Cococo: AI-Steering Tools for Music Novices Co-Creating with Generative Models[1]

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

ABSTRACT

In this work[1], we investigate how novices co-create music with a deep generative model, and what types of interactive controls are important for an effective co-creation experience. Through a needfinding study, we found that generative AI can overwhelm novices when the AI generates too much content, and can make it hard to express creative goals when outputs appear to be random. To better match co-creation needs, we built Cococo, a music editor web interface that adds interactive capabilities via a set of AI-steering tools. These tools restrict content generation to particular voices and time measures, and help to constrain non-deterministic output to specific high-level directions. We found that the tools helped users increase their control, self-efficacy, and creative ownership, and we describe how the tools affected novices' strategies for composing and managing their interaction with 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 J. Cai. 2020. Cococo: AI-Steering Tools for Music Novices Co-Creating with Generative Models[1]. In *IUI '20 Workshops, March 17, 2020, Cagliari, Italy.* ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

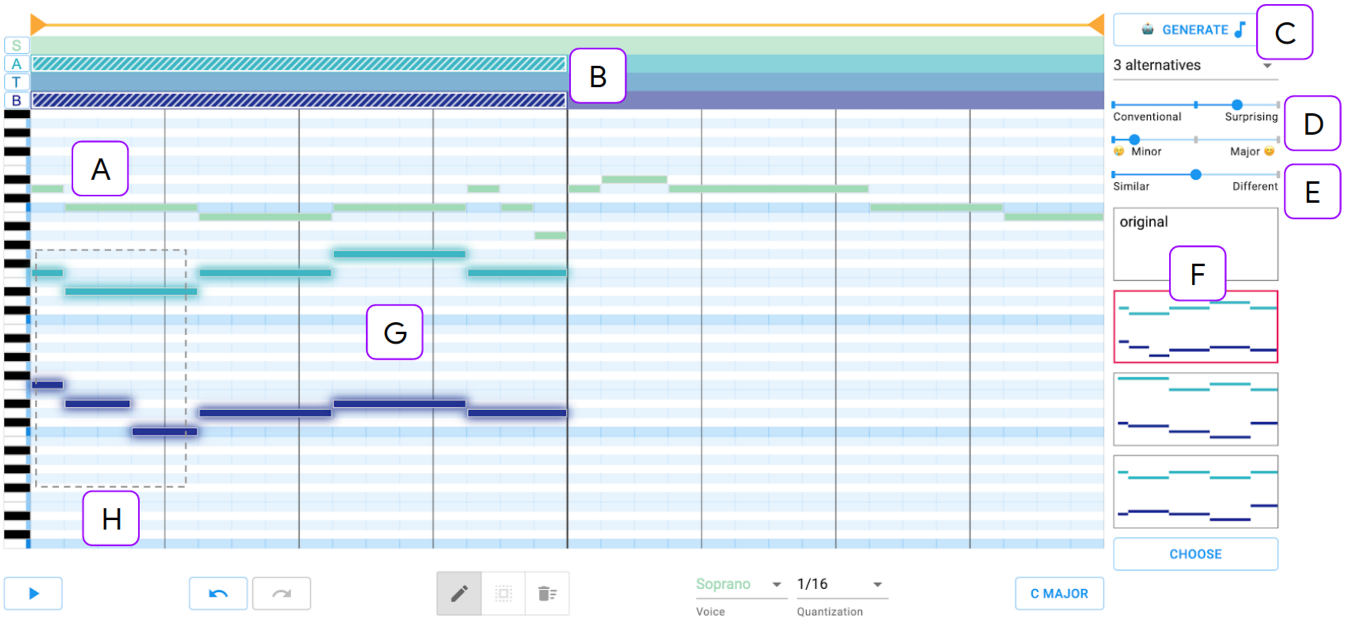
Recent generative music models have made it conceivable for novices to create an entire musical composition from scratch, in partnership with a generative model. For example, the widely available Bach Doodle [9] sought to enable anyone on the web to create a four-part chorale in the style of J.S. Bach 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 in order 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 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. Together, these findings inform the design of future human-AI interfaces for co-creation.

[1] This workshop paper is a shortened summary of the full CHI'20 paper [10]
[2] This work was completed during the first author's summer internship at Google.



3 alternatives

Soprano
Voice

1/16
Quantization

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.

2 NOVICE'S NEEDS FOR CO-CREATION

To understand challenges when composing music with generative models, we conducted a 25 minute needfinding study with 11 music composition novices. We observed novices use a tool that mirrored conventional interfaces for composing music with deep generative models [9].

Participants experienced **information overload**: they struggled to evaluate the generated music due to the amount of AI-generated content. Participants struggled to identify which note was causing a discordant sound after multiple generated voices were added to their original. Participant were naturally inclined to work on the composition *"bar-by-bar or part-by-part"*; however in contrast to these expectations, the generated output felt like it *"skipped a couple steps"* and made it difficult to follow all at once.

Participants struggled to express desired musical objectives due to the AI's **non-deterministic output**. 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. Participants likened this frustrated feeling to *"rolling dice"* to generate a desired sound, and instead wished to control generation based on relevant musical objectives.

3 COCOCO

Based on identified user needs, we developed Cococo (collaborative co-creation), a music editor web-interface [3] 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 [7], 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 [3, 5], 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.

3.0.1 Voice Lanes. Voice Lanes allow specifying for which voice(s) and for which time steps to generate music. 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 (Figure 1B). To support this type of request, we pass a custom generation mask to the Coconet

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

model including only the user-selected voices and time-slices to be generated.

3.0.2 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. Users first choose the number of alternatives to be generated (Figure 1C), audition each alternative by clicking on the different preview thumbnails (Figure 1F), and listen to an alternative which is substituted within the larger musical context (Figure 1G).

3.0.3 Example-based Slider. While prototyping the Multiple Alternatives feature, we found that the non-determinism inherent in Coconet could cause generated samples to be both (1) random and unfocused, or (2) too similar to each other and lack diversity. 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. Example-based sliders use soft priors to guide music generation.

3.0.4 Semantic Sliders. We implemented two semantic sliders in Cococo (Figure 1D) to constrain generated output along meaningful dimensions: a conventional vs. surprising slider, and a major (happy) vs. minor (sad) slider. Users can adjust how predictable vs. unusual notes should be using the "conventional" and "surprising" dimensions of the slider. The conventional/surprising slider adjusts the *temperature* (T) of the sampling distribution [4]. 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 constrains generated notes to a happier (major) quality or a sadder (minor) quality. This slider defines a soft prior that adjusts the sampling distribution to have higher probabilities for the most-likely major triad (for happy) or non-major triad (for sad) at each time-step.

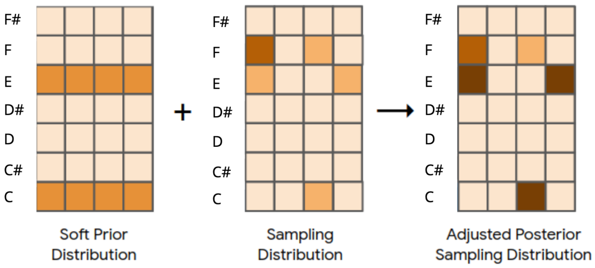


Figure 2: Visualization of using soft priors to adjust a model's sampling distribution. The shape of the distributions are simplified to 1 voice, 7 pitches, and 4 timesteps. In CoCoCo, the actual shape is 4 voices, 46 pitches, and 32 timesteps

3.0.5 Soft Priors: a Technique for AI-Steering. 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). We provide visual intuition for how these distributions interact in Figure 2. 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. 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, thus making it possible for the model's output to adhere to both the original context and the additional user-desired qualities.

The example-based and semantic sliders define a soft prior to modulate the model's generated output. When the user sets the example-based slider to more "similar," Cococo defines a soft prior with higher probabilities for notes in the example. Conversely, for a slider setting of more "different," Cococo defines a soft prior with lower probabilities for notes in the example.

The minor/major slider uses a slightly more complicated approach to define the soft prior distribution. When the user sets the slider to happy (major), for example, Cococo defines the soft prior by asking what is the most likely major triad at each time slice within the model's sampling distribution. The log likelihood of a triad is computed by summing the log probability of all the notes that could be part of the triad (e.g., for C major triad, this includes all the Cs, Es, and Gs in all octaves). We repeat this procedure for all possible major triads to determine which 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 triads.

4 USER STUDY

We conducted a within-subjects study to compare the user experience of 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 [3, 5]. Through a quantitative survey study, we seek to answer **RQ1** How the AI-steering tools in Cococo affects user perceptions of the creative process and the creative artifacts made with the AI. Through qualitative interviews and observations, we seek to understand **RQ2** How music novices apply the AI-steering tools within Cococo in their creative process? What patterns of use and strategies arise?

4.0.1 Method. 21 music composition novices participated in the study. Each participant first completed an online tutorial of the two

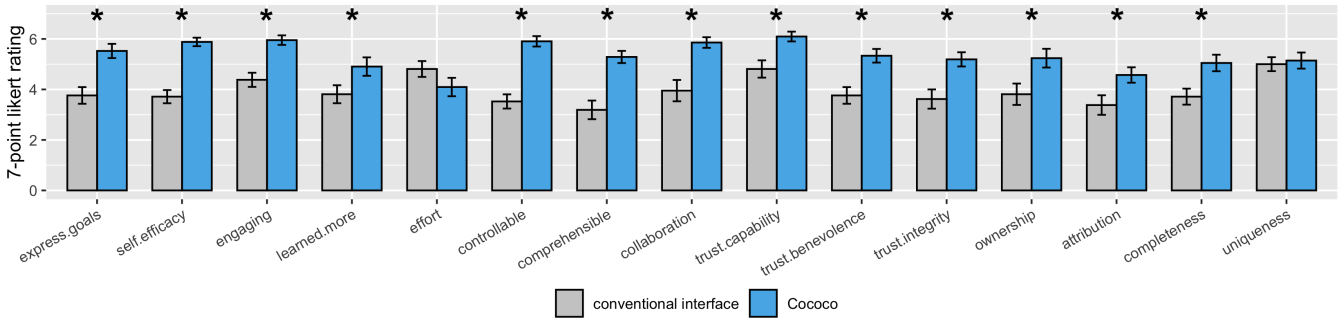


Figure 3: Results from post-study survey comparing the conventional interface and Cococo, with standard error bars.

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 [14] 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 [8]. 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.

4.0.2 Quantitative Measures. For our quantitative questionnaire, we evaluated the following outcome metrics. 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 [13] that were rephrased for music composition. **Effort**: Users answered the effort question of the NASA-TLX [6], 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 [12]. **Trust**: Participants rated the system along Mayer's dimensions of trust [11]: 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."*

5 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.

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

6.1 Effects of Partitioning AI Capabilities into Semantically-Meaningful Components

AI-Steering tools allowed participants to build up the composition from smaller components, bit-by-bit. For example, one participant who used the Voice Lanes said, *"I'm trying to get the bass right, then the tenor right, then soprano and alto right, and build bit-by-bit"* (P2). Participants who worked bit-by-bit thought about their compositions in semantically-meaningful chunks, such as melody vs. background or separate 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).

Working bit-by-bit helped participants feel less overwhelmed and better understand their compositions. For example, those working 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). Participants then 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).

Through this bit-by-bit process, participants learned how subcomponents can combine to achieve desired musical outcomes. For instance, one 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).

6.2 Effects of Constraining Non-Determinism in Generated Output

AI-Steering tools helped to constrain the non-deterministic output inherent in the generative model. As a result, the tools allowed users to steer generation in desired directions when composing with AI. Multiple Alternatives reduced the uncertainty that AI-generated output would be misaligned with a user's 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). 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).

Participants constrained generation by setting the sliders to their outer limits. This enabled them 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). 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.

Participants also used the tools to consider how a specific input configuration affects the limits of AI output. For example, one participant used the Voice Lanes to generate multiple alternatives for a single-voice harmony. This enabled them to consider the limits imposed by specific voice components: *"Maybe the dissonance [in the single-voice] 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). The Multiple Alternatives capability further enabled this participant to systematically infer that the specific configuration of existing voice components was unlikely to produce better results through the observation of multiple poor results generated for the single-voice.

6.3 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. 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. When participants who could not find an explicit tool that mapped to their envisioned primitive, they repurposed the tools as "proxy controls" to enact their strategy. For example, 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).

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. To overcome these challenges, participants developed a strategy of "leading by example" by populating the input 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.

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. While we found evidence that the tools helped improve feelings of self-efficacy (See Quantitative Findings), there were also times when participants doubted their own musical abilities. Novices experienced self-doubt when poor sounding music was generated based off of their user-composed notes as the input context. For example, one user said, *"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 composing 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 had they worked bit-by-bit: *"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.

7 DISCUSSION

7.0.1 Partition AI Capabilities into Semantically-Meaningful Tools. 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. Unexpectedly, 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.

7.0.2 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 [2]. 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.

7.0.3 Bridge Novice Primitives with Desired Creative Goals. Though we created an initial set of dimensions for AI-steering, we were surprised that participants already had a set of go-to primitives to express high-level creative goals, such a long notes to convey calmness or ascending notes to express triumph and escalation. When the interactive dimensions did not explicitly map to these primitives, 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] 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

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

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

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

[6] 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.

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

[8] 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.

[9] 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).

[10] 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* (Honolulu, HI USA) *(CHI '20)*. ACM, New York, NY, USA, 13. https://doi.org/10.1145/3313831.3376739

[11] 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.

[12] 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

[13] 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.

[14] 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].