Cococo: AI-Steering Tools for Music Novices Co-Creating with Generative Models

Ryan Louie
Northwestern University
Evanston, IL
ryanlouie@u.northwestern.edu

Andy Coenen
Google Research
Mountain View, CA
andycoenen@google.com

Cheng Zhi Huang
Mountain View, CA
chengzhiannahuang@gmail.com

Michael Terry
Google Research
Cambridge, MA
michaelterry@google.com

Carrie Cai
Google Research
Mountain View, CA
cjcai@google.com

ABSTRACT

While generative deep neural networks (DNNs) have demonstrated their capacity for creating novel musical compositions, less attention has been paid to the challenges and potential of co-creating with these musical AIs, especially for novices. In a needfinding study with a widely used, interactive musical AI, we found that the AI can overwhelm users with the amount of musical content it generates, and frustrate them with its non-deterministic output. To better match co-creation needs, we developed AI-steering tools, consisting of Voice Lanes that restrict content generation to particular voices; Example-Based Sliders to control the similarity of generated content to an existing example; Semantic Sliders to nudge music generation in high-level directions (e.g., happy/sad); and Multiple Alternatives to generate multiple possibilities to choose from. In a summative study (N=21), we discovered the tools not only increased users' trust, control, comprehension, and sense of collaboration with the AI, but also contributed to a greater sense of self-efficacy and ownership of the composition relative to the AI.

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**; User studies; Collaborative interaction.

ACM Reference Format:
Ryan Louie, Andy Coenen, Cheng Zhi Huang, Michael Terry, and Carrie Cai. 2020. Cococo: AI-Steering Tools for Music Novices Co-Creating with Generative Models. In *IUI '20 Workshops, March 17, 2020, Cagliari, Italy*. ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

Recent generative music models have made it conceivable for novices to create an entire musical composition from scratch, in partnership with a generative model. For example, the widely available Bach Doodle [23] sought to enable anyone on the web to create a four-part chorale in the style of J.S. Bach by writing only a few notes, allowing an AI to fill in the rest. While this app makes it conceivable for even novices with no composition training to create music, it is not clear how people perceive and engage in co-creation activities like these, or what types of capabilities they might find useful.

In a need-finding study we conducted to understand the novice-AI co-creation process, we found that generative music models can sometimes be quite challenging to co-create with. Novices experienced *information overload*, in which they struggled to evaluate and edit the generated music because the system created too much content at once. They also struggled with the system's *non-deterministic output*. While the output would typically be coherent, it would not always align with users' musical goals at the moment. Having surfaced these challenges, this paper seeks to understand what interfaces and interactive controls for generative models are important to promote an effective co-creation experience.

As a step towards explicitly designing for music novices co-creating with generative models, we present Cococo (collaborative co-creation), a music editor web-interface for novice-AI co-creation that augments standard generative music interfaces with a set of AI-steering tools: 1) *Voice Lanes* that allow users to define for which time-steps (e.g. measure 1) and for which voices (e.g. soprano, alto, tenor, bass) the AI generates music, before any music is created, 2) an *Example-based Slider* for expressing that the AI-generated music should be more or less like an existing example of music, 3) *Semantic Sliders* that users can adjust to direct the music toward high-level directions (e.g. happier / sadder, or more conventional / more surprising), and 4) *Multiple Alternatives* for the user to select between a variety of AI-generated options. To implement the sliders, we developed a *soft priors* approach that encodes desired qualities specified by a slider into a prior distribution; this soft prior

[1] This workshop paper is a condensed retelling of work detailed in the paper "Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models" accepted at CHI '20 http://dx.doi.org/10.1145/3313831.3376739

[2] This work was completed during the first author's summer internship at Google.

is then used to alter a model's original sampling distribution, in turn influencing the AI's generated output.

In a summative evaluation with 21 music novices, we found that AI-steering tools not only increased users' trust, control, comprehension, and sense of collaboration with the AI, but also contributed to a greater sense of self-efficacy and ownership of the composition relative to the AI. We also reveal how AI-Steering tools affected novices co-creation process, such as by working with smaller, semantically-meaningful components and reducing the non-determinism in AI-generated output. Taken together, these findings inform the design of future human-AI interfaces for co-creation.

2 RELATED WORK

2.0.1 Human-AI Co-creation. The acceleration of AI capabilities has renewed interest in how AI can enable human-AI co-creation in domains such as drawing [5, 9, 26, 30], creative writing [4, 13], design ideation [27], video game content generation [17], and dance [25]. Across these domains, a core challenge has been developing collaborative AI agents that can adapt their actions based on the goals and behaviors of the user. To this end, some systems design the AI to generate output conditioned upon the surrounding context of human-generated content [4, 9, 13], while others leverage user feedback to better align AI behavior to user intents [5, 17, 27]. Research has also observed that users desire to take initiative in their partnership with AI [30], with controllability and comprehensibility being key challenges to realizing this vision [1]. Building on this need, our work enables users to express their preferences to an AI collaborator through a variety of means.

Much of the prior work in this space has focused on the domains of drawing or writing. There have been relatively fewer HCI efforts examining generative DNN music agents of similar prowess. Building on prior work examining AI as a peer in the creative process, our work contributes to the broader literature by investigating human-AI co-creation in music.

2.0.2 Interactive Interfaces for ML Music Models. To support music makers in the composition process, researchers have developed ML-powered interfaces and devices that map user inputs to musical structures so users can interactively explore musical variations. Examples of such systems include those that allow users to find chords to accompany a melody [16, 35], experiment with adventurous chord progressions [12, 22], use custom gestural inputs to interpolate between synthesizer sounds [11], or turn free-hand sketches into harmonious musical textures [10].

More recently, progress in generative DNNs has introduced fully-generative music interfaces capable of performing auto-completion given a seed of user-specified notes [18, 23, 33]. Beyond supporting single sub-components, these systems can produce full scores that automatically mesh well with local and distant regions of music. Thus, there is potential to now support users in a wide range of musical tasks (e.g., harmonizing melodies, elaborating existing music, composing from scratch), all within one interface. While recent research has made these fully-generative interfaces increasingly available to musicians and novices alike [7, 18, 23, 33], there has been relatively little HCI work examining how to design interactions with these contemporary models to ensure they are effective for co-creation, especially for novices. Our research contributes an integrative understanding of how interfaces to these capable AIs can be designed and used, how these capabilities affect the composing experience, and users' attitudes towards AI co-creation.

2.0.3 Deep Generative Music Models. As their name implies, generative deep neural networks can synthesize content. Research has demonstrated the potential for modeling and synthesizing music, ranging from single-voice sequences [8] and multi-part music [14, 28], to music with variable parts at each time step [2] and music with long-term structure over minutes [20, 24, 31].

In contrast to models that (typically) generate music chronologically from left to right, *in-filling* models can more flexibly support co-creation by allowing users to specify regions at any point in the music, then auto-filling those gaps. Examples include DeepBach [18] and Coconet [21], both trained on four-part Bach Chorales. Researchers have also created models designed to support interaction mechanisms that grant users more control. For example, there are emerging approaches aimed at learning a continuous latent space so that users can interpolate between music [32], or explore a space of musical alternatives [6]. In our work, we adopt *soft priors* as a general approach that provides additional ways for users to direct their exploration. In contrast to hard constraints, our approach allows DNNs to simultaneously consider the original context (encoded in the model's original sampling distribution) and additional desired qualities (encoded in a soft prior distribution), without needing to retrain the model.

3 FORMATIVE NEEDFINDING STUDY

Our research focuses focus on the needs of music novices co-creating music with deep generative generative models. Thus, we conducted a 25 minute elicitation study with 11 music composition novices to understand their challenges when interacting with a deep generative model to compose music. The interface mirrored the generative *infilling* capabilities found in conventional interfaces for deep generative models [23], where users can manually draw notes and request the AI to fill in the remaining voices and measures, or erase any part of the music and request the AI to fill in the gap. Overall, we found that users struggled to evaluate the generated music and express desired musical elements, due to *information overload* and *non-deterministic output*.

3.0.1 Information Overload. While the deep generative models were capable of infilling much of the song based on only a few notes from the user, participants found the amount of generated content overwhelming to unpack, evaluate, and edit. Specifically, they had difficulty determining why a composition was off, and expressed frustration at the inability to work on smaller, semantically meaningful parts of the composition. For example, one user struggled to identify which note was causing a discordant sound after multiple generated voices were added to their original: *"It was difficult because all the notes were put on the screen already... I can identify places where it doesn't sound very good, but it's actually hard to identify the specific note that is off."* Some participants naturally wanted to work on the composition *"bar-by-bar or part-by-part"*; in contrast to expectations, the generated output felt like it *"skipped a couple steps"* and made it difficult to follow all at once: *"Instead of* *giving me four parts of harmony, can it just harmonize one? I can't manage all four at once."*

3.0.2 Non-deterministic output. Even though the AI was capable of generating notes that were technically coherent to the context of surrounding notes provided by users, the stochastic nature of the system meant that its output did not always match the user's current musical objectives. For example, a participant who had manually created a dark, suspenseful motif was dismayed with how the generated notes were misaligned with the original feeling of the motif: *"the piece lost the essence of what I was going for. While it sounds like nice music to play at an upscale restaurant, the sense of climax is not there anymore."* Even though what was produced sounded harmonious to the user, they felt incapable of giving feedback about their goal in order to constrain the kinds of notes the model generated. Despite being *technically* aligned to context, the music was *musically* mis-aligned with user goals. As a result, participants wished there were ways to go beyond randomly *"rolling dice"* to generate a desired sound, and instead control the generation based on relevant musical objectives.

4 COCOCO

Based on identified user needs, we developed Cococo (collaborative co-creation), a music editor web-interface for novice-AI co-creation that augments standard generative music interfaces with a set of *AI steering tools* (Figure 1). Cococo builds on top of Coconet [21], a state-of-the-art deep generative model trained on 4 part harmony that accepts incomplete music as input and outputs complete music. Coconet works with music that can have 4 parts or *voices* playing at the same time (represented by **S**oprano **A**lto **T**enor **B**ass), are 2-measures long or 32 *timesteps* of sixteenth-note beats, and where each voice can take on any one of 46 *pitches*. Coconet is able to *infill* any section of music, including gaps in the middle or start of the piece. To mirror the most recent interfaces backed by these infill capabilities [7, 18], Cococo contains an *infill mask* feature, with which users can crop a passage of notes to be erased using a rectangular mask, and automatically infill that section using AI. Users can also manually draw and edit notes.

Beyond the infill mask, Cococo distinguishes itself with its *AI steering tools*. In the following subsections, we describe in detail each of the four tools. Additionally, we illustrate the co-creation workflow enabled by these tools in Figure 1.

4.1 Voice Lanes

Voice Lanes within Cococo allows a user to specify the voice(s) for which to generate music within a given temporal range. With this capability, users can control the amount of generated content they would like to work with. This was designed to address information overload caused by Coconet's default capabilities to infill all remaining voices and sections at a time. For example, a user can request the AI to add a single accompanying bass line to their melody by highlighting the bass (bottom) voice lane for the duration of the melody, prior to clicking the generate button (see Figure 1b). To support this type of request, we pass a custom generation mask to the Coconet model including only the user-selected voices and time-slices to be generated.

4.2 Audition Multiple Alternatives

Cococo provides affordances for auditioning multiple alternatives generated by the AI. This capability was designed based on formative feedback, in which users wanted a way to cycle through several generated suggestions to decide which was the most desirable. We allow the user to select the number of alternatives to be generated and displayed (with a default of three). A thumbnail preview of each alternative is displayed and can be selected for audition within the editor, allowing the user to hear it within the larger musical context. The musical chunk used as a prior to generation is accessible via the top thumbnail preview (labeled "original") so that users can always compare what the previous version of the piece sounded like, and opt to not use any of the generated alternatives.

4.3 Example-based Slider

While prototyping the Multiple Alternatives feature, we found that the non-determinism inherent in a deep generative model like Coconet can lead to two opposite, but undesirable extremes: generated samples can be too random and unfocused, or they can be too similar to each other and lack diversity. For example, when the generation area was small relative to surrounding context, generated results would become repetitive: there were a limited set of likely notes for this context according to the model. As a solution, we developed the example-based slider for expressing that the AI-generated music should be more or less like an existing example of music. Before this slider is enabled, the user must select a reference example chunk of notes, either by using the most recent set of notes generated by AI, or manually selecting a reference pattern using the voice lanes or infill mask. Example-based sliders also use soft priors to guide music generation.

4.4 Semantic Sliders

We implemented two semantic sliders in Cococo to influence what the generative DNN creates: a conventional vs. surprising slider, and a major (happy) vs. minor (sad) slider. This was based on formative observations that users wanted to control both musical qualities (e.g., how much the musical phrase or motif should stand out from what is already there) and emotional qualities (e.g., should the notes together produce happy or sad tones).

Users can make the generated notes more predictable given the current context by specifying more "conventional" on the slider, or more unusual by specifying more "surprising." The conventional/surprising slider adjusts the *temperature* (T) of the sampling distribution [15]. A lower temperature makes the distribution more "peaky" and even more likely for notes to be sampled that had higher probabilities in the original distribution (conventional), while higher temperatures makes the distribution less "peaky" and sampling more random (surprising).

The major vs. minor slider allows users to direct the AI to generate note combinations with a happier (major) quality or a sadder (minor) quality (See Figure 1D). To generate a passage that follows a more major or minor tone, we define a soft prior (described below) that encourages the sampling distribution to generate the most-likely major triad (for happy) or non-major triad (for sad) at each time-step.

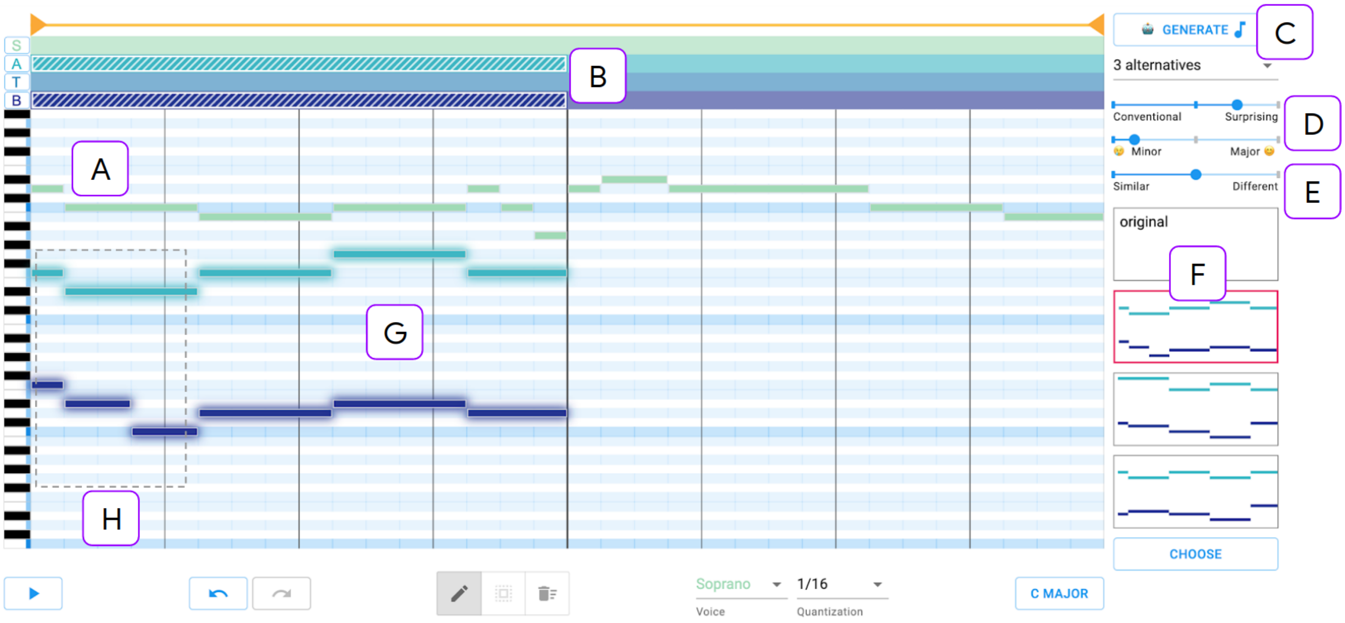

Figure 1: Users of Cococo can manually write some notes (A), specify which voices and in which time range to request AI-generated music using Voice Lanes (B), click Generate (C) infill the music given the existing notes, constrain generation along specific dimensions of interest using the Semantic Sliders (D) and Example-Based Slider (E), or audition Multiple Alternatives (F) of generated output by selecting a sample thumbnail to temporarily substitute it into the music score (shown as glowing notes in this figure (G)). Users can also use the Infill Mask (H) to crop a section of notes to be infilled again using AI.

4.5 Soft Priors: a Technique for AI-Steering

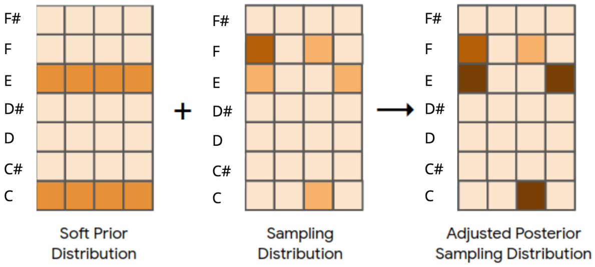

Figure 2: Visualization of using soft priors to adjust a model's sampling distribution. The shape of the distributions are simplified to 1 voice, 7 pitches, and 4 timesteps. In CoCoCo, the actual shape is 4 voices, 46 pitches, and 32 timesteps

The Example-based and Semantic Sliders make use of a soft prior to modulate the model's generated output. These priors enable users or an AI-steering tool designer to add control to existing generative models without needing to retrain them. The model's sampling distribution is a softmax [15] probability distribution over all possible pitches, for each voice and for each time step; high probabilities are assigned to the pitches that are likely given the infill's surrounding musical context. The soft prior approach enables the generation of output that adheres to *both* the surrounding context (encoded in the model's sampling distribution) and additional desired qualities (encoded in a prior distribution). More formally, we use the equation below to alter the distribution used to generate outputs:

$$p_{\text{adjusted}}(x_{v,\,t}|x_C) \propto p_{\text{coconet}}(x_{v,\,t}|x_C)\, p_{\text{softprior}}(x_{v,\,t})$$

where $p_{\text{coconet}}(x_{v,t}|x_C)$ gives the sampling distribution over pitches for voice v at time t from Coconet given musical context x_C (C gives the set of v, t positions constituting the context), $p_{\text{softprior}}(x_{v,t})$ encodes the distribution over pitches specified by the user or AI-steering tool designer (serving as soft priors), and $p_{\text{adjusted}}(x_{v,t}|x_C)$ gives the resulting adjusted posterior sampling distribution over pitches.

The soft priors $p_{\text{softprior}}(x_{v,t})$ are defined so that notes that should be encouraged are given a higher probability, and those discouraged are given a lower, but non-zero probability. This setup allows for two desirable properties. First, since none of the note probabilities are forced to zero, very probable notes in the model's original sampling distribution can still be likely after incorporating the priors. Second, even though the priors are specified for particular voice and time steps, their effects can propagate to other parts of the piece. For example, as Coconet fills in the music, it will try to generate transitions that go smoothly between parts with a soft prior and parts without. Together, these make it possible for the model's output to adhere to both the original context and the additional user-desired qualities.

The soft priors technique powers Cococo's example-based slider and semantic sliders. When the user sets the example-based slider to more "similar," we create a soft prior that has higher probabilities for notes in the example. Conversely, for a slider setting of more "different," we create a soft prior that has lower probabilities for notes in the example. The soft prior is then used to alter the sampling distribution according to Equation 1 and Figure 2.

The minor/major slider uses a slightly more complicated approach to define the soft prior distribution. To encourage notes from a major chord, for example, we construct the soft prior by asking what is the most likely major chord at each time slice within the model's sampling distribution. The log likelihood of a chord is computed by summing the log probability of all the notes that could be part of the chord (e.g., for C major chord, this includes all the Cs, Es, and Gs in all octaves). We repeat this procedure for all possible major chords to determine which chord is the most likely for a time slice. We then repeat this procedure for all time slices to be generated, in order to create our soft prior for most likely major chords; this soft prior is used to alter the sampling distribution to create the adjusted posterior sampling distribution as shown in Figure 2.

The features described above were implemented as a React.js web application[3], backed by an open source browser-based implementation [33] of the Coconet model. We modified Coconet to include soft priors.

5 USER STUDY

We conducted a user study to evaluate the extent to which AI-steering tools support novice composers' needs, and to uncover how the tools affect the user experience of co-creating with AI. To this end, we used a within-subjects experimental design to compare the experience using Cococo to that of the conventional interface. The conventional interface is aesthetically similar to Cococo, but does not contain the AI-steering tools. To mirror the most recent deep generative music interfaces, the conventional interface does include the *infill-mask* feature, which enables users to crop any region of the music and request that it be filled in by the AI [7, 18].

5.0.1 Method. 21 music composition novices participated in the study. Each participant first completed an online tutorial of the two interfaces on their own (30 minutes). Then, they composed two pieces, one with Cococo and one with the conventional interface, with the order counterbalanced (15 minutes each). As a prompt, users were provided a set of images from the card game Dixit [36] and were asked to compose music that reflected the character and mood of one image of their choosing. This task is similar to image-based tasks used in prior music studies [22]. Finally, they answered a post-study questionnaire and completed a semi-structured interview (20 minutes). So that we could understand their thought process, users were encouraged to think aloud while composing.

5.0.2 Measures. For our quantitative questionnaire, we evaluated the following outcome metrics to understand users' compositional experience and attitudes towards the AI. All items below were rated on a 7-point Likert scale (1=Strongly disagree, 7=Strongly agree) except where noted below.

The following set of metrics sought to measure users' compositional experience. **Creative expression**: Users rated *"I was able to express my creative goals in the composition made using [System X]."* **Self-efficacy**: Users answered two items from the Generalized Self-Efficacy scale [34] that were rephrased for music composition. **Effort**: Users answered the effort question of the NASA-TLX [19], where 1=very low and 7=very high. **Engaging**: Users rated *"Using [System X] felt engaging."* **Learning**: Users rated *"After using [System X], I learned more about music composition than I knew previously."* **Completeness** of the composition: Users rated *"The composition I created using [System X] feels complete (e.g., there's nothing to be further worked on)."* **Uniqueness** of the composition: Users rated *"The composition I created using System X feels unique."*

In addition, we evaluated users' attitudes towards the AI. **AI interaction issues**: Users rated the extent to which the system felt *comprehensible* and *controllable*, two key challenges of human-AI interaction raised in prior work on DNNs [30]. **Trust**: Participants rated the system along Mayer's dimensions of trust [29]: capability, benevolence, and integrity. **Ownership**: Users rated two questions, one on ownership (*"I felt the composition created was mine."*), and one on attribution (*"The music created using [System X] was 1=totally due to the system's contributions, 7=totally due to my contributions."*). **Collaboration**: Users rated *"I felt like I was collaborating with the system."* The following set of metrics sought to measure users' compositional experience:

In the study, we called the two systems "System 1" and "System 2" (counterbalanced) to avoid biasing participants, but we refer to them here as "Cococo" and "conventional interface" for clarity.

6 QUANTITATIVE FINDINGS

Results from the post-study questionnaire are shown in Figure 3. We conducted paired t-tests using Benjamani-Hochberg correction to account for the 15 planned-comparisons, using a false discovery rate $Q = 0.05$.

In regards to users perceptions of the creative process, we found Cococo significantly improved participants ability to **express their creative goals**, **self-efficacy**, perception of **learning more** about music, and **engagement** compared to the conventional interface. No significant difference was found in **effort**; participants described the two systems as requiring different kinds of effort: While Cococo required users to think and interact with the controls, the conventional interface's lack of controls made it effortful to express creative goals. Users perceptions of the **completeness** of their composition made with Cococo was significantly higher than the conventional interface; however, no significant difference was found for **uniqueness**.

The comparisons for users' attitudes towards the AI were all found to be statistically significant: Cococo was more **controllable**, **comprehensible**, and **collaborative** than the conventional interface; participants using Cococo expressed higher **trust** in the AI, felt more **ownership** over the composition, and **attributed** the music to more of their own contributions relative to the AI.

7 QUALITATIVE FINDINGS

In this section, we first report how AI-Steering tools supported novices' composing strategies and experience, including 1) working with smaller, semantically meaningful components and 2) reducing non-determinism through testing a variety of constrained settings

[3] https://github.com/pair-code/cococo

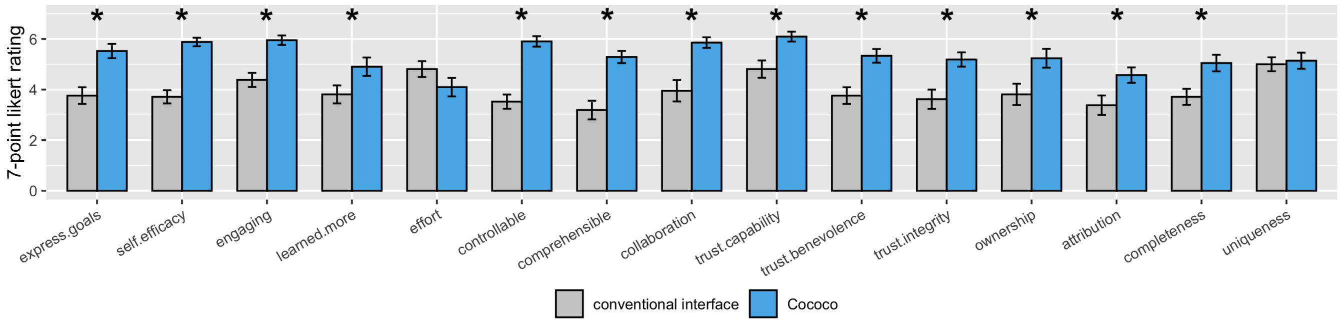

Figure 3: Results from post-study survey comparing the conventional interface and Cococo, with standard error bars.

for generation. We then describe 3) how novice's prior mental models shaped their interaction with AI.

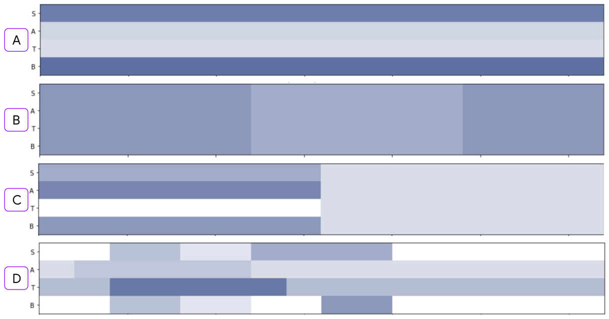

Figure 4: Common Patterns of using Voice Lanes, visualized using interaction data from 4 archetypal participants (darker-colored segments were performed by users before lighter-colored segments): (A) Voice-by-voice (most common), (B) Temporal Chunks, (C) Combination of Voice-by-Voice and Temporal Chunks, and (D) Ad-hoc Bits

7.1 Effects of Partitioning AI Capabilities into Semantically-Meaningful Components

7.1.1 Composing Bit-By-Bit. AI-Steering tools allowed participants to build up the composition from smaller components, bit-by-bit. Many participants used the Voice Lanes to iteratively develop one voice layer at a time, in a "brick-building" fashion (Figure 4a): *"I'm trying to get the bass right, then the tenor right, then soprano and alto right, and build bit-by-bit"* (P2). Others used the temporal aspect of the Voice Lanes, by generating all four voices in a temporal region (Figure 4b). Some tried a combination of the voice-wise and temporal approaches, by working voice-wise in the first half of the song, then letting the AI continue a whole temporal chunk in the second half (Figure 4c).

Participants who worked bit-by-bit thought about their compositions in semantically-meaningful chunks. They used separate voices to differentiate between melody and background or between different musical personas. For example, one participant gave the tenor voice an *"alternating [pitch] pattern"* to express indecision in the main melody, then gave other voices *"mysterious... dinging sounds"* as a harmonic backdrop (P4). Some also divided the music into temporally distinct chunks to illustrate evolution in time. One participant requested more conventional chords to make the introduction more calm, then requested more minor and surprising notes to build tension as the piece progressed.

7.1.2 Overcoming Information Overload. Working with smaller components helped participants manage AI-induced information overload. Participants who worked voice-by-voice could better handle the combination of multiple voices: *"As someone who cannot be thinking about all 4 voices at the same time, it's so helpful to generate one at a time"* (P2). Working bit-by-bit gave participants intermediate "checkpoints", where they could *"intervene after [the AI] generated [content]... stop it in the middle... and change it to feel different, before it kept going"* (P14). In contrast, in the conventional interface, the AI fully auto-completed the music at once. As a result, participants resorted to "sculpting" and refining the AI's fully-generated music by repeatedly using the Infill Mask. Echoing the results in our need-finding study, some participants found the amount of resultant content overwhelming.

7.1.3 Identifying and Debugging Problematic AI Output. By building up the music bit-by-bit, users became familiar with their own composition during the creation process, which enabled them to more quickly identify the *"cause"* of problematic areas later on. For example, one participant indicated that *"[because] I had built [each voice] independently and listened to them individually,"* this helped them *"understand what is coming from where"* (P7). Conversely, if multiple voices were generated simultaneously, participants found it difficult to understand the complex interactions: *"It's harder to disentangle what change caused what... when I make a change, there could be this mixed reaction...it propagates to [multiple] things at once"* (P6). By enabling users to generate bit-by-bit from the ground up, and incrementally evaluate the music along the way, the tools may have enabled novices to better understand and subsequently "debug" their own musical creations.

7.1.4 Learning and Discovering Musical Structure. The tools also helped participants learn how sub-components affect the whole piece. One participant described how they came to understand *"that having that soprano up [at this bar]... gives a total injection of a different emotion,"* which they only realized by using the Voice

Lanes to place a single voice within a single bar. Another participant learned that *"a piece can become more vivid by adding both a minor and major chord"* after they applied the major/minor slider to generate two contrasting, side-by-side chunks (P12). Thus, while the conventional AI could do everything on its own, partitioning the AI's capabilities into smaller, semantically meaningful tools helped people learn music composition strategies that they could re-use in the future.

7.2 Effects of Constraining Non-Determinism in Generated Output

AI-Steering tools helped to control the non-deterministic output inherent in the generative model. As a result, the tools allowed users to 1) steer generation in desired directions when composing with AI, and 2) test the limits of the AI by constraining generation with a variety of settings when requesting output.

7.2.1 Steering Music Generation in Desired Directions. Participants who used the Multiple Alternatives feature reduced the uncertainty that AI-generated output would be misaligned with their musical goals. Participants could simply generate a range of possibilities, audition them, and choose the one closest to their goal before continuing.

During different phases of the composing process, participants used the sliders to constrain the large space of possibilities that could be generated. The Semantic Sliders were sometimes used to set an initial trajectory for generated music: *"Because I was able to give more inputs to [Cococo] about what my goals were, it was able to create some things that gave me a starting point"* (P8). In analysis of logs, 12 of the 21 participants modified the default values of the slider parameters prior to their first generation request to Cococo. Sliders were also used to refine what the AI had already generated: *"It was... not dramatic enough. Moving the slider to more surprising, and more minor added more drama at the end"* (P5). Applying the example-based slider, participants moved the setting to "similar" to push content closer to an example that embodied their musical goals: *"Work your magic on these notes, but keep it similar so they won't move around too much"* (P1). They set the slider to "different" when the initial AI-generated notes were *"not sounding good"* (P15) or when all the generated options needed to be *"totally scrapped"* (P13) because all were of opposite quality to the sound the user desired.

7.2.2 Testing the Limits of the AI. The tools also enabled participants to test the limits of the AI. One participant, while using Voice Lanes to generate multiple alternatives for a single-voice harmony, discovered that the AI may be constrained by what's musically possible: *"Maybe the dissonance is happening because of how I had the soprano and bass... which are limiting it... so it's hard to find something that works"* (P15). Here, the Voice Lanes helped this user consider the limits imposed by a specific voice component, enabling them to reflect on the limits of the AI in a more semantically meaningful way. The Multiple Alternatives capability further enabled this participant to systematically infer that this particular setting was unlikely to produce better results through the observation of multiple poor results produced by the AI.

Some participants also set the sliders to their outer limits to test the boundaries of AI output. For example, one participant moved a slider to the "similar" extreme, then incrementally backed it off to understand what to expect at various levels of the slider: *"On the far end of similar, I got four identical generations, and now I'm almost at the middle now, and it's making such subtle adjustments"* (P18). These interactive adjustments allowed the user to quickly explore the limits of what they can expect the AI tools to generate, aiding construction of a mental model of the AI's capabilities. In contrast, when using the conventional interface, participants could not as easily discern whether undesirable model outputs were due to AI limits, or a simple luck of the draw.

7.3 Effects of Users' Prior Mental Models

Participants brought with them prior mental models that impacted how they interacted with the generative model. First, many participants already had a set of primitives for expressing high-level musical goals, including basic concepts such as pitch, note density, and shape. They re-purposed the tools to express these primitives, even when the tools did not explicitly map to these primitives. Second, several participants believed that the AI model was superior to their skills as novice composers. As such, when specific errors arose during the composing process, they often blamed their own efforts for these mistakes and hesitated to play an active role in the process.

7.3.1 Expressing Musical Primitives through Proxy Controls. Many participants seemed to share a common set of musical primitives to express high-level, creative goals. For example, higher pitches were used to communicate a light mood, long notes to convey calmness or drawn-out emotions, and a shape of ascending pitches to communicate triumph and escalation.

During the study, participants who could not find an explicit interface that mapped to these primitives re-purposed the AI-steering tools as "proxy controls" to enact these strategies. For example, users sometimes hoped that the surprising vs. conventional slider would be correlated with note density and tempo. A common pattern was to set the slider to "conventional" to generate music that was *"not super fast... not a strong musical intensity"* (P9), and to "surprising" for generating *"shorter notes... to add more interest"* (P15). Participants also turned to heuristics, such as knowledge that bass lines in music tend to contain lower pitches, to "reverse-engineer" which Voice Lanes to select in an attempt to control pitch range. Multiple steering tools were also sometimes combined to achieve a desired effect, such as using "conventional" in conjunction with the bass Voice Lane to create slow and steady music.

In some cases, even use of the AI-steering tools did not succeed in generating the desired quality. For example, the music produced using the "similar" setting was not always similar along the user-envisioned dimension, and the surprising slider did not systematically map to note density, despite being correlated. Facing these challenges, participants developed a strategy of "leading by example" by populating surrounding context with the type of content they desired from the AI. For instance, one participant manually drew an ascending pattern in the first half of the alto voice, in the hopes that the AI would continue the ascending pattern in the second half.

7.3.2 *Novice Self-Efficacy and Self-Doubt.*

Across most composing scenarios, participants described how AI-steering tools instilled a sense of self-efficacy and agency. Building up bit-by-bit made participants feel *"more useful as a composer"* by helping them to trace how their efforts on smaller components combined together to create the whole musical piece. In contrast, using the conventional interface which generated a full-composition at once felt more like the *"machine is doing all the work"* (P3). In addition, indicating what type of music was generated promoted creative agency: *"the sliders really help to express [myself] in a way [I] wouldn't be able to do in music notes or words"* (P7). Participants attributed their sense of ownership to the availability of choice afforded by the Multiple Alternatives tool: *"There are options, but I don't feel like I have to use them... it's not like the [AI] is telling me 'This is the correct thing to do here'... so I felt I definitely had ownership in the music"* (P9).

While there were indications that the tools helped improve feelings of self-efficacy, there were also times when participants doubted their own musical abilities when they were unable to obtain desirable results. In one scenario, novices experienced self-doubt when poor sounding music was generated using user-composed notes as the input context. Because the AI generates music given a surrounding "seed" context, users who were dissatisfied with AI output often wondered whether they had provided a low-quality seed, leading to suboptimal AI output: *"All the things it's generating sound sad, so it's probably me because of what I generated"* (P11). In cases such as this, participants seemed unable to disambiguate between AI failures and their own compositional flaws, and placed the blame on themselves.

In other scenarios, novices were hesitant to interfere with the AI music generation process. For instance, some assumed that the AI's global optimization would create better output than their own local control of sub-units: *"Instead of doing [the voice lanes] one by one, I thought that the AI would know how to combine all these three [voices] in a way that would sound good"* (P1). While editing content, others were worried that making local changes could interfere with the AI's global optimization and possibly *"mess the whole thing up"* (P3). In these cases, an incomplete mental model of how the system functions seemed to discourage experimentation and their sense of self-efficacy.

8 DISCUSSION

8.0.1 *Partition AI Capabilities into Semantically-Meaningful Tools.*

While generative DNN models can create full artifacts coherent to a surrounding context, their capabilities may need to be partitioned into smaller, semantically meaningful tools to promote effective co-creation. Our results suggest that AI-steering tools played a key role in breaking the co-creation task down into understandable chunks and generating, auditioning, and editing these smaller pieces until users arrived at a satisfactory result.

One unexpected side effect was that novices quickly became familiar with their own creations through composing bit-by-bit, which later helped them debug problematic areas. Interacting through semantically meaningful tools also helped them learn more about music composition and effective strategies for achieving particular outcomes (e.g., the effect of a minor key in the composition). Ultimately, AI-steering tools affected participants' sense of artistic ownership and competence as amateur composers, through an improved ability to express creative intent. In sum, beyond reducing information overload, tools that partition AI capabilities into semantically-meaningful components may be fundamental to one's notion of being a creator, while opening the door for users to learn effective strategies for creating in that domain.

Our work also uncovers the dual challenges and opportunities of sophisticated DNNs: although such models can be difficult to decompose, they also expose a flexible space for modification. We found the use of "soft priors" to be a relatively lightweight method for nudging the AI's output without retraining the model. This particular technical approach is likely to be applicable to human-AI co-creation tooling in domains where a probability sampling distribution is exposeable from a deep generative model. For example, in writing, soft priors could be used to generate text that favors simpler vocabulary or adheres to a particular topic.

8.0.2 *Onboard Users and Divulge AI Limitations.*

While participants were able to develop productive strategies using AI-steering tools, they were sometimes hesitant to make local edits for fear of adversely affecting the AI's global optimization. These reactions suggest that participants could benefit from a more accurate mental model of the AI. Previous research suggests benefits of educating users about the AI and its capabilities [1], or providing onboarding materials and exercises [3]. For example, an onboarding tutorial could demonstrate contexts in which the AI can easily generate content, and situations where it is unable to function well. For instance, the system could automatically detect if the AI is overly constrained and unable produce a wide variety content, and display a warning sign on the tool icon. Or, semantic sliders could divulge certain variables they are correlated with but not systematically mapped to, to set proper expectations when users leverage them as proxies. This could help users better debug the AI when it produces undesirable results. It could also prevent them from incorrectly attributing themselves and their lack of experience in composing as the source of the error, rather than the AI being overly constrained.

8.0.3 *Bridge Novice Primitives with Desired Creative Goals.*

Though we created an initial set of concepts for AI-steering, we were surprised to discover that participants were *already* prepared with their own set of primitives to express high-level creative goals, such a long notes to convey calmness, ascending notes to express triumph and escalation, or temporal separation to convey tension vs. resolution. When they could not find an explicit interactive control for a primitive, they re-purposed the existing tools as proxy controls to achieve the desired effect. Given this, one could imagine directly supporting these common go-to strategies. Given a wide range of possible semantic levers, and the technical challenges of exposing these dimensions in DNNs, model creators should at minimum prioritize exposing dimensions that are the most commonly relied upon. For music novices, we found that these included pitch, note density, shape, voice and temporal separation. Future systems could help boost the effectiveness of novice strategies by helping them bridge between their primitives to high-level creative goals, such as automatically "upgrading" a series of plodding bass line notes to create a foreboding melody.

REFERENCES

[1] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 3, 13 pages. https://doi.org/10.1145/3290605.3300233

[2] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. 2012. Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription. *International Conference on Machine Learning* (2012).

[3] Carrie J. Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. 2019. "Hello AI": Uncovering the Onboarding Needs of Medical Practitioners for Human-AI Collaborative Decision-Making. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 104 (Nov. 2019), 24 pages. https://doi.org/10.1145/3359206

[4] Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A Smith. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces*. ACM, 329–340.

[5] Nicholas Davis, Chih-PIn Hsiao, Kunwar Yashraj Singh, Lisa Li, and Brian Magerko. 2016. Empirically Studying Participatory Sense-Making in Abstract Drawing with a Co-Creative Cognitive Agent. In *Proceedings of the 21st International Conference on Intelligent User Interfaces* (Sonoma, California, USA) *(IUI '16)*. ACM, New York, NY, USA, 196–207. https://doi.org/10.1145/2856767.2856795

[6] Monica Dinculescu, Jesse Engel, and Adam Roberts (Eds.). 2019. *MidiMe: Personalizing a MusicVAE model with user data.*

[7] Monica Dinculescu and Cheng-Zhi Anna Huang. 2019. *Coucou: An expanded interface for interactive composition with Coconet, through flexible inpainting.* https://coconet.glitch.me/

[8] Douglas Eck and Juergen Schmidhuber. 2002. Finding temporal structure in music: Blues improvisation with LSTM recurrent networks. In *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*.

[9] Judith E Fan, Monica Dinculescu, and David Ha. 2019. collabdraw: An Environment for Collaborative Sketching with an Artificial Agent. In *Proceedings of the 2019 on Creativity and Cognition*. ACM, 556–561.

[10] Morwaread M Farbood, Egon Pasztor, and Kevin Jennings. 2004. Hyperscore: a graphical sketchpad for novice composers. *IEEE Computer Graphics and Applications* 24, 1 (2004), 50–54.

[11] Rebecca Anne Fiebrink. 2011. Real-time human interaction with supervised learning algorithms for music composition and performance. *PhD dissertation, Princeton University* (2011).

[12] Satoru Fukayama, Kazuyoshi Yoshii, and Masataka Goto. 2013. Chord-sequence-factory: A chord arrangement system modifying factorized chord sequence probabilities. (2013).

[13] Katy Ilonka Gero and Lydia B Chilton. 2019. Metaphoria: An Algorithmic Companion for Metaphor Creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 296.

[14] Jon Gillick, Adam Roberts, Jesse Engel, Douglas Eck, and David Bamman. 2019. Learning to Groove with Inverse Sequence Transformations. *arXiv preprint arXiv:1905.06118* (2019).

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[16] James Granger, Mateo Aviles, Joshua Kirby, Austin Griffin, Johnny Yoon, Raniero Lara-Garduno, and Tracy Hammond. 2018. Lumanote: A Real-Time Interactive Music Composition Assistant.. In *IUI Workshops*.

[17] Matthew Guzdial, Nicholas Liao, Jonathan Chen, Shao-Yu Chen, Shukan Shah, Vishwa Shah, Joshua Reno, Gillian Smith, and Mark O Riedl. 2019. Friend, collaborator, student, manager: How design of an ai-driven game level editor affects creators. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 624.

[18] Gaëtan Hadjeres, François Pachet, and Frank Nielsen. 2017. DeepBach: a Steerable Model for Bach Chorales Generation. In *International Conference on Machine Learning*. 1362–1371.

[19] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[20] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. 2019. Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. In *International Conference on Learning Representations*.

[21] Cheng-Zhi Anna Huang, Tim Cooijmnas, Adam Roberts, Aaron Courville, and Douglas Eck. 2017. Counterpoint by Convolution. *ISMIR* (2017).

[22] Cheng-Zhi Anna Huang, David Duvenaud, and Krzysztof Z Gajos. 2016. Chordripple: Recommending chords to help novice composers go beyond the ordinary. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*. ACM, 241–250.

[23] 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. *ISMIR* (2019).

[24] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. 2019. Music Transformer. In *International Conference on Learning Representations*.

[25] Mikhail Jacob and Brian Magerko. 2015. Interaction-based Authoring for Scalable Co-creative Agents.. In *ICCC*. 236–243.

[26] Pegah Karimi, Mary Lou Maher, Nicholas Davis, and Kazjon Grace. 2019. Deep Learning in a Computational Model for Conceptual Shifts in a Co-Creative Design System. *arXiv preprint arXiv:1906.10188* (2019).

[27] Janin Koch, Andrés Lucero, Lena Hegemann, and Antti Oulasvirta. 2019. May AI?: Design Ideation with Cooperative Contextual Bandits. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 633.

[28] Feynman Liang. 2016. BachBot: Automatic composition in the style of Bach chorales. *Masters thesis, University of Cambridge* (2016).

[29] Roger C Mayer, James H Davis, and F David Schoorman. 1995. An integrative model of organizational trust. *Academy of management review* 20, 3 (1995), 709–734.

[30] Changhoon Oh, Jungwoo Song, Jinhan Choi, Seonghyeon Kim, Sungwoo Lee, and Bongwon Suh. 2018. I Lead, You Help but Only with Enough Details: Understanding User Experience of Co-Creation with Artificial Intelligence. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 649, 13 pages. https://doi.org/10.1145/3173574.3174223

[31] Christine Payne. 2019. *MuseNet*. https://openai.com/blog/musenet

[32] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. 2018. A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. In *International Conference on Machine Learning (ICML)*. http://proceedings.mlr.press/v80/roberts18a.html

[33] Adam Roberts, Curtis Hawthorne, and Ian Simon. 2018. Magenta.js: A JavaScript API for Augmenting Creativity with Deep Learning. In *Joint Workshop on Machine Learning for Music (ICML)*.

[34] Ralf Schwarzer and Matthias Jerusalem. 1995. Generalized self-efficacy scale. *Measures in health psychology: A user's portfolio. Causal and control beliefs* 1, 1 (1995), 35–37.

[35] Ian Simon, Dan Morris, and Sumit Basu. 2008. MySong: automatic accompaniment generation for vocal melodies. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 725–734.

[36] Wikipedia contributors. 2019. Dixit (card game) — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Dixit_(card_game)&oldid=908027531. [Online; accessed 19-September-2019].