Threshold Designer Adaptation: Improved Adaptation for Designers in Co-creative Systems

Emily Halina and **Matthew Guzdial**

Department of Computing Science, Alberta Machine Intelligence Institute (Amii)
University of Alberta, Edmonton, Alberta, Canada
{ehalina, guzdial}@ualberta.ca

Abstract

To best assist human designers with different styles, Machine Learning (ML) systems need to be able to adapt to them. However, there has been relatively little prior work on how and when to best adapt an ML system to a co-designer. In this paper we present threshold designer adaptation: a novel method for adapting a creative ML model to an individual designer. We evaluate our approach with a human subject study using a co-creative rhythm game design tool. We find that designers prefer our proposed method and produce higher quality content in comparison to an existing baseline.

1 Introduction

The study of creative Artificial Intelligence (AI) has increased in recent years [Veale and Cardoso, 2019]. Creative AI agents have been proposed and developed for many domains, including music, painting, and robotics [Carnovalini and Rodà, 2020; Yalçın *et al.*, 2020; Fitzgerald *et al.*, 2017]. These advancements have also led to the development of systems that allow human designers to interact directly with AI agents co-creatively [Guzdial *et al.*, 2018]. Many of these co-creative AI systems utilize modern Machine Learning (ML) methods due to the recent strides ML methods have made in a number of creative fields [Franceschelli and Musolesi, 2021].

A major challenge in designing ML models for co-creative systems is "adaptability," the notion of changing a model's behaviour to fit a given designer's preferences. Particularly within artistic domains, there is a need for co-creative systems to adapt to an individual designer's artistic sensibilities. For example, two artists who interact with the same co-creative art tool may have vastly disparate styles such that an ML model could not match both of them without adaptation [Gatys *et al.*, 2015]. However, it is still not fully understood how and when it is best to adapt an ML model to a designer within a co-creative context. Further understanding this problem of designer adaptation would allow us to bolster the capabilities of these systems in a variety of domains.

In this paper we explore the problem of designer adaptation within the domain of game design. AI has long been used to aid game developers with both game design and asset creation tasks [Yannakakis and Togelius, 2018]. In particular, the application of AI methods to generate game content is known as Procedural Content Generation (PCG) [Hendrikx *et al.*, 2013]. Several co-creative PCG systems—with and without the use of ML—have been developed in prior work [Guzdial *et al.*, 2018; Charity *et al.*, 2020]. The application of ML to co-creative PCG is relatively novel, and as such there has not been deep prior investigation into adaptation for human designers [Guzdial *et al.*, 2019].

Towards the goal of effectively adapting to individual designers, we propose "threshold designer adaptation," a novel method for adapting creative ML models to individual designer input. This strategy revolves around determining a threshold of training data to retrain at by approximating human interaction using existing data. We hypothesize that this strategy will help models adapt to individual designers in a meaningful way, particularly in domains that require internal consistency and structure, such as painting and music. This is due to the model retraining only after there is enough context for the human-authored data to be meaningful.

In order to investigate the problem of designer adaptation, we designed and developed KiaiTime [Halina and Guzdial, 2021a], an adaptive co-creative PCG system for the rhythm game Taiko no Tatsujin (Taiko) [TPGPL *et al.*, 2020]. The purpose of this tool is to help a designer create Taiko charts: a series of pre-determined game objects (notes) that are hit by the player to the beat of a song. An example of a chart is shown in Figure 1. Given that expert Taiko designers regularly collaborate in chart generation, we contend this domain is well-suited for studying co-creation and designer adaptation. KiaiTime uses an ML model to determine where to place notes based on both the song's audio and previously placed notes. This ML model is set up to test different adaptation strategies, with KiaiTime collecting training data from individual designers as they interact with the system.

To evaluate the effectiveness of threshold designer adaptation, we conducted a human subject study using the KiaiTime system. We found that designers preferred interacting with the model using threshold designer adaptation over a baseline approach. We also found that designers created higher quality charts when interacting with a model using threshold designer adaptation.

The main contributions of this work to the broader creative AI community are as follows:

- Threshold designer adaptation, a novel method for adapting co-creative ML systems to designers.

- A discussion and system overview of KiaiTime, a co-creative AI system for studying designer adaptation.

- A detailed account of the methodology, design, and results of a human subject study used to evaluate a co-creative AI system.

2 Related Work

2.1 AI Co-creation

Human-AI co-creation systems are systems in which a human designer and AI agent work together on a task [Davis, 2013]. Many co-creative AI systems generate content based on a given "prompt" from a human user [Galanos *et al.*, 2021; Gero and Chilton, 2019; Champandard, 2016]. However, these systems only allow for minimal collaboration and adaptation, as the human only influences the input given to an unchanging AI model. In comparison, we focus on iterative, long-term interactions that change the underlying model's predictions to appropriately suit a user on arbitrary input. Other co-creative AI systems act as enhanced content "editors" that supplement human creators in various ways, falling under the broader category of creativity support tools [Frich *et al.*, 2019]. An example is MakeWrite, a co-creative system for helping users with communication disorders write via the use of automatic constraints [Neate *et al.*, 2019]. These systems focus on a human user adapting their behaviour toward a desired outcome rather than an AI adapting to the user.

Our framework is a mixed-initiative co-creative system, which is a type of system in which both the AI agent and human user "take the initiative" contributing to a collective artifact [Liapis *et al.*, 2016]. A distinguishing quality of mixed-initiative systems is the shared agency between the human and AI agent [Zhu *et al.*, 2018]. However, existing mixed-initiative AI systems generally do not retrain at all, leaving the AI partner to be "stubborn" and unchanging.

2.2 Games

Co-creative frameworks utilizing PCG techniques have been developed for many game domains. The majority of prior mixed-initiative and co-creative PCG approaches do not utilize ML techniques, and instead rely on other non-ML AI approaches [Sturtevant *et al.*, 2020; Charity *et al.*, 2020; Liapis *et al.*, 2013]. While some of these tools do adapt to designers, they do so only in a human-authored and predetermined way, limited to their specific game domain. We instead investigate general adaptation strategies.

Procedural Content Generation via Machine Learning (PCGML) refers to the subsection of PCG approaches utilizing ML. There are still many open problems to be explored within co-creative PCGML systems, one of which is designer adaptation [Summerville *et al.*, 2018]. Due to intrinsic challenges with ML approaches such as lack of training data for individual users, the majority of prior co-creative PCGML systems for games do not include designer adaptation. The singular example of an adaptive co-creative PCGML system evaluated with a human subject study is Guzdial et.

al.'s "Morai Maker" [Guzdial *et al.*, 2019], a mixed-initiative PCGML editor for Super Mario Bros.-style levels. Within the Morai Maker editor, the AI agent actively trains based on implicit user feedback that takes into account whether the AI's suggestions were kept or deleted by the human partner. This training methodology informed the "naive" training strategy which was used as a baseline in our human subject study. To the best of our knowledge, our system KiaiTime is the first co-creative tool for chart generation.

3 Threshold Designer Adaptation

Our primary contribution is our threshold designer adaptation approach. Threshold designer adaptation is a novel approach for adapting creative ML models to individual designers within co-creative tools. This approach relies on determining a threshold δ of training data needed to begin retraining, thus the name. To determine the value of δ we rely on a simulation of co-creative interaction with existing human data. In this way, we can approximate how different values of δ will perform when a human designer interacts with the AI.

There is high variability in the amount of retraining data needed to make a meaningful change to an ML model's output depending on architecture and domain. As such, we propose δ be determined empirically by employing existing human data to find the best value to retrain at. In order to do this, we mimic the co-creative process as closely as possible, using a collection of existing human data created by different designers. For each designer, the initial ML model is retrained on the first δ training instances created by that designer for a range of possible δ values. After this retraining, we measure the model's performance on the remaining training instances created by that designer. This is done by measuring how well the model matches the designer's unseen training instances via a domain-dependant process, note accuracy in our case. Values of δ that perform well theoretically produce models which are further adapted to the individual designer using them. Given the lack of prior co-creative PCGML work that features designer adaptation, to the best of our knowledge this methodology for determining "when to adapt" to a given designer in a co-creative context is novel.

At runtime, we begin with a co-creative system using an ML model that has been pretrained on training data that reflects the output of many different designers. As an individual designer interacts with the co-creative system, the system collects "retraining data" D_r based on the input of that designer. The form of this input varies depending on application, and in our case consists of segments of designer-created Taiko chart. After sufficient retraining data is collected, the ML model is retrained in order to tweak its output to accurately match the designer using the tool.

Whenever a designer's input creates a new instance of training data (x, t), the system adds it to the set of retraining data D_r. When this occurs, the following inequality is evaluated:

$$|D_r = \{x^{(m)}, t^{(m)}\}_{m=1}^M| \geq \delta(k+1)$$

If the total number of retraining samples M is greater than the predefined retraining threshold δ multiplied by the number of
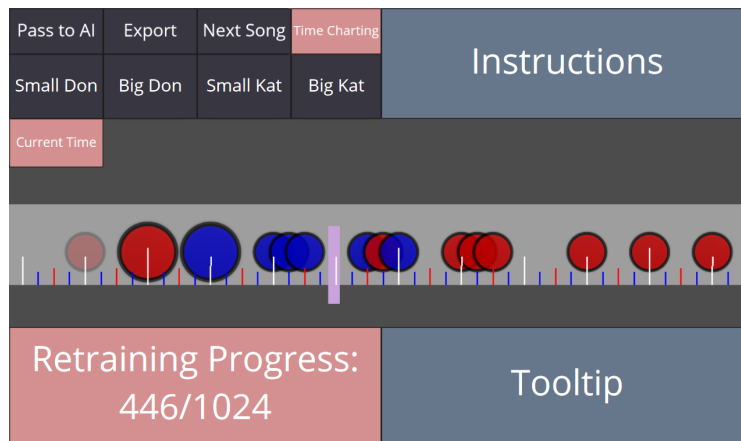
Figure 1: Screenshot of the final KiaiTime editor. Users place notes on the grey timeline in the center of the screen to create a rhythm game chart along with the AI partner. Users move the timeline in real time to the beat of the song they are charting.

times we have retrained k, the system retrains the ML model. Each time the system retrains the model, k increments.

4 KiaiTime

In this section we provide an overview of KiaiTime, a co-creative system for rhythm game chart generation. KiaiTime utilizes threshold designer adaptation to adapt its ML model to act as a partner for human designers. Section 4.1 reviews the details of KiaiTime's ML model, along with our methodology for determining the value of the retraining threshold δ for our specific task.

The KiaiTime editor is a co-creative PCGML system we used as a testbed for different designer adaptation techniques [Halina and Guzdial, 2021a]. KiaiTime's name alludes to "kiai time," a mechanic in Taiko and other rhythm games which provides a visual highlight to the most intense parts of songs. A video showcasing the functionality of the KiaiTime editor is included within the supplementary materials,[1] along with further descriptive detail regarding the editor.

Figure 1 depicts a screenshot of the final KiaiTime editor. In the editor, human users collaborate with an AI partner in a turn-based structure to create a Taiko chart for a pre-selected song. Taiko is a rhythm game where players attempt to hit a series of pre-determined game objects (notes) in time to the beat of a song. KiaiTime allows the user to create this series of notes, which are called "charts." These charts are later playable in real time.

Designers have access to the four different types of Taiko notes, each representing a different button press or "drum hit." Designers place notes on the timeline using left mouse clicks, and remove notes using right mouse clicks. The multi-coloured ticks on the timeline each represent a 1/16th beat of a given song, with placed notes automatically snapping to these ticks to match the music. These ticks are coded by size and colour, representing the beat division of different measures. This type of colour-coded beat division is standard within other Taiko editors [TPGPL *et al.*, 2020].

---

[1]For supplementary materials and source code, see https://github.com/emily-halina/KiaiTime

At any point the user may press the "Pass to AI" button to begin the process of ending their turn. Once the button is pressed, the user selects a region for the AI partner to "fill in," querying the ML model for its predictions across the given region. The ML model uses the audio and chart data around the selected region to make its predictions, outputting a series of notes that the editor places in the chosen region.

4.1 Threshold Designer Adaptation Implementation

KiaiTime uses the pre-existing "TaikoNation" system for generating the AI partner's predictions [Halina and Guzdial, 2021b]. TaikoNation utilizes a Long Short Term Memory Recurrent Neural Network (LSTM RNN) to generate predictions of where to place notes in a Taiko chart based on both the song being charted as well as previously placed notes. The model is pretrained on 100 high-quality human-authored charts from a variety of designers, and is retrained on the fly when used in KiaiTime to match the style of individual designers. These 100 charts are split into 23 millisecond segments, resulting in approximately 1.3 million training samples. This model was chosen over other rhythm game chart generation ML models [Donahue *et al.*, 2017; Lin *et al.*, 2019] due to TaikoNation's improved notion of human-like "patterning," the placement of game objects in a congruent fashion in terms of both rhythm and chart context. The full technical details regarding TaikoNation's architecture and training data can be found within the corresponding TaikoNation paper [Halina and Guzdial, 2021b].

In order for KiaiTime to best adapt to human designers using threshold designer adaptation, we needed to determine both how and when to adapt the pretrained TaikoNation model to an individual user. This requires us to determine both the retraining threshold δ described in Section 3, as well as other hyperparameters which affect the model's learning process. To do this, we utilized a grid search approach, exhaustively sweeping over a manually defined space of hyperparameters in order to find the most effective combination [Feurer and Hutter, 2019]. The hyperparameters we tuned in addition to the retraining threshold δ were learning rate α,

maximum number of training epochs ϵ, and batch size b. For each combination of hyperparameters, we trained the model on the first δ training instances extracted from a chart, then queried the model to complete the rest. The model's output was judged both on the diversity of patterns used within a chart, as well as how closely the model was able to match what the human author actually created. We repeated this process for three charts made by three different designers, attempting to recreate the co-creative environment as closely as possible. We chose these three charts due to the disparate artistic sensibilities of their respective designers. The system converged to a consistent δ value for all three designers, and so we felt no need to expand to additional designers. For each hyperparameter, we selected a small set of prospective values to test with the goal of shrinking the search space and preventing overfitting. The choices for the retraining threshold δ correspond to $10\%, 20\%, 40\%$, and 80% of a 2-minute chart respectively.

Through the grid search, we found the following set of hyperparameters that were used for retraining TaikoNation.

$$\{\delta = 1024 \ (20\%), \ \alpha = 10^{-3}, \ \epsilon = 5, \ b = 4\}$$

This set of parameters performed well on predetermined metrics from prior work [Halina and Guzdial, 2021b] while also producing impressive qualitative results from our perspective. For consistency, we used these hyperparameters for both threshold designer adaptation and our baseline approach discussed in Section 5.1.

5 Human Subject Study

In order to evaluate the performance of threshold designer adaptation, we conducted a human subject study using KiaiTime. During our study design, we identified 2 major objectives which informed our study methodology and our interpretation of the study's results. These objectives are as follows:

- **(O1):** To identify whether threshold designer adaptation helps to adapt ML models to designers.

- **(O2):** To further understand the open problem of adapting to designers within a co-creative context.

O1 arose from the desire to evaluate the value of threshold designer adaptation as an approach. O2 arose from the desire to inform future research into co-creative adaptation.

In order to complete these objectives, we conducted a within-subject study with 20 human designers, comparing threshold designer adaptation to an existing naive baseline approach defined in Section 5.1. We chose a within-subject structure rather than a between-subject structure due to the vast differences between individual designers. By comparing the results of each designer with each adaptation strategy, we can evaluate which approach helps our ML model adapt best to individual designers.

Prior to conducting the study, we identified the following two hypotheses:

- **(H1):** Threshold designer adaptation will help designers create better charts than the baseline naive approach.

- **(H2):** Designers will prefer threshold designer adaptation to the baseline naive approach.

5.1 Naive Adaptation Strategy

Our baseline adaptation strategy, which we refer to as the "naive" strategy, entails retraining the model whenever new training data is generated by the human user. In contrast to threshold designer adaptation, whenever a new piece of data (x, t) is added to the retraining dataset D_r, the ML model is retrained using all of the data in D_r. This naive strategy is analogous to the implicit training strategy used in the Morai Maker system [Guzdial et al., 2019]. We chose the adaptation strategy from the Morai Maker system as, despite the difference in genre, it is the only other adaptive PCGML tool for a game design task. Thus, this naive approach represents the current state of the art for this task. This training is done with the same set of empirically determined hyperparameters used by our threshold designer adaptation strategy. We also use the same TaikoNation pre-trained model as the base model, which is retrained with individual designer's data [Halina and Guzdial, 2021b]. We made these decisions to allow for consistency in the comparison between the two strategies.

5.2 Study Methodology

In this subsection we briefly cover our study methodology in the order a subject would experience it. Our study design was reviewed and approved by our university ethics board (Pro00112032). For the full details of our study methodology, please refer to the supplementary materials.

Participants reached out to research personnel via email to book a 1 hour timeslot after seeing publicly posted advertisements for the remote study. After acknowledging a consent form, during their timeslot participants downloaded a copy of the KiaiTime editor to their local machine. Throughout the study this editor interfaced with a server run by research personnel which handled the adaptation and contributions of the ML model.

After a brief tutorial explaining the game domain of Taiko and the functionality of the editor and AI partner, participants were tasked with creating two charts for two randomly ordered songs. These songs were *polygon* by Sota Fujimori and *citrus* by kamome sano, two electronic dance music tracks from rhythm game soundtracks with similar rhythmic structure and length. While creating each chart, participants had access to the ML model employing either the naive or threshold designer adaptation strategy chosen at random. When participants moved on to their second song, the ML model was reset to the default model, and the adaptation strategy was changed to the unused strategy. Both the ordering of the songs and ordering of adaptation strategies were randomized using a study ID system in order to ensure an even distribution of conditions.

We collected information from participants in three ways. The first was through a survey which asked participants to rank the two approaches in qualitative categories, along with asking demographic questions. The ranking questions asked participants which ML model learned better from them, was more frustrating to use, created higher quality contributions, and which they enjoyed most. The survey is based on prior co-creative system surveys, and the full text is provided within the supplementary materials [Guzdial et al., 2019]. The second was through a logging file which tracked the

Approach	Avg Time Spent	Avg End Turn Count	Human Notes Kept	AI Notes Kept	Patterns Used
Naive	20.6 mins	39.5	74.2%	59.6%	18.83%
Threshold	24.1 mins	48.6	72.1%	***72.5%**	***21.72%**

Table 1: Quantitative information extracted from participant logs and created charts. Average Time Spent is in minutes, Average End Turn Count is the average number of times the "Pass to AI" button was pressed. Human Notes Kept and AI Notes Kept are the average percentages of notes placed by each respective agent that were not deleted. Patterns Used is the average percentage of possible patterns used in charts created with each respective adaptation strategy. Statistically significant results ($p < 0.05$) are denoted with an asterisk (*).

Approach	Learned Better	Most Frustrating	Higher Quality	Enjoyed Most
Naive	8	10	8	6
Threshold	11	9	9	***11**

Table 2: Quantitative survey results for the naive and threshold adaptation strategies across all participants. Each ranking question was optional, and could be skipped if a participant had no strong preference. Statistically significant results ($p < 0.05$) are denoted with an asterisk (*)
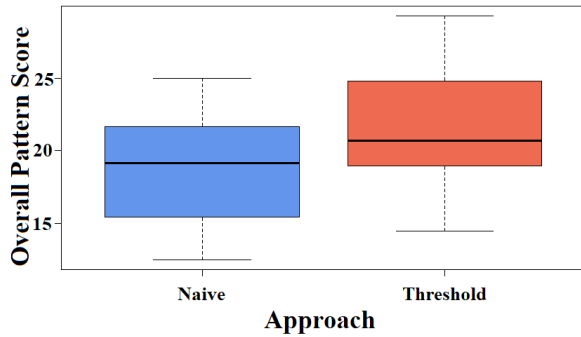

Figure 2: Boxplot depicting the overall pattern scores of charts produced with the naive and threshold adaptation strategies.

time participants spent in the editor, the amount of times they passed control over to the AI partner, and the percentage of notes they kept from both the AI and their own contributions. The third was from the charts the participants produced when working with each adaptation strategy, which we analyzed using a pre-established metric for Taiko charting quality [Halina and Guzdial, 2021b].

5.3 Study Results

In this subsection we discuss the results of our study, and provide an analysis of the quantitative data. 20 participants completed our study and survey. Of these participants, 17 identified as male, 2 identified as female, and 1 chose to not disclose their gender. 18 participants placed themselves in the 18-24 age range, with 2 participants placing themselves in the 25-30 age range. Half the participants were "experts," meaning they have authored high-quality charts according to community standards, and the other half were "novices" [TPGPL *et al.*, 2020]. This population does not have sufficient diversity to draw broad, universal lessons from this study. However, this population is consistent with the general demographic of chart creators, and thus fits our needs for the initial evaluation of threshold designer adaptation.

For each of our demographic categories (age, gender, experience level), we ran a multivariable analysis of variance (MANOVA) test with the survey ranking results and logged data as dependent variables. We found no significant correlation between any of the demographic categories and the study results. We also checked if either the ordering of the songs or the order of the adaptation strategies made a significant impact on the results. We found neither variable had a significant impact on the results, and thus we can safely treat our data as all coming from the same condition.

Table 1 shows the quantitative information extracted from each participant's logs and charts. We focus on four different pieces of information drawn from the logs divided by adaptation approach: the average time spent charting with the AI partner, the average number of times participants gave the AI partner a turn, and the percentages of notes placed by the human and AI that were kept by the end of the charting process. We deemed these statistics to be important as they represent the amount of interaction between the participants and AI partners, as well as the quality of that interaction. Similar analysis of logged files has been used in prior work [Guzdial *et al.*, 2019]. We found that participants spent more time interacting with the threshold partner, gave the threshold partner a turn more frequently, and kept substantially more notes from the threshold partner in comparison to the naive partner. To evaluate if there was a significant difference in the distributions of each of these statistics, we performed a Mann-Whitney Wilcoxon test. We chose this test because we found with the Shapiro-Wilk normality test that our data did not follow a normal distribution ($p < 0.0005$). With these tests, we found there was a statistically significant difference between the ratio of AI partner notes kept from the naive and threshold strategies with $p < 0.03$. This suggests that participants were more accepting of the AI partner's contributions when it was using threshold designer adaptation, and may imply the contributions are of a higher quality than those from the naive strategy.

To further support H1, we analyzed the quality of charts created with both adaptation strategies. We examined the proportion of possible patterns used in charts created with

each adaptation strategy, called Overall Pattern Score. Patterns are defined as unique rhythmic sequences which are 8 timesteps (184ms) in length. This is a suitable metric for estimating chart quality empirically, as higher quality charts tend to use a wider variety of patterns [Halina and Guzdial, 2021b; 5argon, 2018]. Figure 2 depicts a boxplot of the Overall Pattern Scores for charts created using each adaptation strategy. After ensuring that the Overall Pattern Score for both the naive and threshold charts followed a normal distribution with the Shapiro-Wilk normality test (with significance threshold $p < 0.05$), we performed a paired t-test on the two distributions. With this test we found the charts created with the threshold partner used significantly more patterns than the charts created with the naive partner with $p < 0.05$, 95% CI $(-\infty, -0.08)$. These logged results support our hypothesis (H1) that threshold designer adaptation would help designers create better charts than the baseline naive approach. While further investigation is required, we would encourage other human-AI co-creative systems to adapt with the threshold designer adaptation approach.

Table 2 contains the results of the survey ranking questions across all participants divided by adaptation strategy. We found an overall positive preference towards the threshold strategy for each question. To see if there was a significant difference in distribution within any of the survey questions, we performed a Mann-Whitney Wilcoxon test on each of the experiential features. We chose this test because this data does not follow a normal distribution. With this test, we found a significant difference between the two adaptation strategies in the "enjoyed most" ranking with $p < 0.05$. This result supports our hypothesis H2 that designers would prefer threshold designer adaptation over the baseline naive approach.

6 Discussion

In this section we reflect on our study, addressing the limitations of our study design and technical approach, and discuss potential avenues for future work.

6.1 Limitations

Our major evaluation limitation is the use of only a single creative domain. In an ideal evaluation, we would test the approach across multiple creative domains to assess its performance. We believe our study was sufficient for our initial proposal of the method, leaving the application to other domains as future work.

Another limitation is the lack of a strong assurance that our chosen threshold value δ is the "correct" choice. With additional resources, we would be able to sweep over a larger number of threshold values, narrowing down to an exact correct choice using a larger set of individual designers. However, even with only the three charts we used to determine the value of δ, our results suggest our chosen value was at least a reasonable choice.

A potential concern is that retraining in general is a detriment to the ML model's performance. However, we found during our grid search that models trained using threshold designer adaptation outperformed the baseline TaikoNation [Halina and Guzdial, 2021b] model in recreating a portion of

a human chart. Nevertheless, it is important to consider this possibility in future work on designer adaptation.

6.2 Future Work

Threshold designer adaptation has the potential to be applied in many domains for the purpose of co-creativity. This strategy can be used for any arbitrary set of hyperparameters and metrics for evaluation that are chosen to suit a specific domain. While we offer an improved strategy for designer adaptation over the prior state of the art approach, there is still no empirically defined "best practice" for adapting to users. The problem of designer adaptation is very domain specific, and thus in order to make progress towards finding this "best practice," it must be evaluated with more co-creative systems in different creative domains. However, due to the lack of adaptation research and the quality of our results, we view our threshold designer adaptation as the new state of the art for the task of designer adaptation. One additional avenue for future work is the usage of adaptive co-creative agents on a longer term basis with individual users. This could entail looking at adaptation strategies that could be applied over weeks or months, examining how creative ML models can be fine-tuned to fit a particular designer over a longer time span. This could be useful for artistic practices that may benefit from long-term context of a designer's tastes, such as music composition or painting.

As with any artificial intelligence application, it is important to acknowledge the potential risk of the application taking away the livelihood of people or discouraging human creation. KiaiTime and the threshold designer adaptation strategy are designed with co-creation in mind rather than as a replacement for a human designer. The systems described in this paper are not a replacement for human designers, but rather function as tools for human designers to use to assist their creative process and inspire novel designs. We will continue to reflect on these potential risks when developing future versions of the tool and approach.

7 Conclusions

In this paper we presented threshold designer adaptation, a novel method of adapting to individual designers in a co-creative context. We ran a human subject study investigating two strategies for designer adaptation using KiaiTime, the first co-creative PCGML system for rhythm game chart design. We found that designers made higher quality charts when interacting with an ML model using threshold designer adaptation over a baseline approach. We also found designers preferred working with the model using threshold designer adaptation. We hope our approach contributes to the future of designer adaptation in co-creative systems and allows ML models to better fit individual designers.

Acknowledgements

References

[5argon, 2018] 5argon. Game design and notecharting. https://exceed7.com/native-audio/rhythm-game-crash-course/game-design-and-notecharting.html, 2018. Accessed: 2022-05-17.

[Carnovalini and Rodà, 2020] Filippo Carnovalini and Antonio Rodà. Computational creativity and music generation systems: An introduction to the state of the art. *Frontiers in Artificial Intelligence*, 3:14, 2020.

[Champandard, 2016] Alex J Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv preprint arXiv:1603.01768*, 2016.

[Charity *et al.*, 2020] Megan Charity, Ahmed Khalifa, and Julian Togelius. Baba is y'all: Collaborative mixed-initiative level design. In *2020 IEEE Conference on Games (CoG)*, pages 542–549. IEEE, 2020.

[Davis, 2013] Nicholas Mark Davis. Human-computer co-creativity: Blending human and computational creativity. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013.

[Donahue *et al.*, 2017] Chris Donahue, Zachary C Lipton, and Julian McAuley. Dance dance convolution. In *International conference on machine learning*, pages 1039–1048. PMLR, 2017.

[Feurer and Hutter, 2019] Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated machine learning*, pages 3–33. Springer, Cham, 2019.

[Fitzgerald *et al.*, 2017] Tesca Fitzgerald, Ashok K Goel, and Andrea Thomaz. Human-robot co-creativity: Task transfer on a spectrum of similarity. In *ICCC*, pages 104–111, 2017.

[Franceschelli and Musolesi, 2021] Giorgio Franceschelli and Mirco Musolesi. Creativity and machine learning: A survey. *arXiv preprint arXiv:2104.02726*, 2021.

[Frich *et al.*, 2019] Jonas Frich, Lindsay MacDonald Vermeulen, Christian Remy, Michael Mose Biskjaer, and Peter Dalsgaard. Mapping the landscape of creativity support tools in hci. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2019.

[Galanos *et al.*, 2021] Theodoros Galanos, Antonios Liapis, and Georgios N Yannakakis. Affectgan: Affect-based generative art driven by semantics. In *2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 01–07. IEEE, 2021.

[Gatys *et al.*, 2015] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[Gero and Chilton, 2019] Katy Ilonka Gero and Lydia B Chilton. Metaphoria: An algorithmic companion for metaphor creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[Guzdial *et al.*, 2018] Matthew Guzdial, Nicholas Liao, and Mark Riedl. Co-creative level design via machine learning. *arXiv preprint arXiv:1809.09420*, 2018.

[Guzdial *et al.*, 2019] Matthew Guzdial, Nicholas Liao, Jonathan Chen, Shao-Yu Chen, Shukan Shah, Vishwa Shah, Joshua Reno, Gillian Smith, and Mark O Riedl. 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*, pages 1–13, 2019.

[Halina and Guzdial, 2021a] Emily Halina and Matthew Guzdial. A demonstration of kiaitime: A mixed-initiative pcgml rhythm game editor. In *The 17th AAAI conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*. AAAI, 2021.

[Halina and Guzdial, 2021b] Emily Halina and Matthew Guzdial. Taikonation: Patterning-focused chart generation for rhythm action games. *arXiv preprint arXiv:2107.12506*, 2021.

[Hendrikx *et al.*, 2013] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1):1–22, 2013.

[Liapis *et al.*, 2013] Antonios Liapis, Georgios N Yannakakis, and Julian Togelius. Sentient sketchbook: computer-assisted game level authoring. In *Proceedings of the 8th International Conference on the Foundations of Digital Games*, pages 213–220, 2013.

[Liapis *et al.*, 2016] Antonios Liapis, Gillian Smith, and Noor Shaker. Mixed-initiative content creation. In *Procedural content generation in games*, pages 195–214. Springer, 2016.

[Lin *et al.*, 2019] Zhiyu Lin, Kyle Xiao, and Mark Riedl. Generationmania: Learning to semantically choreograph. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, pages 52–58, 2019.

[Neate *et al.*, 2019] Timothy Neate, Abi Roper, Stephanie Wilson, and Jane Marshall. Empowering expression for users with aphasia through constrained creativity. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–12, 2019.

[Sturtevant *et al.*, 2020] Nathan Sturtevant, Nicolas Decroocq, Aaron Tripodi, Carolyn Yang, and Matthew Guzdial. A demonstration of anhinga: A mixed-initiative epcg tool for snakebird. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, pages 328–330, 2020.

[Summerville *et al.*, 2018] Adam Summerville, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgård, Amy K Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. Procedural content generation via machine learning (pcgml). *IEEE Transactions on Games*, 10(3):257–270, 2018.

[TPGPL *et al.*, 2020] TPGPL, Semro, and MegaApplePi. osu!taiko. https://osu.ppy.sh/wiki/en/Game_mode/osu!taiko, 2020. Accessed: 2021-08-16.

[Veale and Cardoso, 2019] T. Veale and F.A. Cardoso. *Computational Creativity: The Philosophy and Engineering of Autonomously Creative Systems*. Computational Synthesis and Creative Systems. Springer International Publishing, 2019.

[Yalçın *et al.*, 2020] Özge Nilay Yalçın, Nouf Abukhodair, and Steve DiPaola. Empathic ai painter: A computational creativity system with embodied conversational interaction. In *NeurIPS 2019 Competition and Demonstration Track*, pages 131–141. PMLR, 2020.

[Yannakakis and Togelius, 2018] Georgios N Yannakakis and Julian Togelius. *Artificial intelligence and games*. Springer, 2018.

[Zhu *et al.*, 2018] Jichen Zhu, Antonios Liapis, Sebastian Risi, Rafael Bidarra, and G Michael Youngblood. Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2018.