Designing Live Human-AI Collaboration for Musical Improvisation

NIC BECKER, Stanford University, USA
RYAN LOUIE, Stanford University, USA
JOHN THICKSTUN, Stanford University, USA
PERCY LIANG, Stanford University, USA

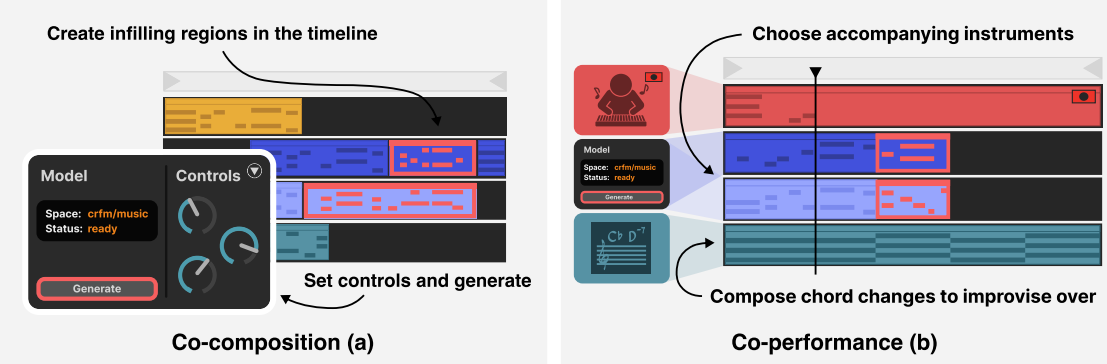

Fig. 1. We present two workflows for interacting with music AI natively in a digital audio workstation. Figure 1(a) depicts an offline composition environment where musicians collaboratively compose with AI which generates notes in empty clips based on the surrounding composition. Figure 1(b) illustrates a live improvisation setting where musicians perform alongside AI-generated accompaniment, dynamically crafted in response to their play and an underlying chord progression.

Recent progress in generative models for music has highlighted the need for interactive, controllable systems that service the goals of musicians. In this paper, we introduce a system that integrates a compositional assistant and live improvisational partner directly into the modern music producer's toolkit–the digital audio workstation. Our system design is guided by 1) integration with modern music production software, 2) non-linear songwriting workflows, and 3) enabling generative AI to support live improvisation tasks. We find that anticipatory transformer models are well suited for these goals, and present a method for adapting an anticipatory model for live improvisation. We call on future work to further explore human-AI co-performance by designing systems to be accessible and integrated into the workflows of domain experts.

CCS Concepts: • **Human-centered computing** → **Interactive systems and tools**; • **Applied computing** → **Sound and music computing**.

Additional Key Words and Phrases: Generative AI, Human-AI Collaboration, Musical Improvisation

1 INTRODUCTION

Generative AI promises a powerful new set of tools for creative work, yet many questions remain about how best to integrate AI into artists' creative practice. We are beginning to see an effort to integrate image and video generation models into familiar software environments for artists [2, 21]. In contrast, music production software has been slow to integrate generative models, despite significant improvements in the quality of AI-generated music [4, 9, 23] and demonstrations of how generative AI could power new tools for music composition [5, 10, 12, 17, 20]. This discrepancy motivates an investigation of how generative AI could be better integrated into musicians' workflows.

We identify three design goals for music co-creation tools that may increase their potential adoption and usefulness for practicing musicians. First, building generative AI tools as plug-ins for established ecosystems, rather than as standalone software, enables practitioners to adapt the tools to their practice rather than adjusting their practice to fit the constraints of the tool [15]. Second, tools should support a non-linear compositional process where musicians have the flexibility to work on different parts of a song in any order, rather than enforcing a linear, start-to-finish creative process. Finally, while recent efforts in generative AI for music co-creation have been situated in tools for offline composition, reimagining generative AI as a dynamic partner in real-time improvisation opens doors into less explored forms of music co-creation.

Guided by these design goals, we present a system for leveraging generative AI for both offline compositional tasks and real-time improvisation within a software environment central to modern music production: the digital audio workstation (DAW). First, we illustrate two workflows which enable musicians to (1) co-compose in a non-linear manner with AI and (2) improvise over chord changes alongside an AI partner. Second, we detail the technical challenges of this improvisation task, and describe how integrating models that can “look ahead" into the DAW can meet the demands of the task. Lastly, we discuss the role domain experts can play in building better generative tools and call for future work to explore human-AI co-performance in other domains.

2 BACKGROUND

In this work, we consider interaction with generative models for symbolic music, where music is represented as a sequence of discrete tokens that is roughly analogous to a musical score. The task of creating an accompaniment to a melody or filling in missing parts of a musical piece are examples of a more general infilling task where full musical sequences are realized given partial observation of a subset of notes. Prior work on infilling [10, 13, 18] has powered demonstrations [13, 16, 22] of interactive musical experiences that support composition in a linear, left-to-right fashion with strict restrictions on genre, meter, or number of instruments. Recently, there has also been effort to develop plug-ins that integrate multi-track infilling tools into a DAW [17, 22]. In these cases, infilling is utilized as a compositional tool for offline music co-creation with AI. Prior work on human-AI co-creation systems that support “simultaneous play" include a proposal for using reinforcement learning for online accompaniment generation [14], and *BachDuet* [6], which generates monophonic counterpoint to monophonic human input. To our knowledge, there has not been a demonstration of human-AI co-creation in an improvisation setting that does not place strict restrictions on genre, instruments, and polyphony.

With the aim of using multi-track infilling models to power human-AI co-performance, we build a system for embedding generative models for symbolic music directly in the DAW without placing restrictions on genre, meter, or instruments generated. We target two applications: a compositional assistant that aids in non-linear multi-track infilling and a live improvisation tool that accompanies human input in real-time.

3 DESIGNING INTERFACES FOR INTERACTION WITH MUSIC AI

Musical improvisation is the creative process of spontaneously composing and performing music in the moment. To coordinate their playing, musicians often improvise within some kind of framework that is agreed upon beforehand, whether that be a chord progression, melody, or rhythmic structure. Drawing from the jazz tradition, where musicians use "lead sheets" to establish a harmonic framework to improvise within, we envision a scenario where a generative model and a human musician engage in a musical dialogue. This dialogue is structured around a pre-established chord progression, allowing both parties to contribute creatively within a common framework; see Figure 1.

In Section 3.1, we propose a system for human-AI live improvisation in addition to co-composition that integrates into a digital audio workstation. Then, in Section 3.2 we summarize several technical challenges presented by live improvisation and how anticipatory transformer models [23] can be adapted to meet these challenges.

3.1 Human-AI Music Co-Creation within a Digital Audio Workstation

3.1.1 DAW integration. We develop a MIDI plug-in for Ableton Live [3] as a general interface for embedding symbolic music models directly into the DAW[1]. Our plug-in allows users to use generative tools for non-linear multi-track infilling and live human-AI improvisation natively within their existing production workflow. We design the plug-in for doing inference remotely, with support for models hosted on Hugging Face Spaces [11] or elsewhere with Gradio [1].

3.1.2 Composing with music AI. As an example of how generative AI can be useful for musicians, consider a songwriter composing in the DAW with a digital keyboard to record performances. Figure 1(a) shows a scenario where the songwriter has composed parts of several tracks in the arrangement and assigned a program number to indicate the instrument of each track. The songwriter chooses to generate musical content by assigning an instrument to each track, creating two empty clips and clicking generate in our plug-in. A generative model then returns samples corresponding to the specified instruments into these clips (shown in red). If the generated clips are not to the songwriter's liking, they can undo and re-generate, selectively retain portions of each clip, or record in new notes with the keyboard as desired.

3.1.3 Improvising with music AI. Our real-time improvisational tool will take advantage of the same interface and integration described in Section 3.1.2. In the live setting in Figure 1(b), a user composes a chord track and chooses one track to capture real-time input (e.g. from a digital keyboard). After the plug-in is set to live mode, the plug-in monitors the playhead (shown in Figure 1(b) as a vertical black line) and generates new sequences in chunks corresponding to a generation interval (equivalent to δ_2 in Figure 2). Playback is handled entirely by the DAW, so the user has full control over how each accompanying track produces sound alongside their performance. Integration with a DAW also makes recording this performance straightforward. Consequently, a user can flexibly change between the compositional mode of Section 3.1.2, and the improvisational mode described here.

3.2 Adapting Anticipatory Models for Live Improvisation

Whereas standard generative pretrained Transformer (GPT) models predict the next note given the past, the anticipatory music transformer [23] predicts the next note given both the past and a short window of upcoming notes in the future. Anticipatory models thus support flexible, non-linear composition workflows within a song that is in progress of being composed. Adapting the anticipatory music transformer for the task of live improvisation presents several technical challenges: if a generative model is to play along with humans in real-time, it must both be able to look ahead to upcoming chord changes *and* take into account that notes played live by a human will be received with some latency in any practical system. Thus, we detail three technical contributions of this work: (1) creating a dataset of lead sheets in order to make a lead-sheet conditioned model, (2) adapting anticipatory models to handle latency in human input, and (3) expanding the anticipatory music transformer to provide control over the instruments generated.

3.2.1 Generating lead sheets to train a chord-conditioned anticipatory model. To our knowledge, there is no readily available dataset of paired lead sheets and symbolic music performances. To enable training a chord-conditioned anticipatory model, we create chord progressions for each song in the Lakh MIDI dataset [19] using a heuristic

[1]https://github.com/caenopy/lab

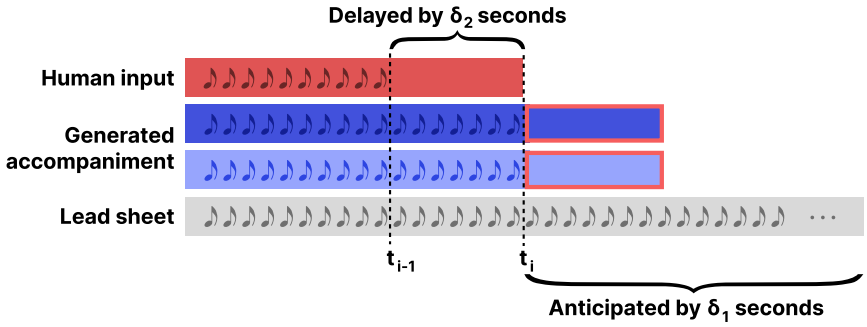

Fig. 2. We train an anticipatory model to generate musical accompaniment conditioned on chords anticipated by δ_1 seconds and human input delayed by δ_2 seconds. At time t_i, the model is prompted to generate in δ_2 second chunks, as highlighted above.

algorithm based on [7] designed to identify the harmonic content of each bar of music. We then construct training examples with anticipated control sequences of these chords, following the procedure outlined in [23].

3.2.2 Accounting for latency in human input with anti-anticipation. We mitigate variability in this latency by requiring the model to generate live accompaniment in chunks of a fixed interval (highlighted in Figure 2). We train an autoregressive anticipatory model to generate music conditioned on anticipated chord changes and delayed human input. We do this by adapting the anticipation method to "anticipate" notes in the chord track by δ_1 seconds and "anti-anticipate" notes played by a human by δ_2 seconds (Figure 2), where δ_2 is the generation interval at inference time. We achieve the latter by extending the anticipatory modeling framework to allow for anticipation by a negative time interval.

3.2.3 Instrument control. Providing musicians control over what instruments are generated as an accompaniment is crucial for making music co-creation systems more steerable. We expand the event vocabulary in [23] to include global instrument control tokens such that the model can generate sequences conditioned on specified instruments.

4 DISCUSSION AND CONCLUSION

It is currently difficult for AI researchers to deploy generative models for symbolic music in an environment accessible to musicians, and we believe this to be one reason for why the adoption of generative AI creativity support tools for symbolic music lags behind that of other domains [2, 8, 21]. Standalone generative AI experiences place a critical bottleneck on studying what types of interactive controls are important for giving domain experts effective co-creation tools. By bringing these models into the musician's toolkit, we take a first step that is needed to study the design-space of creative tooling for domain experts.

One context where this observation is particularly salient is live performance, where real-time interaction is largely supported by integration in the digital audio workstation. The affordances of the DAW make it easier for musicians to integrate generative AI alongside the instruments, effects, and presets familiar in their practice. However, we acknowledge that DAWs are only one of many entry points researchers can take in exploring live musical performance as a human-AI co-creation paradigm. In this paper, we've considered one perspective on musical improvisation which has shaped the design and implementation of our system. We look forward to future work exploring ways generative AI can support other co-performance domains in real-time.

REFERENCES

[1] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. 2019. Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. http://arxiv.org/abs/1906.02569 arXiv:1906.02569 [cs, stat].

[2] Adobe. 2024. Photoshop AI Features. https://www.adobe.com/products/photoshop/ai.html. Accessed: 2024-02-23.

[3] Ableton AG. 2024. Ableton Live: Music Production Software. https://www.ableton.com. Accessed: 2024-02-23.

[4] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. 2023. MusicLM: Generating Music From Text. https://doi.org/10.48550/arXiv.2301.11325 arXiv:2301.11325 [cs, eess].

[5] Théis Bazin and Gaëtan Hadjeres. 2019. NONOTO: A Model-agnostic Web Interface for Interactive Music Composition by Inpainting. https://doi.org/10.48550/arXiv.1907.10380 arXiv:1907.10380 [cs, eess].

[6] Christodoulos Benetatos, Joseph VanderStel, and Zhiyao Duan. 2020. BachDuet: A Deep Learning System for Human-Machine Counterpoint Improvisation. (June 2020). https://doi.org/10.5281/ZENODO.4813234 Publisher: Zenodo.

[7] Joshua Chang. 2024. chorder 0.1.2. https://github.com/joshuachang2311/chorder. GitHub repository.

[8] Minsuk Chang, Stefania Druga, Alex Fiannaca, Pedro Vergani, Chinmay Kulkarni, Carrie Cai, and Michael Terry. 2023. The Prompt Artists. http://arxiv.org/abs/2303.12253 arXiv:2303.12253 [cs].

[9] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. 2024. Simple and Controllable Music Generation. https://doi.org/10.48550/arXiv.2306.05284 arXiv:2306.05284 [cs, eess].

[10] Jeff Ens and Philippe Pasquier. 2020. MMM : Exploring Conditional Multi-Track Music Generation with the Transformer. (2020).

[11] Hugging Face. 2024. Hugging Face Spaces - Hosted ML demo pages with no fuss. https://huggingface.co/spaces. Accessed: 2024-02-23.

[12] Gaëtan Hadjeres and Léopold Crestel. 2021. The Piano Inpainting Application. http://arxiv.org/abs/2107.05944 arXiv:2107.05944 [cs, eess].

[13] Cheng-Zhi Anna Huang, Curtis Hawthorne, Adam Roberts, Monica Dinculescu, James Wexler, Leon Hong, and Jacob Howcroft. 2019. The Bach Doodle: Approachable music composition with machine learning at scale. https://doi.org/10.48550/arXiv.1907.06637 arXiv:1907.06637 [cs, eess, stat].

[14] Nan Jiang, Sheng Jin, Zhiyao Duan, and Changshui Zhang. 2020. RL-Duet: Online Music Accompaniment Generation Using Deep Reinforcement Learning. http://arxiv.org/abs/2002.03082 arXiv:2002.03082 [cs, eess].

[15] Jingyi Li, Eric Rawn, Jacob Ritchie, Jasper Tran O'Leary, and Sean Follmer. 2023. Beyond the Artifact: Power as a Lens for Creativity Support Tools. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3586183.3606831

[16] Ryan Louie, Andy Coenen, Cheng Zhi Huang, Michael Terry, and Carrie J. Cai. 2020. Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–13. https://doi.org/10.1145/3313831.3376739

[17] Martin E. Malandro. 2023. Composer's Assistant: An Interactive Transformer for Multi-Track MIDI Infilling. https://doi.org/10.48550/arXiv.2301.12525 arXiv:2301.12525 [cs, eess].

[18] Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. 2018. This Time with Feeling: Learning Expressive Musical Performance. http://arxiv.org/abs/1808.03715 arXiv:1808.03715 [cs, eess].

[19] Colin Raffel. 2016. Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching. (2016).

[20] Adam Roberts, Jesse Engel, Yotam Mann, Jon Gillick, Claire Kayacik, Signe Nørly, Monica Dinculescu, Carey Radebaugh, Curtis Hawthorne, and Douglas Eck. 2019. Magenta Studio: Augmenting Creativity with Deep Learning in Ableton Live. In *Proceedings of the International Workshop on Musical Metacreation (MUME)*. http://musicalmetacreation.org/buddydrive/file/mume_2019_paper_2/

[21] RunwayML. 2024. Runway | Create Impossible Video. https://runwayml.com/. Accessed: 2024-02-23.

[22] Renaud Bougueng Tchemeube, Jeffrey Ens, Cale Plut, Philippe Pasquier, Maryam Safi, Yvan Grabit, and Jean-Baptiste Rolland. 2023. Evaluating Human-AI Interaction via Usability, User Experience and Acceptance Measures for MMM-C: A Creative AI System for Music Composition, Vol. 6. 5769–5778. https://doi.org/10.24963/ijcai.2023/640 ISSN: 1045-0823.

[23] John Thickstun, David Hall, Chris Donahue, and Percy Liang. 2023. Anticipatory Music Transformer. https://doi.org/10.48550/arXiv.2306.08620 arXiv:2306.08620 [cs, eess, stat].

Received 27 February 2024