sample-wise DIL Sample-wise DIL ensures that the model learns the complete interaction process for one example, including how the final result satisfies the user while the previous result did not.

To begin with, the final satisfied version of the music is considered as the target sample $\mathbf{y}_1$, and the first iteration version as the target sample $\mathbf{y}_2$. In both cases, the unsatisfied part is fixed as $\mathbf{I}_{\text{mask}}$. The context $\mathbf{x}$ is the music excluding the unsatisfied part from the first interaction. Then, the representations of $\mathbf{y}_1$, $\mathbf{y}_2$, and the predictive representation given $\mathbf{x}$ are denoted as $\mathbf{s}_{\mathbf{y}_1}$, $\mathbf{s}_{\mathbf{y}_2}$, and $\hat{\mathbf{s}}_x$, respectively. The negative similarity Neg_{sam} is calculated as $Neg_{\text{sam}} = Sim(\mathbf{s}_{\mathbf{y}_2}, \hat{\mathbf{s}}_x)$. The positive similarity Pos_{sam} is calculated as $Pos_{\text{sam}} = Sim(\mathbf{s}_{\mathbf{y}_1}, \hat{\mathbf{s}}_x)$. The model is fine-tuned based on the contrastive loss:

$$L_{\text{sam}} = \max(0, Neg_{\text{sam}} - Pos_{\text{sam}} + \alpha). \tag{3}$$

The parameters of the predictor and context encoder are learned through gradient-based optimization, while the parameters of the target encoder are updated via an exponential moving average of the context encoder parameters.

Experiments

Dataset and Implementation Details

Dataset We utilize two datasets for training: Bread[2] (Peng et al. 2023) and LMD (Raffel 2016). The Bread dataset, containing 851,313 polyphonic MIDI files, stands as one of the largest MIDI datasets currently accessible online. It encompasses a total of approximately 7.89 billion tokens after converting MIDI files into sequences. Additionally, the LMD dataset contributes an extra 176,581 polyphonic MIDI files, containing about 1.89 billion tokens. We selected these two datasets as they are the largest collections available, featuring diverse musical styles, instruments, and tracks, essential for enhancing our model's performance and ensuring its capability to manage various musical complexities. Their extensive use and validation in music generation and infilling tasks (Peng et al. 2023; Ens and Pasquier 2020; Mittal et al. 2021) have proven their reliability and effectiveness for such applications. We also conducted a case study using the Pop909 dataset (Wang et al. 2020) to demonstrate the robustness of our model, a standard pop piano benchmark in the music infilling domain, showcasing the model's effectiveness in practical scenarios.

Implementation Details For data representation, we convert symbolic MIDI data into a sequence of tokens following the method used in RWKV Music[3]. This data representation includes three main types of tokens: wait tokens for timing, which encompass 125 different values; combined note, velocity, and instrument tokens totaling 19,968 tokens (128 notes * 12 quantized velocities * 13 binned instruments);

[2]https://huggingface.co/BlinkDL/rwkv-4-music

[3]https://github.com/briansemrau/MIDI-LLM-tokenizer

Model	Bread						LMD					
	$PD_{\text{p}} \uparrow$	$PD_{\text{f}} \uparrow$	$DD_{\text{p}} \uparrow$	$DD_{\text{f}} \uparrow$	$CD_{\text{p}} \uparrow$	$CD_{\text{f}} \uparrow$	$PD_{\text{p}} \uparrow$	$PD_{\text{f}} \uparrow$	$DD_{\text{p}} \uparrow$	$DD_{\text{f}} \uparrow$	$CD_{\text{p}} \uparrow$	$CD_{\text{f}} \uparrow$
XLNet (Chang, Lee, and Yang 2021)	0.67	0.73	0.55	0.71	0.68	0.77	0.68	0.76	0.64	0.78	0.69	0.84
Transformer (Malandro 2023)	0.71	0.73	0.68	0.68	0.75	0.77	0.74	0.77	0.68	0.67	0.76	0.74
RWKV (Peng et al. 2023)	0.75	0.74	0.68	0.71	0.76	0.82	0.73	0.76	0.72	0.70	0.78	0.78
CMI	**0.83**†	**0.78***	**0.80**†	**0.81**†	**0.83**†	**0.88**†	**0.83**†	**0.83**†	**0.85**†	**0.85**†	**0.85**†	**0.88**†

Table 1: Objective evaluation comparison between our proposed model and other models across Bread and LMD datasets in terms of rhythm, melody, and harmony consistency between the past and future content on the music infilling task. † and * indicate statistically significant improvements over RWKV at the 0.05 and 0.1 significance levels.

and 6 instruction tokens: pad, start, end, mask, answer, separation tokens, summing up to a total of 20,099 different tokens. Each sequence of music is set to a length of 4096 tokens. For the masking strategy, we randomly mask tokens out of the 4096, with the ratio varying between 0.1 and 0.8 for each sample. For model training, we use the RWKV-4 560M model as our generative architecture backbone on 4 GPUs since it demonstrates good performance on music generation tasks based on the Bread dataset. In addition, we set the hyperparameter α to 0.1. Each model is trained with an initial learning rate of 1e-6 and a batch size of 4 for 100 epochs, following the guidelines from RWKV Music. The context encoder is a Transformer-based model with a depth of 6 and an encoder embedding size of 512. The predictor is a Transformer-based model with a depth of 3 and an encoder embedding size of 256, which has shown effectiveness in experiments detailed in the Appendix. Training JEP-AGA follows the typical autoregressive generative model, where the model predicts the next token based on past information. The difference in JEP-AGA is that it further concatenates information from the pre-trained context encoder and predictor. The training objective of the generative architecture follows equation 1.

Model	M ↑	N ↑	F ↑	C ↑	DR ↑
XLNet	2.98	2.78	2.56	2.56	0.19
Transformer	3.15	3.12	2.71	2.93	0.25
RWKV	3.27	3.24	2.78	3.00	0.34
CMI	**3.73**†	**3.66**†	**3.10**†	**3.59**†	**0.49**†

Table 2: Subjective evaluation comparison between our proposed model and other models in terms of Deception Rate (DR), Musicality (M), Naturalness (N), Fitness (F), and Creativity (C) on music infilling task. † indicates statistically significant improvements over RWKV at the 0.05 significance level.

Evaluation Metrics

Objective Metrics We evaluate model performance from two perspectives: the quality of generated results and user responses in the HITL process. For the quality of generated results, we assess how well the infilled part aligns with its past (p) and future (f) content using the average overlapped area of pitch distribution (PD), duration distribution (DD), and chord distribution (CD). These metrics, widely recognized in music composition for similarity comparisons (Hu et al. 2024; Dai et al. 2023; Min et al. 2023; Chang, Lee, and Yang 2021), yield six measures: PD_{p}, PD_{f}, DD_{p}, DD_{f}, CD_{p}, and CD_{f}.

For user responses, we focus on two perspectives: the increase in user satisfaction (IUS) and the reduction in interaction cost (RIC). IUS is calculated as $IUS = (SS_{\text{proposed}} - SS_{\text{base}})/SS_{\text{base}}$, and RIC is calculated as $RIC = (IC_{\text{proposed}} - IC_{\text{base}})/IC_{\text{base}}$, where SS is the Satisfaction Score reflecting the similarity between generated results and the user's satisfactory content, and IC denotes the Interaction Cost, indicating the minimum number of interaction rounds needed for the user to reach a satisfactory SS score γ, and the baseline is CMI without applying the user response mechanism. A higher positive IUS indicates improved user satisfaction, while a larger negative RIC reflects reduced interaction cost by the proposed model.

To simulate user interaction, we randomly mask a portion of the music and keep it constant across iterations to represent the unsatisfied section. Then, we randomly mark a segment of equal length from the remaining part as the satisfied section. At each iteration i, SS is calculated as follows:

$$SS_i = \frac{Diff(f_{\text{neu},0}, f_{\text{dis},0}) - Diff(f_{\text{neu},i}, f_{\text{dis},i})}{\max(Diff(f_{\text{neu},0}, f_{\text{dis},0}), Diff(f_{\text{neu},i}, f_{\text{dis},i}))}, \quad (4)$$

$Diff(a, b)$ calculates the pairwise difference between two vectors a and b: $||a - b||_2$, which measures the Euclidean distance. The terms $f_{\text{neu},i}$ and $f_{\text{dis},i}$ denote the user's neutral and unsatisfied parts at the i-th iteration, as determined by the target encoder. The scale of SS_i ranges from -1 to 1, indicating the percentage of satisfactory increase (> 0) or decrease (< 0) at each iteration. The interaction cost IC is calculated as follows:

$$IC = \min\{i \mid SS_i \geq \gamma\}. \quad (5)$$

We set a threshold $\gamma = 10\%$ for SS_i to determine the number of interaction rounds required for the model to meet the threshold.

Subjective Metrics and Participants In addition to the objective metrics, we also conduct a subjective evaluation to ensure the quality of the generated results. We collected 41 listener reports by distributing a survey on social media. Among the 41 participants, 16 were men and 25 were women;

7 were under age 20, 22 were between 20 and 30, 8 were between 31 and 40, 4 were between 41 and 50, and 1 was older than 50. In addition, 12 had musical training for more than 5 years and were familiar with music theory; the others had no or less than 5 years of musical training. All participants in the subjective experiment were informed of the study's purpose and provided their consent for the use of their data in our research.

The listener is asked to listen to a piece of music where some parts are composed by the model. First, participants must identify which sections of the music were composed by the model. We calculate the Deception Rate (DR) by determining the overlap between the actual machine-infilled length and the participant's guessed length. We then average these results and subtract from 1. This metric aims to assess the perceptibility of the infilled music sections to listeners. A high DR indicates that the machine-generated parts convincingly mimic human composition, suggesting effective blending, while a low DR points to the need for more natural-sounding infilling. Next, the machine-composed part will be revealed, and the participant will evaluate the machine-composed parts in terms of Musicality (M), Naturalness (N), Fitness (F), and Creativity (C) on a scale from 1 (lowest) to 5 (highest) (Min et al. 2023).

Comparison with SOTA Models

To validate the music infilling performance of our proposed model, we conducted comparisons with several SOTA music infilling works. It is important to note that comparing our model with those based on diffusion or VAE requires a sophisticated redesign of the entire music infilling pipeline, including data representation, training, and inference. For a fair comparison, we selected autoregressive-based SOTA models specifically designed for music infilling: an XLNet-based model (Chang, Lee, and Yang 2021), a Transformer-based model (Malandro 2023), and an RWKV-based model (Peng et al. 2023). All the compared models are trained and tested on the same data with the same music representation methods.

Tables 1 and 2 present a comparison of our proposed model's performance with that of other models in both objective and subjective experiments. The objective experimental results demonstrate that our model can infill music with a more similar distribution with previous and future content in rhythm, melody, and harmony compared to other models across both datasets. Regarding the subjective experimental results, the higher scores across musicality, naturalness, fitness, and creativity underscore the effectiveness of our approach compared to other models. Additionally, the higher deception rate (DR) suggests that the machine-composed music can deceive the human ear more effectively than other models can, although it cannot fool human ears in nearly half of the cases.

Ablation Analysis

Ablating JEP-AGA Tables 3 and 4 present the model performance with and without JEP-AGA in both objective and subjective experiments. Both experiments demonstrate superior performance when applying JEP-AGA. This improvement stems from the high-level predictive representation provided by the encoder-predictor structure. The predictor in JEP-AGA is capable of modeling spatial uncertainty in the music sequence from a partially observable context, thereby enhancing the model's ability to generate notes that are contextually similar.

Model	$PD_{\text{p}} \uparrow$	$PD_{\text{f}} \uparrow$	$DD_{\text{p}} \uparrow$	$DD_{\text{f}} \uparrow$	$CD_{\text{p}} \uparrow$	$CD_{\text{f}} \uparrow$
w/o JEP-AGA	0.74	0.74	0.70	0.71	0.77	0.80
w/ JEP-AGA	**0.83**†	**0.81**†	**0.83**†	**0.83**†	**0.84**†	**0.88**†

Table 3: Ablating the JEP-AGA: objective evaluation comparison. † indicates statistically significant improvements over w/o JEP-AGA at the 0.05 significance level.

Model	M ↑	N ↑	F ↑	C ↑	DR ↑
w/o JEP-AGA	3.27	3.24	2.78	3.00	0.34
w/ JEP-AGA	**3.68**†	**3.63**†	**3.15**†	**3.34**†	**0.45**†

Table 4: Ablating the JEP-AGA: subjective evaluation comparison.

Model	M ↑	N ↑	F ↑	C ↑	DR ↑
w/o DIL	3.68	3.63	**3.15**	3.34	0.45
w/ DIL	**3.73**	**3.66**	3.10	**3.59**	**0.49**

Table 5: Ablating the DIL: subjective evaluation comparison.

Ablating DIL Figure 3 illustrates the increase in user satisfaction (IUS) and the reduction of interaction cost (RIC) compared to CMI without applying DIL over 20 different tests. Each test involves 5 different music pieces, with each piece undergoing a maximum of 10 rounds of interaction; the values represent the average score across 50 interactions. The x-axis indicates the test index, and the y-axis shows the average percentage increase or decrease in IUS and RIC. The results show a positive IUS score across all samples, indicating that the proposed model effectively learns from user feedback during the interaction process, making the generated music more aligned with the user's preferences. Furthermore, the data reveals that the proposed model significantly reduces human effort in the HITL process.

Table 5 displays the subjective experiment results with and without interaction. The scores for musicality, naturalness, and creativity increase after the interaction occurs, while the fitness score remains at a high level. The decrease in fitness score is because, without interaction, CMI learns the structure that fits the content as closely as possible. Learning from user interactions, however, increases the possibility of generating creative content, which can introduce surprises in terms of melody, instrument, rhythm, and more. While this creativity can enhance the music, it might also cause participants to feel a subtle sense of inconsistency.

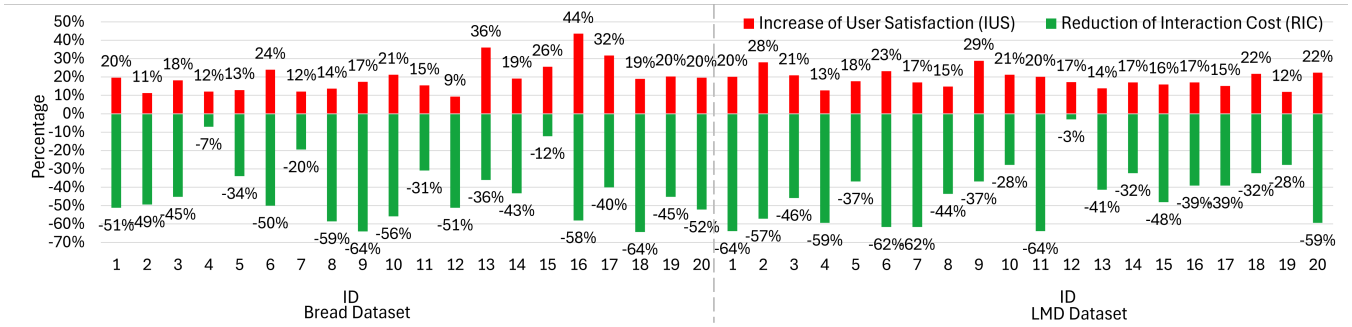

Figure 3: Ablating the DIL: objective evaluation results for 20 random tests, comparing Increase of User Satisfaction (IUS) and Reduction of Interaction Cost (RIC) between the proposed model and CMI without applying DIL. Blue bars represent the average IUS per HITL interaction, while orange bars show the average RIC per interaction. Overall, our proposed model consistently improves user satisfaction and reduces interaction costs across various user requests.

	Bread		LMD	
Strategy	IUS ↑	RIC ↓	IUS ↑	RIC ↓
Iteration-wise	+6.5%	-24.6%	+13.2%	-20.9%
Sample-wise	+12.2%	-34.0%	+11.7%	-35.8%
Both	**+20.1%**	**-43.7%**	**+18.5%**	**-44.1%**

Table 6: Ablating the learning strategy. CMI is beneficial for learning user responses, with adopting both strategies achieving the highest performance.

Ablating Learning Strategy in DIL Table 6 presents the IUS and RIC results of employing various learning strategies on two datasets. The data indicate that utilizing both iteration-wise and sample-wise DIL enhances user satisfaction and reduces interaction costs, with the combined application of both strategies yielding superior performance. This suggests that CMI can enhance interaction experiences with the HITL approach, potentially paving the way for the development of practical music infilling tools in the future.

Case Study

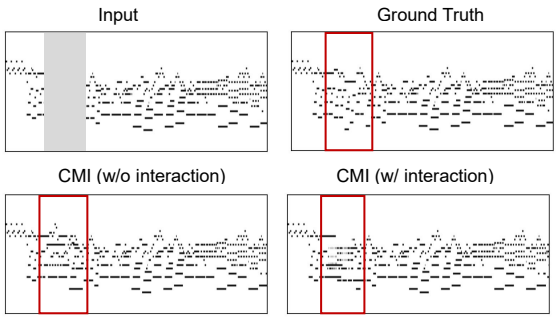

Figure 4: CMI can infill the music with high quality. Moreover, CMI avoids generating similar content to the user's unsatisfactory part.

Figure 4 displays a pop piano piece from the Pop909 dataset, infilled by CMI. The original version was acceptable by the user but criticized for lacking novelty, especially in the middle two phrases. Upon user request, CMI restructured these phrases, enhancing creativity. Without the DIL mechanism, CMI successfully infills two new, cohesive phrases, smoothly transitioning from the existing context to the generated content and seamlessly returning to the fourth phrase. With the DIL mechanism, CMI leverages insights from other phrases of the music to avoid generating similar structures to the user's unsatisfactory part. It consolidates the two short phrases into one longer phrase, imbuing it with a sense of progression while maintaining smooth transitions throughout, thereby validating the improved performance of CMI. We provide more audio samples in the supplementary materials.

Conclusion

In conclusion, our work introduces the Collaborative Music Inpainter (CMI), a novel HITL paradigm to music infilling. The Joint Embedding Predictive Autoregressive Generative Architecture (JEP-AGA) enhances music generation by learning high-level predictive representations during the generative process. The Dynamic Interaction Learner (DIL) allows users to iteratively refine music segments, effectively merging human creativity with machine precision. Trained extensively on large MIDI datasets and capable of generating sequences up to 4096 tokens, the CMI demonstrates robustness and suitability for complex musical compositions. Empirical results confirm CMI's superior performance in music infilling, highlighting its efficiency in producing high-quality music with minimal human intervention. This not only streamlines the music creation process but also fosters new avenues for creative collaboration between humans and AI in music composition. We aim to expand the CMI approach to non-autoregressive-based music infilling tasks in the future, such as VAE-based and diffusion-based models, which have shown promising performance in music generation. Additionally, we plan to develop a user interaction interface for CMI. We believe that CMI will inspire further research in AI-assisted music generation, leading to more interactive tools for composers and musicians worldwide.

Acknowledgements

This work is supported by The Hong Kong Polytechnic University for the project MusicGPT: Revolutionizing the Soundscape with AI-Powered Interactive Music Creation (P0048382), and the National Natural Science Foundation of China (72350710798).

References

Alain, G.; Chevalier-Boisvert, M.; Osterrath, F.; and Piche-Taillefer, R. 2020. Deepdrummer: Generating drum loops using deep learning and a human in the loop. *arXiv preprint arXiv:2008.04391*.

Arous, I.; Dolamic, L.; Yang, J.; Bhardwaj, A.; Cuccu, G.; and Cudré-Mauroux, P. 2021. Marta: Leveraging human rationales for explainable text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 5868–5876.

Assran, M.; Duval, Q.; Misra, I.; Bojanowski, P.; Vincent, P.; Rabbat, M.; LeCun, Y.; and Ballas, N. 2023. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15619–15629.

Chang, C.-J.; Lee, C.-Y.; and Yang, Y.-H. 2021. Variable-length music score infilling via XLNet and musically specialized positional encoding. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 97–104.

Chen, K.; Wang, C.-i.; Berg-Kirkpatrick, T.; and Dubnov, S. 2020. Music sketchnet: Controllable music generation via factorized representations of pitch and rhythm. In *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, 77–84.

Copet, J.; Kreuk, F.; Gat, I.; Remez, T.; Kant, D.; Synnaeve, G.; Adi, Y.; and Défossez, A. 2024. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36.

Dai, S.; Ma, X.; Wang, Y.; and Dannenberg, R. B. 2023. Personalised popular music generation using imitation and structure. *Journal of New Music Research*, 1–17.

Donahue, C.; Lee, M.; and Liang, P. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2492–2501. Association for Computational Linguistics.

Ens, J.; and Pasquier, P. 2020. Mmm: Exploring conditional multi-track music generation with the transformer. *arXiv preprint arXiv:2008.06048*.

Hadjeres, G.; Pachet, F.; and Nielsen, F. 2017. Deepbach: a steerable model for bach chorales generation. In *International conference on machine learning*, 1362–1371. PMLR.

Hu, Z.; Liu, Y.; Chen, G.; Ma, X.; Zhong, S.; and Luo, Q. 2024. Responding to the Call: Exploring Automatic Music Composition Using a Knowledge-Enhanced Model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 521–529.

Hu, Z.; Ma, X.; Liu, Y.; Chen, G.; and Liu, Y. 2022. The Beauty of Repetition in Machine Composition Scenarios. In *Proceedings of the 30th ACM International Conference on Multimedia*, 1223–1231.

Hu, Z.; Ma, X.; Liu, Y.; Chen, G.; Liu, Y.; and Dannenberg, R. B. 2023. The beauty of repetition: an algorithmic composition model with motif-level repetition generator and outline-to-music generator in symbolic music generation. *IEEE Transactions on Multimedia*.

Huang, C.-Z. A.; Cooijmans, T.; Roberts, A.; Courville, A.; and Eck, D. 2017. Counterpoint by convolution. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, 211–218.

Ippolito, D.; Huang, A.; Hawthorne, C.; and Eck, D. 2018. Infilling piano performances. In *NIPS Workshop on Machine Learning for Creativity and Design*, volume 2.

Ji, S.; and Yang, X. 2024. MusER: Musical Element-Based Regularization for Generating Symbolic Music with Emotion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12821–12829.

Lam, M. W.; Tian, Q.; Li, T.; Yin, Z.; Feng, S.; Tu, M.; Ji, Y.; Xia, R.; Ma, M.; Song, X.; et al. 2024. Efficient neural music generation. *Advances in Neural Information Processing Systems*, 36.

LeCun, Y. 2022. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1).

Lin, L.; Xia, G.; Zhang, Y.; and Jiang, J. 2024. Arrange, inpaint, and refine: Steerable long-term music audio generation and editing via content-based controls. *arXiv preprint arXiv:2402.09508*.

Malandro, M. E. 2023. Composer's Assistant: An Interactive Transformer for Multi-Track MIDI Infilling. *arXiv preprint arXiv:2301.12525*.

Matsumoto, Y.; Ito, H.; Terasawa, H.; Yamamoto, Y.; Hiraga, Y.; and Matsubara, M. 2022. Human-In-The-Loop Chord Progression Generator With Generative Adversarial Network. In *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 612–618. IEEE.

Min, L.; Jiang, J.; Xia, G.; and Zhao, J. 2023. Polyffusion: A diffusion model for polyphonic score generation with internal and external controls. In *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023, Milan, Italy, November 5-9, 2023*, 231–238.

Mittal, G.; Engel, J. H.; Hawthorne, C.; and Simon, I. 2021. Symbolic Music Generation with Diffusion Models. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 468–475.

Monarch, R. M. 2021. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster.

Oh, S. W.; Lee, J.-Y.; Xu, N.; and Kim, S. J. 2019. Fast user-guided video object segmentation by interaction-and-propagation networks. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, 5247–5256.

Peng, B.; Alcaide, E.; Anthony, Q.; Albalak, A.; Arcadinho, S.; Cao, H.; Cheng, X.; Chung, M.; Grella, M.; GV, K. K.; et al. 2023. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*.

Raffel, C. 2016. *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching. 331 Ph. D.* Ph.D. thesis, thesis, Columbia University.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021.

Wang, Z.; Chen, K.; Jiang, J.; Zhang, Y.; Xu, M.; Dai, S.; Gu, X.; and Xia, G. 2020. Pop909: A pop-song dataset for music arrangement generation. *arXiv preprint arXiv:2008.07142*.

Wei, S.; Xia, G.; Zhang, Y.; Lin, L.; and Gao, W. 2022. Music phrase inpainting using long-term representation and contrastive loss. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 186–190. IEEE.

Yu, B.; Lu, P.; Wang, R.; Hu, W.; Tan, X.; Ye, W.; Zhang, S.; Qin, T.; and Liu, T.-Y. 2022. Museformer: Transformer with fine-and coarse-grained attention for music generation. *Advances in Neural Information Processing Systems*, 35: 1376–1388.

Zhou, Y.; Koyama, Y.; Goto, M.; and Igarashi, T. 2020. Generative melody composition with human-in-the-loop Bayesian optimization. In *Proceedings of the 2020 Joint Conference on AI Music Creativity (CSMC-MuMe 2020)*.