A Case-Based Reasoning Approach to Plugin Parameter Selection in Vocal Audio Production

Michael Clemens[0000−0002−4507−8421], Nancy N. Blackburn[0000−0003−4371−2676], Rushit Sanghrajka[0000−0003−2368−3388], Monthir Ali, M. Gardone[0000−0002−0881−4135], Shilpa Thomas, Hunter Finney, and Rogelio E. Cardona-Rivera[0000−0002−6720−568X]

University of Utah, Salt Lake City UT 84105, USA

Abstract. The field of intelligent systems for music production aims to produce co-creative tools to aid and support musicians' decision-making while targeting a specific aesthetic in their musical artifact. While case-based reasoning (CBR) approaches have been used to assist music generation and recommendation, music production has not yet been explored. This paper proposes using CBR within a co-creative agent to assist musicians in their aesthetic goals through a vocal audio plugin. Results show that although participants were interested in using a co-creative agent throughout the production process, they acted against the vocal plugin parameter recommendations set by the agent. Participants showed frustration when the co-creative agent acted in a way that deviated from set expectations. From this research, we posit that explainability is an essential aspect of effective CBR models within co-creative agents.

Keywords: computational creativity, co-creative agents, cased-based reasoning

1 Introduction

Audio production is the manipulation and design of audio for consumer media [1]. It is a general term encompassing audio manipulation, design tasks in music production, sound recording, audio post-production, audio mixing, mastering, live sound, and sound design (e.g., for music, film, theater, games). From initial musical ideation to the final release of the artifact, musicians (i.e., instrumentalists, engineers, producers) operate complex audio tools to achieve their desired audio artifact. For example, audio engineers may select appropriate microphones, amplifiers, and digital effects based on music genres, target audience, and creative purpose [2].

Skeuomorphism, when digital implementations of instruments and effects are made to appear as their real-world counterparts, is common amongst audio processing and effects tools' interfaces. Novice users of audio processing tools, however, are likely to have never used hardware equipment such as analog mixing consoles or analog synthesizers. They are thus likely to be unfamiliar with the interaction metaphors of hardware tools [1]. Further, considerable audio-specific technical and theoretical understanding is needed to properly transfer a desired audio concept to the software via the specific tools afforded in these interfaces [3].

To assist musicians and alleviate these technological cognitive burdens, the field of intelligent systems for music production aims to propose and produce automatic tools to aid and support music actors' decision-making [4]. While case-based reasoning (CBR) has been used to study creativity quantitatively [5–7], there has been no support for qualitative metrics of CBR effectiveness within the field of musical aesthetics. Our two research questions are:

- **RQ1**: *How does CBR within a co-creative agent affect music producers' musical artifacts when evaluated on aesthetics?*

- **RQ2**: *How do music producers use adaptive plugins in a vocal production chain?*

We evaluated these research questions through a CBR implementation using an adaptive audio plugin to produce vocal samples. We also explored some music producers' apprehension regarding co-creating with an intelligent agent in the studio. The main contribution of this application paper concerns the latter exploration through an experiment and semi-structured interviews with music producers.

2 Related Works

Within intelligent audio plugin literature, two main tasks have been targeted: *parameter tuning* and recommendation. Plugin recommendation deals mainly with the selection and the arrangement of plugins within a particular audio chain. Stasis et al. [8] used Markov chains to describe plugin sequences based on timbrel descriptive phrases, intended audio effects and music genre for plugin suggestion. Despite the promising results, recommendations are only provided by four separate plugins. Their work investigates how machine learning approaches may be used to recommend audio plugin arrangements.

Moura da Silva et al. [2] address the problem of selecting audio plugins, i.e., effect implementations, as a recommendation task and employ two different methodologies: supervised learning and collaborative filtering. Parameter tuning seeks to identify the most appropriate parameter settings (e.g., gain, treble level) of a specific plugin. The approaches for selecting default parameter values (factory presets) have been based on music genre [9, 10] automatic identification of parameter values based on audio samples [11] and adaptive control of digital audio effects using either linear dynamical transformations [12] or nonlinear approaches [13, 14].

However, none of these approaches for *adjusting parameters* or choosing the sequencing of plugins have used CBR within their implementation. The following section reviews the literature in which CBR has been effective in audio domains.

2.1 CBR in Music

As a technology, CBR [6, 15, 16] works to solve problems by reusing (typically through some form of adaptation) answers to similar, previously solved problems. CBR is based on the idea that similar problems have similar solutions. CBR is suited for problems when (1) there are numerous examples of previously solved similar problems available and (2) a major part of the information involved in problem-solving is tacit, that is, difficult to explain and generalize [17]. An additional advantage of CBR is that each new solved problem can be corrected and recalled, allowing the system to enhance its problem-solving capabilities through experience. CBR has been used in the literature for audio domains in two main applications: **music generation** and **music recommendation**.

Music Generation Pereira et al. [18] explored computational approaches in music composition based on CBR and planning techniques. Their research focused on developing new solutions by retaining, modifying, and extrapolating knowledge from previously expert-created music analysis. This research was pivotal in that it sparked interest in creating musical agents using CBR.

Lopez de Mantaras et al. [19] developed SaxEx, an intelligent musical system based on CBR techniques capable of producing emotive, monophonic music resembling human performance. Their research demonstrated that user interaction within the CBR process is necessary to engage the agent and musician in a co-creative process. TempoExpress [20] is another musical CBR system that automatically performs musically acceptable tempo transformations. The research in music generation using CBR has waned as much of the work now uses neural networks, deep learning models, and ensemble techniques [21]. However, CBR for recommendation systems in music saw a later surge of interest.

Music Recommendation Since music recommender systems are based on both recommendation systems and music information retrieval (MIR), they must cope with the constraints of both domains [22]. Traditional MIR techniques employ content-based audio-related techniques, which, in addition to the general constraints of content-based systems such as overspecialization and limited diversity, necessitate a deeper understanding of the application domain [23, 24].

Gatzioura et al. [25] used a hybrid approach that employed CBR after a contextual pre-filtering process, allowing them to find the most similar previously recommended lists [26]. They found that adding contextual pre-filtering increased the recommenders' accuracy and computational performance as compared to commonly used methods in the field. Lee et al. [27] also found that their system, *C_Music*, increased in performance as well using a context-aware system that employed CBR.

Although it has not been explored in the literature, we believe CBR has the potential to be effective in tailoring the aesthetic decisions of artists producing music for a specific genre. We experimented with a Max for Live plugin using CBR and evaluated our tool through quantitative and qualitative measures to test our hypothesis.

3 Our Co-Creative Audio CBR Plugin

We developed a system to explore the potential that a CBR-powered co-creative plugin holds for audio vocal production. Rather than creating an entirely new tool, our intended users were music producers familiar with *Live*[1], a popular Digital Audio Workstation (DAW) published by Ableton. We relied on *Max for Live*—a connector that affords using the Max/MSP graphical programming language[2] within Live, as illustrated in Figs. 1 and 2.

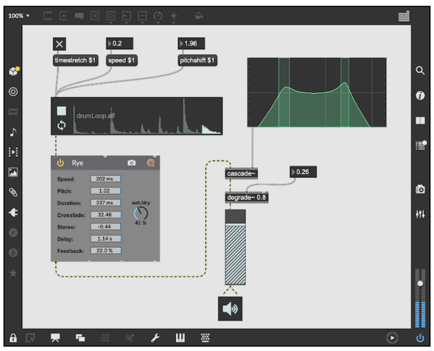

Fig. 1. Max for Live allows a user to use the Max/MSP graphical programming language to manipulate audio.

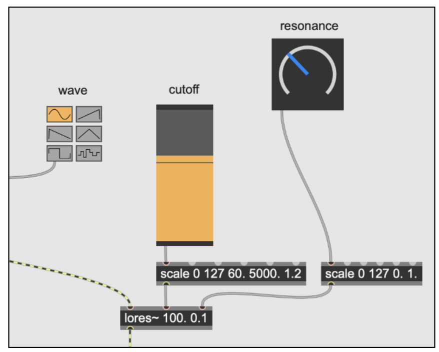

Fig. 2. Through Max for Live, users can connect audio to effects controlled via dials and sliders.

We configured our system to support the following four audio editing components published by Ableton and independent artists. The **Poundcake Compressor**[3] by artist *artsux* replicates a low-pass gate compressor that manipulates the amplitude and timbre of an input signal. The **Color Limiter**[4] by *Amazing Noises* and the *Ableton Team* manipulates loudness, ceiling, saturation, and color of an input signal. The **GMaudio Dynamic EQ**[5] by artist *groovmekanik* supports attenuating/accentuating unwanted/wanted signal frequencies. Finally, the **Convolution Reverb**[6] by the *Ableton Team* supports manipulating spatial effects of an input signal (e.g., rendering within a cathedral hall, nightclub).

3.1 Our CBR: Parameter Tuning over Audio Components

We arranged our supported audio editing components in series, as listed above—i.e. an input vocal audio signal would always flow from Compressor to Limiter, then to Dynamic EQ, and finally to Reverb. Each component has a unique set of parameters that modulate its performance over the input signal to produce

[1] https://www.ableton.com/en/live/
[2] https://cycling74.com/products/max
[3] https://www.maxforlive.com/library/device/6346/poundcake-compressor
[4] https://www.ableton.com/en/packs/creative-extensions/
[5] https://maxforlive.com/library/device/5768/gmaudio-dynamic-eq
[6] https://www.ableton.com/en/packs/convolution-reverb/

an output one. We designed our CBR algorithm around *few-shot learning* [28] to suggest individual component *parameter values* based on the input *vocal audio signal properties* recorded over several *trials* (i.e. audio editing sessions). The algorithm learns over three trials to produce parameter value suggestions for the next two. That is, the CBR algorithm uses trials 1-3 to identify values for trials 4 and 5; it uses trials 6-8 to identify values for trials 9 and 10.

Our CBR algorithm produces suggestions based on two key prosodic elements of the input vocal audio: its measured average *fundamental frequency* f_0 and corresponding *standard deviation* σ_0. To constrain the *expressive range* [29] of our plugin, we limited the input vocals available to users to a set of 10 samples collected at random from royalty-free libraries offered by Noiiz.[7] Each collected sample was processed with *Praat*, a linguistic tool for prosodic and verbal trait analysis [30]. Table 1 showcases the relevant prosodic data from each sample.

Table 1. Vocal audio sample data, analyzed using Praat. Our CBR algorithm suggests audio editing component parameters based on average *fundamental frequency* f_0 and *standard deviation* σ_0 of user-chosen input signals from the set of 10 below. For reference, we list each sample alongside its vocal range classification [31].

Sample id	**Vocal range**	**Length** (s)	$f_0 \pm \sigma_0$ (Hz)
1	Mezzo-Soprano	3.692	247.58 ± 35.76
2	Soprano	2.75	358.39 ± 29.98
3	Soprano	7.742	484.88 ± 48.99
4	Soprano	3.552	370.94 ± 47.33
5	Soprano	5.178	358.65 ± 41.44
6	Tenor	5.365	311.52 ± 54.65
7	Baritone	7.253	251.37 ± 26.95
8	Tenor	4.085	285.82 ± 33.74
9	Baritone	5.461	259.17 ± 18.54
10	Tenor	5.547	311.35 ± 20.29

3.2 Our CBR 4-Phase Cycle

As mentioned, our system learns over three trials and performs CBR for the subsequent two. During the three learning trials, the system offers CBR-based parameter recommendations *as if* the user had selected (the same) three input cases corresponding to the highest, lowest, and a (random) middle vocal audio fundamental frequency from the set in Table 1. During the subsequent two non-learning trials, the system offers CBR-based parameter recommendations based on *actual* inputs selected by the user. We describe our co-creative system's operation below in terms of the four canonical, ordered CBR phases [15].

[7] https://www.noiiz.com

Retrieve Phase. Based on the fundamental frequency (f_0) of the current vocal sample, our system finds the two closest cases (i.e., trials) and interpolates the user's selected parameter values from those values. Although the standard deviation is used in the subsequent phase, only the fundamental frequency was used during the retrieval process.

Reuse Phase. Our system takes these data and uses Equation 1 below to compute new parameter values for audio components, which are directly set by the agent as soon as the user starts the trial.

$$p = \left(p_1 + (f - f_1) \cdot \frac{p_2 - p_1}{f_2 - f_1}\right) \cdot random\left(0.1 \cdot \frac{\sigma}{\sigma_{max}}\right) \tag{1}$$

Above, p is the new parameter value, p_1 and p_2 are parameters values from the two closest cases, f is the f_0 of the current voice sample, and f_1 and f_2 are the f_0 from the two closest cases. Further, the final term in Equation 1 is meant to use the standard deviation of the vocal audio to generate some random noise such that the reused case does not reflect a direct interpolation (represented by the preceding terms). The *random* function uses the random object in Max/MSP that can output a -1 or 1 *randomly.* The noise manifests as a positive or negative update, whose absolute value does not exceed 10% of the standard deviation for all the voice samples used. Equation 1 offers much greater salience to f_0 than to σ on the resulting vocal characteristics to be reused.

Revise Phase. The user can revise recommended parameter values based on their own aesthetic goals while producing the audio.

Retain Phase. The system stores the final result of the trial in its corresponding case library, which can now be used by the system in subsequent trials.

4 Co-Creative CBR Utility and Usability Study

We conducted an Institutional Review Board-approved user study, which offered $10.00 to individual participants who were asked to use our plugin in order to produce 10 music tracks for two individual genres. We later assessed their experience via a semi-structured interview. After data collection, we evaluated the plugin via a mixed-methods analysis designed to assess the usability of our co-creative audio CBR plugin, as well as the plugin's utility for targeting a particular musical genre during vocal audio production.

4.1 Recruitment and Duration

We recruited ten music producers ($N = 10$) through convenience sampling via a local Ableton Live User Group, the Ableton Live Users Facebook Group, the Max for Live Users Facebook Group, and emails. The first author (A1) conducted the experiment and interviews remotely via Zoom. The experiment setup and trials lasted between 30 minutes to 1 hour. The semi-structured interviews lasted between 15 and 30 minutes. The interviews were audio-recorded and transcribed verbatim for later analysis.

Most participants self-identified as male ($N = 8$), live in the United Stated ($N = 9$), with the majority living in the Midwest ($N = 5$). As summarized in Table 2, all but one participant have at least one year of active music production experience, and all but two have used Ableton Live. Although not all participants were familiar with Live *for* vocal production (i.e. some use it for musical scenes, or as a non-production experimental workspace), all participants were familiar with common DAWs and their interaction patterns.

Table 2. Summary of experiment/interview participant demographics. We refer to each participant by a randomly assigned Id(entification) number prefixed with "P-".

Id	**Experience** (years)	**Role**	**Technologies Used**
P-1	10	Guitarist	Ableton Live, Soft Tube VSTs, Super8 (Reaper)
P-2	2/12	Composer	Ableton Live, FL Studio, Little Altar Boy. Logic Pro X, Novation Launchpad,
P-3	19	Guitarist	Ableton Live, Archetype: Tim Henson, Cubase, EZ Drummer, Novation Launchkey, Pitch Proof, Soldano Amp
P-4	6	Composer	Ableton Live, Cecilia, Max/MSP, Reaper
P-5	10	Singer	Ableton, Apple Drummer, FL Studio, Logic Pro X, Pro Tools
P-6	8	Producer	Ableton Live, Neutron, Ozone, Valhalla Reverbs, Waves Compressors
P-7	10	Producer	Ableton Live, EQ Eight, Max, Reaper, RX
P-8	10	Bassist	Audacity, EZ Drummer, Sonar Cakewalk
P-9	2	Guitarist	Ableton Live, FL Studio, Isotope, Logic Pro X, Ozone
P-10	1	Composer	GarageBand, Sibelius

4.2 Study Procedure

Participants were instructed to produce tracks for two dichotomous genres: acoustic for soprano-leaning vocals (tracks 1–5), and R&B for the tenor-leaning vocals (tracks 6–10). Producers had a maximum of three minutes to produce each track; therefore, the total time for the experiment was 30 minutes. Five participants were assigned to group one: producing soprano-leaning vocals followed by tenor-leaning vocals. The remaining five were assigned to group two: producing tenor-leaning vocals followed by soprano-leaning vocals. This grouping was implemented to alleviate any potential ordering effects from the different vocal ranges. Each trial's starting and ending parameter values were recorded to evaluate the effectiveness of the case-based reasoning algorithm implemented in the plugin.

4.3 Data Collection

Parameter Telemetry. Telemetry data was collected to determine the effectiveness of using the co-creative agent (Table 3). One participant's data were excluded since it was corrupted and not recoverable. Another participant's data were excluded due to the file missing most of its data. We developed a python script to extract data from JSON files and print them into a CSV file. Participants adjusted a total of 35 parameters across the plugin's four audio editing components.

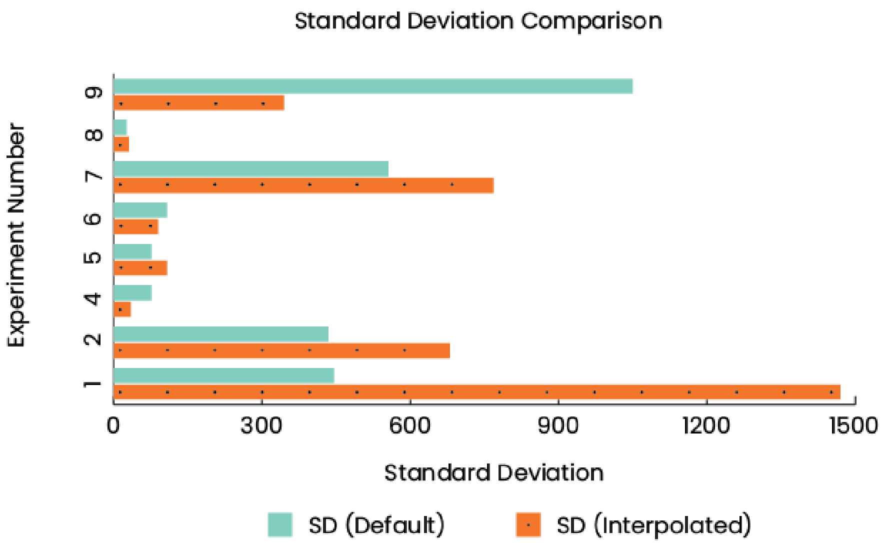

Fig. 3. Standard deviation comparison of audio component parameters for 9 participants, whom each completed 10 trials. Figure 3 shows the standard deviation of data when participants start from default values (blue bar) versus when they start from interpolated values generated by the co-creative agent (orange bar). The results show that while some participants adjusted the parameters less when starting from interpolated values, others adjusted them more. There is thus no clear trend in the data.

Interviews. The interviewer used a preset question guide (Appendix) to conduct the semi-structured interview. The guide was organized as follows:

1. Broad overview to gain insights about their overall experience and receive the plugin data.
2. Specific questions about their creative flow aimed at understanding their process of editing the tracks and the qualities of the sound they attempted to adjust.
3. Specific questions about the plugin and then about the co-creative agent: specifically asking participants whether they noticed the changes the co-creative agent made to their default settings, how they felt about the updates, and how they would have received the input if they had been aware of the co-creative agent's purpose upfront.

4. Demographics to understand their familiarity with producing music tracks, expertise with various tools, and experience with co-creative agents outside of the experiment
5. Final questions were asked to allow participants to share anything else and ensure they received their remediation.

Interview questions asked were initially vague to get a baseline: asking participants to describe their process and workflow in editing the music tracks provided to them and recording their experience. These were followed by questions asking them if they noticed any changes in some of the experiments and letting them express what they thought. In some cases, if participants seemed not to mention the updated parameters, the interviewer directly asked about it, asking them if they noticed that the default values of the parameters were different. The interviewer then let the participants talk about it: their surprise, their response as to how they handled the changed parameters, and what they thought of the default values.

The penultimate segment of the interview involved discussing what the co-creative agent actually did, and hearing feedback from the participants on their openness and thoughts on working in a music editing task where a co-creative agent would assist in the process by setting some default values for certain parameters. The final part of the interview was about collecting demographics, and their experience with various music editing software, environments and plug-ins.

Transcription. The transcription of the interviews was done with the help of Otter.ai[8], which uses machine learning for their speech-to-text operations. Files were then corrected by hand to ensure that what was transcribed reflected what was said by the speakers. If either the interviewer or interviewee was unintelligible, the transcribers made a note of that by marking the speech as "[*unintelligible*]" or something similar. Names and other personally-identifiable information in the transcriptions were redacted for the anonymity and privacy of the participants. Three transcribers worked on the interviews before being sent to the coding team.

4.4 Quantitative Analysis

Range of Adjustment (RoA). Adjustments ranged from negative and positive values for each parameter. Hence, we took the absolute value of the difference between each parameter's value at the start and end of each trial to determine the range of adjustment.

We studied the RoA for the 35 parameters in each trial before the participant was satisfied with the results and ended the trial. We recorded an average RoA value of 87.21 for soprano tracks (SD = 120.20) and 69.70 for tenor tracks (SD = 57.80). When participants started from interpolated values generated by the co-creative agent, we recorded an average RoA value of 76.24 (SD = 59.25). These

[8]https://otter.ai

values suggest two things: First, regardless of whether participants started their trials from the default values or the interpolated values generated by the co-creative agent, the difference in RoA was insignificant. Also, the range these parameter adjustments occurred within was substantial, as indicated by the large standard deviation values shown in Figure 3.

Inter-rater Reliability of Coding. The transcripts from the semi-structured interviews were thematically coded using *iterative open coding* [32]. Three of the authors generated codes iteratively and then used the final set of codes to recode previous interviews [33]. Three of the interviews were coded by all three authors to test for inter-rater reliability as reported in Table 3. Fleiss' Kappa score was calculated at 0.499, which represents significant agreement based on McHugh's deductions on Cohen's Kappa [34].

Table 3. Fleiss' kappa for observed and expected inter-rater agreement among coders.

Fleiss' Kappa	Observed Agreement	Expected Agreement
0.499	0.77	0.541

4.5 Qualitative Analysis

The qualitative analysis was aimed at understanding the participants' response to the co-creative agent, helping them with the task of producing music in the context of the plug-in that was developed in Ableton Live. Specifically, while the quantitative data collection techniques collected the metrics of how much the participants were interacting with the parameters, the interviews aimed at understanding the reasoning behind the quantitative results.

This section describes a few insights that were extrapolated from analyzing the interviews. Each interview was coded for the presence of a theme, i.e., the reported numbers signify that the participant expressed a given theme at least once over the span of the interview.

Most of the participants who were acting against the parameters were surprised when told about the co-creative agent changing the parameters (3 out of 4).

> **P-3**: "*Like so when I would hit start trial, all of a sudden it would seem like it would be louder. Um, and I think maybe the initial ones when I hit start trial were kind of the same, almost or not quite as much of a jump.*"
> **P-6**: "*Okay, I mean, because that I did notice that it was kind of, like, around the same area I was putting stuff.*"

When participants acted against the parameters, the interview process revealed that the reason for acting against the changed parameters was that they thought that the CBR agent's suggested parameters were due to a bug in the system, and they reset it or changed it due to that. Consequently, when they were told that the updated parameters were actually from a co-creative agent, most of them expressed surprise. They mentioned that they were unaware that this was intentional. Some of the participants even suggested that having the plug-in inform them explicitly about this would have been helpful in knowing that it was by design.

All participants that used the updated parameters as default values were not surprised when they were told that these parameters were set by the co-creative agent (N=4).

> **P-7**: "*Yeah, yeah, I noticed it.*"
> **P-8**: "*I'd be looking for the dry/wet [parameter] to be down, and then it was already up. It's like, 'Wait a minute.' Yeah. So and that was yeah, I noticed that was different.*"

The participants that actually used the default values of the updated parameters did not express surprise. Some of them expressed that they were aware that the plug-in set the parameters for them, while some of them did not realize that but used the default values as-is in the editing process.

Majority of participants were using the default parameters or were open to using a co-creative agent that helped set parameters for them in the editing process (N=7). When told about what the co-creative agent was doing, most participants expressed openness to the idea of a co-creative agent setting parameters for them in editing tools of the future. This information was interesting to note because it correlates to our predictions that, in theory, people would be open to working with co-creative agents in a music production setting.

> **P-1**: "*I think it would be helpful if you had a lot of tracks to go through. And you were like, you know, they could all sound the same...*"
> **P-6**: "*I think over time, like I could see it being helpful and being a good starting point.*"
> **P-7**: "*I think it's a great way to move forward and save time.*"

Majority of participants reported at least one instance where they had a negative experience during the experiment (N=9) **Roughly half the participants had issues where they were unclear about how to perform an action that they needed to perform in the editing process.** (N=6) **Most participants had opinions on the expected behavior of a sound editing plug-in with a co-creative agent in it.** (N=9) When asked how they would expect a tool to work, all participants had ideas on how they would expect the co-creative agent to work and help them in the editing process.

> **P-4**: "*You know, I really like, um, the idea of experimenting with that, and seeing where it could lead, but I think it would not be, it wouldn't be a regular part of what I do, but would be a curious circumstance that is yet another experiment, um, in a long list.*"
> **P-6**: "*I'd still want to like A/B back and forth with the raw sound and that, just make sure.*"

P-6: "*I think it's definitely something like I can't say for sure if I like if ever use it or not, to be honest, but it's definitely something I would like to mess around with a little bit.*"
P-10: "*...but I also feel like that could hinder like creativity to... to an extent because then I feel like it's kind of boxing you more into just kind of what you are drawn towards, I guess that's how it seems...*"

Users experienced frustration when the co-creative agent acted in a manner disparate from their set expectations. Most users had a set expectation that was unique to them for how a co-creative agent should behave during the creation process. Although some producers were satisfied with the recommended parameter settings, others acted against the agent. These data help demonstrate that explainable models within co-creative agents are essential for fostering effective bi-directional communication between the agent and the artist. Without a level of explainability, artists were often confused by the recommended parameter settings. Some producers were encouraged by having a co-creative agent learn from them. However, many were trepidatious of being boxed into a creative corner. Although it seemed helpful that the agent learned from an individual's style, the worry is that total mimicry would stifle the artist's ability to create unique productions.

5 Conclusion and Future Works

In this work, we evaluated a case-based reasoning approach to vocal audio production using a Max for Live plugin in Ableton Live. The implemented system recommends parameter values tailored towards the producer's musical aesthetic choices for producing vocals within a particular genre. This work provides evidence to suggest that while producers are interested in working with co-creative agents in the studio setting, they act negatively towards an agent's recommendations without sufficient explanation from the co-creative agent for those creative choices. These negative actions occurred most frequently when the agent acted in a manner that deviated from expectations held by the music producers. Without a way for the agent to communicate with the producer during the co-creative process, producers relied solely on held expectations and interpreted recommended parameters from that standpoint. From these data, we posit that explainability is essential to effective co-creative agents. The agent must articulate and provide sufficient rationale for its creative choices to be valuable to music producers.

A potential limitation of this work was the number of cases the CBR agent had access to when recommending audio plugin parameter values. Although few-shot learning can be effective in many domains, this assumption may have been detrimental to the results of our research. Another potential limitation of this work may be the lack of inquiry regarding the agent's solution. Did the producers reject a solution based on the interpretability of the solution or the solution's quality? Future work could explore the explainable co-creative space to make agents better suited for co-creation by exposing the agent's creative process.

6 Appendix: Interview Questions Guide

6.1 Overall Experience

- Please upload your output file (download and test this, please)
- Would you describe your workflow while completing the experiment?
- How was your overall experience with the plugin?

6.2 Experiment Questions (Plugin/Co-Creative Agent)

- How did you feel about the different tracks you worked with?
- Were you aware of the updated presets that occurred after the third track? (IF NOT: explain the process, then ask these questions:)
- How did it affect your workflow? (easier, quicker, recommendation, co-creative)
- In what ways did you find the plugin helpful?
- In what ways was the plugin a hindrance?
- What was your favorite part about the plugin's design?
- What was your least favorite part about the plugin's design?
- What would you change about the plugin design if you could? (optional)
- Would you add or remove any pieces of the plugin for producing a vocal mix? If yes, why?

6.3 Demographics

- How long have you been producing music?
- What is your favorite Digital Audio Workstation (DAW) to produce with?
- How familiar are you with Ableton Live? How long have you used it?
- Have you used AI-based mastering tools such as Landr, Dolby.io, or SoundCloud before in your mixing/mastering process?
- Have you used machine learning (ML) tools such as Magenta in your creative process?
- Do you have any plugins that you use regularly and why?
- Do you tend to use stock presets when using plugins, or do you modify them and create your own?
- Have you used an adaptive plugin before? If so, which one(s)?
- How was your experience with them?

6.4 Wrap-Up

- Is there anything I covered that you would like to revisit or anything that I missed that you would like to add?
- What is your address for your gift card?
- Those are all the questions we have for you. Thanks for your participation.

References

1. M. Cartwright, "Supporting novice communication of audio concepts for audio production tools," Ph.D. dissertation, Northwestern University, 2016.
2. P. M. M. da Silva, C. L. C. Mattos, and A. H. de Souza Júnior, "Audio plugin recommendation systems for music production," in *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2019, pp. 854–859.
3. P. Théberge *et al.*, *Any sound you can imagine: Making music/consuming technology*. Wesleyan University Press, 1997.
4. J. D. Reiss, "Intelligent systems for mixing multichannel audio," in *2011 17th International Conference on Digital Signal Processing (DSP)*. IEEE, 2011, pp. 1–6.
5. S. R. Turner, "A case-based model of creativity," in *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, 1991, pp. 933–937.
6. J. L. Kolodner, "Understanding creativity: A case-based approach," in *European Workshop on case-based reasoning*. Springer, 1993, pp. 1–20.
7. L. M. Wills and J. L. Kolodner, "Towards more creative case-based design systems," in *AAAI*, vol. 94, 1994, pp. 50–55.
8. S. Stasis, N. Jillings, S. Enderby, and R. Stables, "Audio processing chain recommendation," in *Proceedings of the 20th International Conference on Digital Audio Effects,(Edinburgh, UK)*, 2017.
9. V. Goudard and R. Muller, "Real-time audio plugin architectures," *Comparative study. IRCAM-Centre Pompidou. France*, 2003.
10. D. Robillard, "Lv2 atoms: A data model for real-time audio plugins," in *Proceedings of the Linux Audio Conference (LAC-2014)*, 2014.
11. N. Peters, J. Choi, and H. Lei, "Matching artificial reverb settings to unknown room recordings: A recommendation system for reverb plugins," in *Audio Engineering Society Convention 133*. Audio Engineering Society, 2012.
12. A. M. Stark, M. E. Davies, and M. D. Plumbley, "Rhythmic analysis for real-time audio effects," in *International Computer Music Conference, ICMC 2008*. University of Surrey, 2008.
13. E. T. Chourdakis and J. D. Reiss, "A machine-learning approach to application of intelligent artificial reverberation," *Journal of the Audio Engineering Society*, vol. 65, no. 1/2, pp. 56–65, 2017.
14. M. A. M. Ramírez and J. D. Reiss, "End-to-end equalization with convolutional neural networks," in *21st International Conference on Digital Audio Effects (DAFx-18)*, 2018.
15. A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI communications*, vol. 7, no. 1, pp. 39–59, 1994.
16. D. B. Leake, "Case-based reasoning: experiences, lessons, and future directions," 1996.
17. R. L. De Mantaras and J. L. Arcos, "Ai and music: From composition to expressive performance," *AI magazine*, vol. 23, no. 3, pp. 43–43, 2002.
18. F. C. Pereira, C. F. A. Grilo, L. Macedo, and F. A. B. Cardoso, "Composing music with case-based reasoning," in *International Conference on Computational Models of Creative Cognition*, 1997.
19. J. L. Arcos, R. L. De Mántaras, and X. Serra, "Saxex: A case-based reasoning system for generating expressive musical performances," *Journal of New Music Research*, vol. 27, no. 3, pp. 194–210, 1998.

20. M. Grachten, J. L. Arcos, and R. López de Mántaras, "Tempoexpress, a cbr approach to musical tempo transformations," in *European Conference on Case-Based Reasoning*. Springer, 2004, pp. 601–615.
21. P. M. Bodily and D. Ventura, "Musical metacreation: past, present, and future," in *Proceedings of the sixth international workshop on musical metacreation*, 2018.
22. Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbull, "Music emotion recognition: A state of the art review," in *Proc. ismir*, vol. 86, 2010, pp. 937–952.
23. M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.
24. B. Shao, D. Wang, T. Li, and M. Ogihara, "Music recommendation based on acoustic features and user access patterns," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 8, pp. 1602–1611, 2009.
25. A. Gatzioura, M. Sànchez-Marrè *et al.*, "Using contextual information in music playlist recommendations." in *CCIA*, 2017, pp. 239–244.
26. R. L. De Mantaras and E. Plaza, "Case-based reasoning: an overview," *AI communications*, vol. 10, no. 1, pp. 21–29, 1997.
27. J. S. Lee and J. C. Lee, "Context awareness by case-based reasoning in a music recommendation system," in *International symposium on ubiquitious computing systems*. Springer, 2007, pp. 45–58.
28. Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.
29. G. Smith and J. Whitehead, "Analyzing the expressive range of a level generator," in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, 2010, pp. 1–7.
30. P. Boersma and D. Weenink, "Praat: Doing phonetics by computer (6.0. 18)[computer software]," 2019.
31. J. C. McKinney, *The Diagnosis and Correction of Vocal Faults: A Manual for Teachers of Singing and for Choir Directors*. Waveland Press, 2005.
32. A. Strauss and J. Corbin, "Basics of qualitative research techniques," 1998.
33. D. Votipka, R. Stevens, E. Redmiles, J. Hu, and M. Mazurek, "Hackers vs. testers: A comparison of software vulnerability discovery processes," in *2018 IEEE Symposium on Security and Privacy (SP)*, May 2018, p. 374–391.
34. M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.