Taming the chaos: exploring graphical input vector manipulation user interfaces for GANs in a musical context

Ruben Schlagowski, Silvan Mertes, Elisabeth André

Angaben zur Veröffentlichung / Publication details:

Schlagowski, Ruben, Silvan Mertes, and Elisabeth André. 2021. "Taming the chaos: exploring graphical input vector manipulation user interfaces for GANs in a musical context." In *AM '21: Audio Mostly 2021, virtual/Trento, Italy, September 1-3, 2021*, 216–23. New York, NY: ACM. https://doi.org/10.1145/3478384.3478411.

Taming the Chaos: Exploring Graphical Input Vector Manipulation User Interfaces for GANs in a Musical Context

Ruben Schlagowski
ruben.schlagowski@uni-a.de
University of Augsburg
Augsburg, Germany

Silvan Mertes
silvan.mertes@uni-a.de
University of Augsburg
Augsburg, Germany

Elisabeth André
elisabeth.andre@uni-a.de
University of Augsburg
Augsburg, Germany

ABSTRACT

Generative Adversarial Networks (GANs) are a widely used tool for generating highly realistic artificial data. As the output of these networks can show high diversity and novelty, GANs have the potential to be used as creative tools. However, using GANs in this context poses major challenges due to their unpredictability and lack of controllability, making it difficult for creative people to realize their artistic vision. To address this problem, we present two graphical user interfaces that visually order the (otherwise chaotic) latent input space of a GAN that was trained to generate drum samples. Further, these GUIs provide convergent search functions that allow users to fine-tune generated sounds. By doing so, we provide the ability to create sounds more purposefully to sound-affine users such as musicians or sound engineers. Additionally, we present the results of a user study that we conducted in order to explore our approach in accuracy-oriented and creative tasks. Our results indicate that usability and pragmatic qualities play a more important role for users than aesthetic-oriented aspects. Although not improving the accuracy within reproductive tasks, we observed that convergent search functions, if available, were used significantly more often than divergent/randomized search functions.

CCS CONCEPTS

• **Human-centered computing → User interface design**; **Empirical studies in interaction design**.

KEYWORDS

Generative Adversarial Networks, Human-Computer Interaction, Interactive Sound Generation, Sound Design, Audio Synthesis

ACM Reference Format:
Ruben Schlagowski, Silvan Mertes, and Elisabeth André. 2021. Taming the Chaos: Exploring Graphical Input Vector Manipulation User Interfaces for GANs in a Musical Context. In *Audio Mostly 2021 (AM '21), September 1–3, 2021, virtual/Trento, Italy.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3478384.3478411

AM '21, September 1–3, 2021, virtual/Trento, Italy

1 INTRODUCTION

Generative Adversarial Networks (GANs)[12] have been used in a variety of applications to mimic and reproduce human-generated data. However, their use as tools for creative people still remains to be explored. In this paper, we outline a new approach that enables GANs to be used as a novel type of musical instrument or sound design tool.

A big challenge of using GANs in creative applications is their unpredictability. Although GANs can produce deceptively real data such as percussive sounds, potential users of the technology have little opportunity to shape the generated sounds to match their artistic vision or imagination. This stems from the fact that conventional GANs approximate a transformation function that converts random noise vectors to new audio data. As the unstructured noise vectors follow a random distribution, they are by no means interpretable for humans.

To address this problem, we present two different user interfaces, which we call Vector Manipulation Modules (VMMs). These give users the ability to fine-tune the sounds produced by GANs by visualizing the seemingly chaotic input latent space of the generative network as manipulatable objects within two or three spatial dimensions. By doing so, we provide sound-affine users (e.g., musicians or sound engineers) with the ability to directly interact with the GAN. Additionally, we order the visual representation of the latent space regarding the single elements' impact on the generated audio samples. Thus, we ensure a certain degree of predictability of the adjustments a user makes.[1]

The research questions we address in this paper are:

(1) How accurately can users reproduce sounds with our two distinct VMM designs (2D and 3D approaches)?
(2) Which parameter search methods do users prefer while using the VMMs (convergent or divergent)?
(3) How do these VMM designs impact user behaviour both in creative and reproductive/targeted search tasks?
(4) How do users experience interaction with our VMM designs?

To assess our system, we conducted an online user study in which participants interacted with our VMMs which were integrated into a GAN-based drum sequencer. For research questions (1), (2) and (3), we evaluate objective quantitative measures derived from user interaction logs. For research question (4), we look at self-assessment measures such as user experience, the user's state or sensation of flow while working with these interfaces, and self-efficacy.

[1] The web-based demonstrator we used for our online-study is currently hosted as a live-version containing both VMMs at *hcai.eu/pufferfish*.

2 RELATED WORK

2.1 Exploration of Parameter Spaces for Musical Creativity

A typical use case for parameter search is sound synthesis. A variety of approaches and algorithms have been applied here, such as mapping parameters to 2D-Interfaces using Hilbert curves [29], exploration of parameter spaces with evolutionary algorithms [7, 26] or by using machine learning models for mapping gestures to synthesizer parameters [13]. Previous work suggests that different parameter navigation strategies can benefit different stages of the creative process. For instance, Tubb and Dixon [29] found evidence for divergent exploration and convergent honing/fine-tuning behaviors and argued that systems should offer user interfaces enabling both strategies. Another interesting approach is understanding parameter mapping algorithms as new musical instruments that can be used in real-time. For instance, Berndt et al. [2] built a touch-based musical instrument that modulates noise. Other ideas include tangible interfaces such as Snyder's *Birl* [27] that is based on incorporating an artificial neural network within an electronic wind instrument using Fiebrink's *Wikinator* framework [11] for sonification of gestures. A substantial amount of research covers deep learning-based applications for music composition and music generation. In these approaches, a challenge is that often no ways to control the generated output (e.g. tonality) are provided [5]. Fiebrink and Caramiaux [10] described various ways in which machine learning (ML) algorithms themselves can be understood as human-computer interfaces in musical contexts. Explicitly, the authors described that such algorithms can be used as tools for musical interaction, as creative tools, and also as a new type of user interface, e.g., by exposing affordances of ML algorithms in a variety of ways. The authors elaborate on the possibility to expose different control parameters that can affect the characteristics of a trained model. As an example, Fiebrink mentioned work from Morris et al. [23] who exposed parameters such as *happy* or *jazz* factors to tune hidden Markov models generating chords. Kaliakatsos-Papakostas et al. [16] provided user access to the interactive parameters *rhythm*, *density* and *pitch* for LSTMs that were trained to compose music.

2.2 Generative Adversarial Networks

GANs were first introduced by Goodfellow et al. [12] and opened the possibility of generating high-quality artificial data. The basic idea of GANs is that two networks, namely the *Generator* and the *Discriminator*, are trained in an adversarial manner. Hereby, the generator learns to transform a random noise vector to new data that resembles a specific training dataset, while the discriminator learns to distinguish between real data and the fake data generated by the generator. Thus, the generator aims to fool the discriminator, while the discriminator aims to not be fooled. Modifications and improvements of GANs have been applied to various domains and applications. One modification to the original GAN that is particularly interesting for our use case was presented by Donahue et al. [8], who introduced *WaveGAN*. WaveGAN was designed to generate audio data of high quality and has already been applied successfully to the generation of drum sounds in the original publication.

2.3 GANs & Controllability

With the ongoing rise of GANs, approaches for gaining control over the outputs of these models quickly became an active field of research. The first attempt to directly incorporate modifiable features into the training of a GAN framework was presented by Mirza and Osindero [22], who suggested to enhance the input space of a GAN with additional dimensions that are trained in a supervised manner simultaneously to the unsupervised training of the GAN. Those so-called *Conditional GANs* were adapted for various modalities and applications. For example, Lee [20] applied the idea of conditional GANs to the WaveGAN framework. Further approaches for a feature-oriented training of GANs developed quickly, but are mainly limited to image generation tasks [17, 18, 24]. Other methods for controlling the output of GANs focus on dealing with the latent input space, thus performing latent vector manipulations. For example, Dosovitskiy et al. [9] explore different techniques of interpolating between different points in the latent space of a GAN. Härkönen et al. [14] applied *Principal Component Analysis* to the GAN input space. Another approach to search through the latent space of a trained GAN is *Latent Variable Evolution* (LVE), a method where evolutionary algorithms are used to search through the chaotic input space of a GAN. LVE was applied to different domains, such as video games [30] or fingerprint-based biometric systems [4]. It was also deployed successfully for searching through the latent space of a WaveGAN for the purpose of augmenting datasets [21]. Further, LVE was used to give human users the ability to interactively evolve through a learned GAN space [3, 31]. However, all the aforementioned methods focus on controlling the output of GANs by making it more interpretable and transparent. Contrary to that, besides providing visual structure for the input latent space, our approach aims at giving users the possibility to directly interact with the seemingly chaotic latent space.

3 APPROACH

3.1 WaveGAN

WaveGAN, which was first introduced by Donahue et al. [8], is a modification to the original GAN framework specifically designed for the generation of audio data. Therefore, WaveGAN includes minor changes to the original GAN architecture, e.g., the use of one-dimensional filters. For our system, we use a WaveGAN model that was trained on a dataset consisting of drum sound recordings. The total length of the dataset is 0.7 hours. The trained model was made freely accessible by the authors of WaveGAN. For further insights into WaveGAN and the used model, please refer to Donahue et al. [8].

3.2 Sequencer User Interface

To provide a context for our novel vector manipulation user interfaces, we embedded them into an existing web-based drum sequencer UI that was previously published by Chris Donahue as a demonstrator for his WaveGAN architecture (see Figure 1)[2].

A drum sequencer provides the user functionalities to program a drum beat by using a grid UI that can assign notes to different

[2] https://chrisdonahue.com/wavegan/

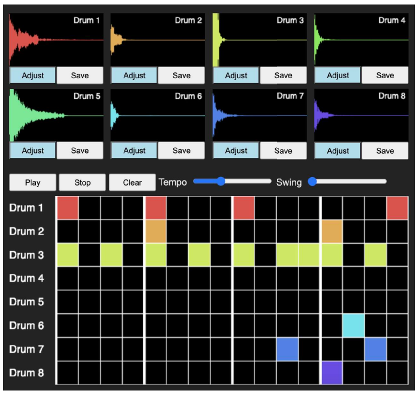

Figure 1: Screenshot of the web-based drum sequencer UI implemented by Chris Donahue.

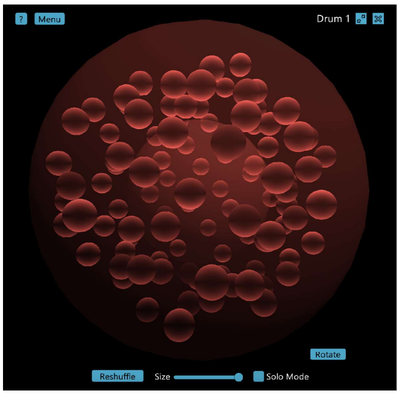

Figure 2: Screenshot of the Vector Manipulation Module *Fibonacci Sphere* (1).

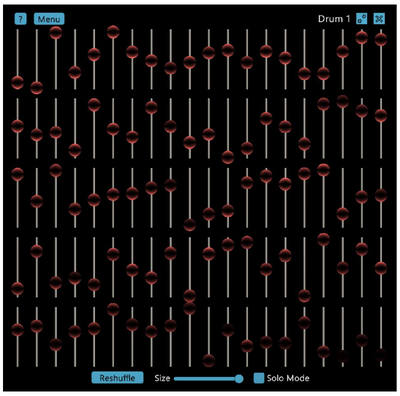

Figure 3: Screenshot of the Vector Manipulation Module *Slider Array* (2).

audio channels. This beat can be played back in a loop and modified at runtime by changing the notes that are assigned to the channels. The drum sequencer we use utilizes sounds generated by WaveGAN.

3.3 Vector Impact Analysis

A critical challenge in creating user interfaces is clarifying the nature of the offer (the affordances) that individual interactive elements provide. To make the initially chaotic structure of the input latent space more understandable in this respect, we analyzed the WaveGAN network regarding the effects that individual vector elements have on generated audio samples. Therefore, we used an offline analysis procedure in which individual random vectors were altered for several epochs. After each manipulation, these altered vectors were fed into WaveGAN and the resulting outputs were compared to the WaveGAN outputs of the original vectors regarding Power Spectral Density (PSD). Subsequently, the mean impact for individual vector elements was calculated. This method led to a ranking, sorting vector elements according to their impact on the output.

In a second step, we investigated the impact individual input values have on the auditory spectrum of the generated sounds by calculating the deviations in PSD for the frequency bands (1) *20 to 4000 Hz* and (2) *4001 Hz to 16000 Hz*

by using the procedure described above on those limited bandwidths, leading to separate rankings for the individual bands. These rankings were used for structuring the visual latent space representations within the VMMs.

3.4 Vector Manipulation Modules

To give users the ability to fine-tune the sounds generated by the WaveGAN, we developed two different UI designs, which we call Vector Manipulation Modules (VMMs). Both designs have a number of controls equal to the GAN's input latent space dimension (d = 100). Each of these controls (*Manipulatable Elements* or MEs) represents a single element of the input latent space. Through manipulation of the MEs, their respective numerical value in the input latent space can directly be changed. All MEs are presented in different spatial constellations within both VMMs:

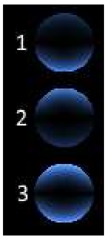

Figure 4: A custom shader was used to illustrate the impact that individual MEs have on different frequency bands. In this example, sphere no. 1 has a comparably high impact on lower bands, sphere no. 2 a high impact on the upper bands and sphere no. 3 has a strong impact on the whole audible spectrum.

(1) The VMM *Fibonacci Sphere* (see Figure 2) presents MEs as a set of 3D spheres that are spatially placed between two larger 3D spheres that illustrate both their minimum and maximum values. Accordingly, the spatial distance of the MEs from the minimum and maximum spheres determines their numerical value. Each ME can be manipulated by the user via drag and drop. By using the fibonacci lattice algorithm [28], positional offsets were calculated for each ME, resulting in the latent input space being represented as a spherical structure.
(2) The VMM *Slider Array* (see Figure 3) presents MEs as an array of sliders, which can be adjusted analogously to the Fibonacci Sphere VMM. The numerical value represented by the MEs is determined by the spatial distance of the slider handles to the limits of the sliders.

The MEs of both VMMs were sorted according to their impact orders that were determined as described in Subsection 3.3. For (1), more impactful spheres were placed in higher positions, while less impactful spheres were placed further down. For (2), more impactful sliders were placed closer to the top left corner of the screen and less impactful spheres closer to the bottom right corner. For both VMMs, the additional UI slider *Size* enables users to hide less impactful MEs.

To visualize the impact that individual MEs have on the upper and lower audible frequency bands, each sphere or slider handle was shaded differently. Spheres and slider handles that have a high

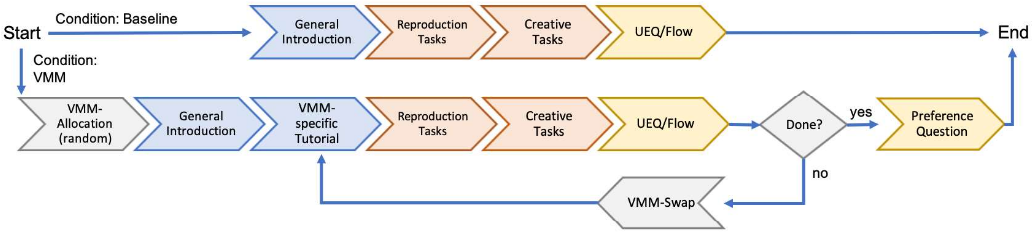

Figure 5: Flow chart of the user study.

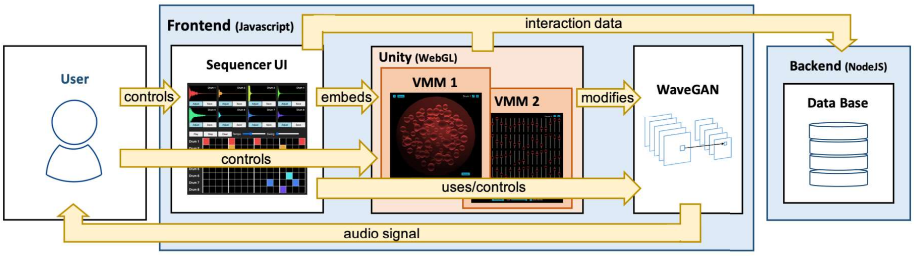

Figure 6: Technical setup of the demonstrator.

impact on lower frequency bands are displayed bright on the bottom, while low impactful MEs remain dark in that region. The same principle applies to the upper half of the MEs to indicate their impact on higher frequency bands. Figure 4 illustrates this principle.

An additional feature that both VMMs provide is a *Reshuffle* button that generates a new random vector which is then passed to WaveGAN. Subsequently, the positions of the MEs are updated accordingly. This feature serves as a divergent search tool that enables users to quickly explore a large variety of sounds.

4 USER STUDY

4.1 Study Procedure

In order to examine our research questions, we conducted a user study which was run as an online experiment. The basic flow of our study can be seen in Figure 5.

Table 1: Descriptive information about the participants.

Characteristic	Baseline	VMM	Total
n	17	21	38
Age			
Mean	27.47	28.24	27.89
Standard Deviation	5.30	6.29	5.88
Gender			
male	15	12	27
female	1	8	9
diverse	0	1	1
no answer	1	0	1

38 Participants (see Table 1) were recruited via Amazon MTurk, where each participant needed to pass a test in advance to verify that she or he is capable of identifying typical drum sounds like snare or bass drums, can assign different drum patterns to their visual representation in a sequencer grid, and is able to identify subtle differences within sound samples. We collected data in a top-level between-groups experiment. Here, one group of participants were provided with the drum sequencer interface featuring solely the divergent search/reshuffle function, i.e., the rhythms could be adjusted, but the sounds could only be altered by randomly reshuffling the input vector for the WaveGAN (see Figure 5: *baseline* condition). The participants that were in the other group (the *VMM* condition) additionally saw both the Fibonacci Sphere and the Slider Array VMMs in randomized order. By measuring our dependent variables after each of these stimuli, we nested a within-subject experiment into the between-groups experiment. By doing so, we were able to ask participants which VMM they preferred. Additionally, experiencing both VMMs enabled participants to provide open feedback concerning both versions.

After randomly being assigned to either the *VMM* or *baseline* condition, demographic data was collected. Then, each group was provided with a basic overview of the sequencer user interface. Subsequently, the participants were asked to construct a simple drum beat to get familiar with the drum sequencer UI, before watching an animated tutorial explaining the interaction principles of their respective VMM (if not in the baseline condition). Afterwards, the participants went through two task sections:

(1) *Reproductive Task Section* The participants were asked to recreate three given drum sounds by using the demonstrator. In the baseline condition this could be achieved solely

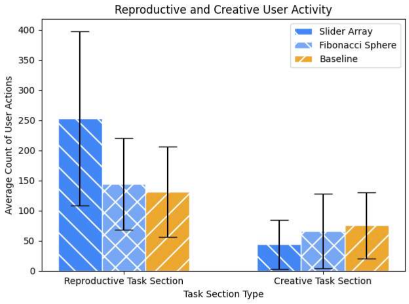

Figure 7: Average Count of user actions for the first reproduction and creative task sections.

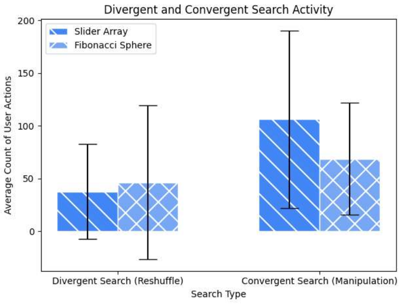

Figure 8: Amount of convergent and divergent search actions in the VMM condition.

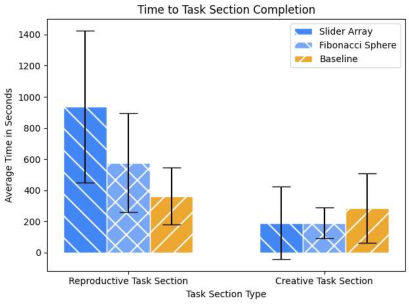

Figure 9: The average time it took for participants to complete their first reproduction and creative task sections.

through divergent search (i.e., reshuffle). In the VMM condition, both divergent search and convergent search (i.e., ME manipulation) functionalities could be used.

(2) *Creative Task Section* In this section, the participants were allowed to freely explore the demonstrator and do whatever they wanted, living out their creativity.

After completing both task sections the questionnaires described in Section 4.2 were filled out. For participants in the baseline condition, the experiment ended at this point, while participants in the VMM condition had to repeat both task sections while using the respective other VMM.

During interaction with the demonstrator, information about user behaviour was logged for objective evaluation. The architecture of the demonstrator used within our study can be seen in Figure 6.

4.2 Evaluation Measures

4.2.1 Objective Measures. The data we collected included logs about interactions with the sequencer UI and the VMMs.

From these logs we derived the following objective measures:

(1) *Reproductive and Creative User Activity* is the number of logged interactions both within reproductive and creative task sections.
(2) *Divergent and Convergent Search Activity* is the number of logged user interactions with the reshuffle functionality (divergent search) as well as ME manipulations (convergent search).
(3) *Time to Task Section Completion* is the time users needed to complete reproductive and creative task sections respectively.
(4) *Auditive Similarity of Vector Results*: From the logs containing the vectors users created within the productive task sections, we reproduced the generated sounds and compared their PSD with the PSD of the target sounds.

To eliminate sequence order effects due to the repetition of task sections within the VMM condition (see figure 5) we only took data of the first reproductive and creative task sections into account for each participant (applies to measure (1), (3) and (4)).

4.2.2 Self-Assessment Measures. In order to assess the users’ subjective experience while interacting with our demonstrator, the following self-assessment measures and questionnaires were used for both VMM and baseline conditions:

(1) We measured the participants’ *User Experience* with the UEQ questionnaire [19]. This questionnaire measures six dimensions of user experience called *perspicuity, efficiency, dependability, stimulation, novelty* and *attractiveness.*
(2) We used the Flow Short Scale [25] to measure the users’ state of *flow*, which has been described as a highly enjoyable psychological state that refers to the *”holistic sensation people feel when they act with total involvement (in an activity)”* [6]. To be able to assess his or her state retrospectively, the items of the questionnaire were slightly modified (i.e., put in past tense).
(3) For *Self-Efficacy*, we used a one-item scale. We used a variation of the scale proposed by Bernacki et al. [1] (*”How confident are you that you would be able to generate exactly the drum sounds that you aim for in the future with the system you just used?”*).

4.2.3 VMM Preference and Open Feedback. After finishing all task sections, participants within the VMM condition were asked which VMM version they liked better. Furthermore, they were encouraged to provide open feedback.

5 RESULTS

For each dependent variable (in order to check if the data is parametric), we used Shapiro’s test to check if the values are normally distributed and a Levene’s test for equal variances. For independent measures (between VMM and baseline), if both normal distribution and equal variances were given, we performed a one-way ANOVA and post hoc t-tests to check if significant differences existed within the tested pairs. If not, we performed a Kruskal-Wallis test and post hoc Mann-Whitney-U tests. For dependent measures (within the VMM condition), which only affected divergent/convergent search activity (see Figure 8), we used Friedmann’s test and post hoc Wilcoxon’s signed rank tests as data was not found to be parametric in that case. After the post hoc tests, we corrected p values with Holm-Bonferri’s method. In the following, we report only

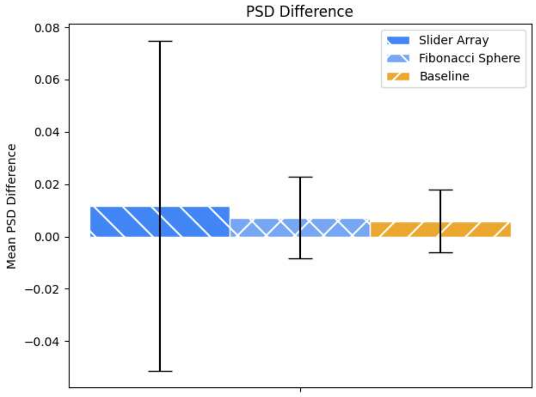

Figure 10: Mean differences of the Power Spectral Density (PSD) between the target sounds and the user-generated sounds. Small differences indicate high similarity.

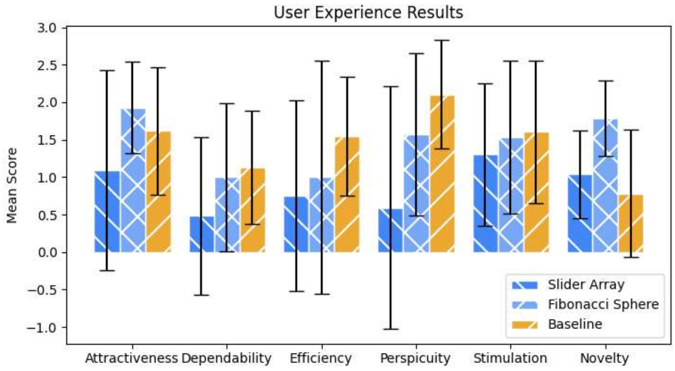

Figure 11: UEQ results for both VMMs and baseline condition. Only the first UEQ results were evaluated for each participant in the VMM condition.

significant differences. All other results are depicted in Figures 7 - 13.

5.1 Objective Results

- Users that were in the VMM condition performed significantly less divergent search actions than convergent search actions ($p = 0.0051$).
- While interacting with the Slider Array VMM, users performed significantly more convergent search actions than divergent search actions ($p = 0.0107$).
- During the reproductive task section, users performed significantly more actions if working with the Slider Array VMM than the participants in the baseline condition ($p = 0.0110$).
- Participants that were confronted with the Slider Array VMM spent significantly more time in the reproductive task section than users in the baseline condition ($p = 0.0001$).
- Participants using the Slider Array VMM used convergent search functions significantly more often than participants using the Fibonacci Sphere VMM ($p = 0.0029$).

5.2 Self-Assessement Results

- Users in the Baseline condition rated their UI significantly higher in terms of *perspicuity* than users in the Slider Array VMM condition ($p = 0.0188$).
- Users rated the Fibonacci Sphere VMM to be significantly more *novel* than the Slider Array VMM ($p = 0.0155$).
- Users interacting with the Fibonacci Sphere VMM perceived the system as significantly more *novel* than users in the baseline condition ($p = 0.0155$).

5.3 User Preference Results

As can be seen in Figure 13, users in the VMM condition tended to like the Slider Array VMM more than the Fibonacci Sphere VMM.

It should be noted, however, that our randomization algorithm ended up assigning 14 participants the Slider Array VMM first, while only 7 participants saw the Fibonacci Sphere VMM first. It is conceivable that this circumstance led to ordering effects that skewed users' responses in terms of VMM preference.

6 DISCUSSION

6.1 Discussion of Results

Pragmatic quality overshadows hedonistic quality. In the evaluation of the user experience, the 3D interface *Fibonacci Sphere* was on average rated better than the 2D interface *Slider Array* in all UEQ dimensions. The greatest differences were found in the dimensions of *novelty* and *attractiveness*, which are hedonistic qualities [19]. This was also reflected individually in the open user feedback. For instance, one participant wrote:

"I don't know how either work... thus, the less conventional sphere's floating around is more fun to experiment with compared to the dozens of sliders that are unlabeled (which I have seen on synths before)"

This can be largely explained due to the visual aesthetic of the sphere, which uses three dimensions for the placements of the manipulatable elements while additionally rotating them, making the MEs appear larger in size when they are closer to the camera.

However, this is in stark contrast to the answers users gave when asked about their preferred interface (see Figure 13). Here, the Slider Array VMM performed substantially better. Furthermore, the objective logs show that users spent more time and were more active while interacting with the Slider Array VMM than with both the Fibonacci Sphere VMM or the baseline system. Furthermore, the convergent search functions of the Slider Array VMM were used significantly more often than their equivalents within the Fibonacci Sphere VMM. This enhanced user engagement is not reflected in the evaluation of the pragmatic UEQ dimensions *efficiency* and *perspicuity* which can be explained by insufficient sample size and large standard deviations. We found possible reasons for these observations while looking at open user feedback, where participants described the slider VMMs as feeling more familiar, more efficient and less confusing.

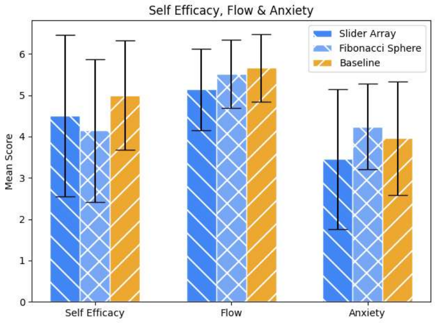

Figure 12: Means for self-efficacy, flow and anxiety for all conditions.

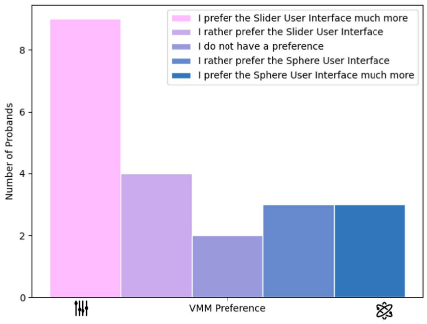

Figure 13: User preferences between the two VMM designs.

Hassenzahl et al. [15] suggest that the user’s experience and perception of a system depend on the mode in which they use the system. In our case, participants are likely to have been in *goal mode* as they participated in order to generate income and therefore maximize efficiency.

Spatial dimensions can limit accessibility. A reason for the lack of pragmatic qualities associated with the Fibonacci Sphere VMM might be the inaccessibility of certain manipulatable elements that were placed behind other MEs. We tried to solve this issue by dynamically rotating the entirety of the 3D-UI and providing users a functionality to rotate it faster, but this might not compensate for the users’ needs to quickly explore the parameter space. One user made this fairly clear by writing:

”The spheres constantly move around making it distracting. You can also lose track of the spheres you clicked on. Also I found that if any of the spheres were directly behind the center sphere, I could not click on them or make any adjustments until they reached the edge of the center sphere or moved out of its radius. The slider interface feels more user-friendly than the sphere one.”

This lack of accessibility is due to a mismatch of spatial dimensions, as a three-dimensional order of the manipulatable elements is presented on a 2D screen. Thus, technologies supporting three spatial dimensions for user interaction such as Virtual and Augmented Reality (Mixed Reality) might be able to resolve this issue.

Users fine-tune their sounds extensively. If given the option to use both the convergent/fine-tuning strategies (manipulation of spheres or slider handles in our VMMS) and divergent search strategies (using the *reshuffle* function), participants used the convergent option more. We found significant differences both for the total count of convergent and divergent search actions as well as solely for the Slider Array VMM. From these results, we conclude that participants accepted our concept of manipulating input latent spaces with VMMs in order to search for targeted sounds more purposefully.

This is in line with the findings of Tubb and Dixon [29] who argued that systems should offer both options, as users will use these functions for different tasks according to their needs. However, it should be noted that for our VMM designs, convergent search functions were placed quite prominently as they occupied most of the screen space and outweighed the divergent functions in number.

More functionality comes with complexity. The baseline version of the drum sequencer UI, which gave participants only the option to look for sounds by creating random input vectors, performed surprisingly well, especially within user experience ratings. For *perspicuity*, the baseline condition even performed significantly better than the sphere VMM. This can be explained by the complexity of the VMMs and the accompanying challenges that users were faced with while learning to use these interfaces and also with potentially unclear communication of affordances. One participant wrote:

”This is definitely something I could see messing with in my own productions, but it definitely needs to be more communicative as to the parameters being fed to the AI and what they are doing.”

As no participant mentioned the custom shaders displaying affected frequency bands or the order of the manipulatable elements in both VMMs, we assume that the impact/affordances might not have been communicated clearly enough. Further research and creative ideas are required to improve in this regard. This could be achieved by using a variety of modalities such as auditory, haptic or tactile feedback.

User effort does not reflect accuracy. When comparing the samples produced by the participants with the target samples for the reproduction tasks, we observed no significant differences between baseline and both VMMs. Instead, the condition which saw the most activity in convergent search (slider array) had the largest deviations in PSD similarity (compare Figures 10 and 8). As such, spending more time and effort for fine-tuning sounds did not result in more accurate solutions.

6.2 Lessons Learned

Efficiency does not reflect user satisfaction. Objective measures such as time to completion and user activity (which we measured by counting user interactions with certain UI functionalities) seem to correlate positively with user approval of the UI they are using. Therefore, time efficiency should not be regarded as something desirable when designing parameter search UIs. This was also reflected in our results, where users of the slider array VMM were significantly slower than baseline participants and substantially slower than users of the sphere VMM during the reproduction task sections.

Temporal constraints limit creative output. Time constraints can heavily impact both temporal measures (such as time to completion) and cumulative measures such as user activity. From our data we observed that participants in the baseline condition spent more time in the creative task sections and showed a slightly increased number of interactions. This could be due to them having less tasks to fulfill in the time frame they were paid for. We assume that these effects also negatively affected the users' likeliness to experience the state of flow or even their perceived self-efficacy, which might be a reason for us not observing any significant effects within these measures. Thus, we recommend giving users the option to exceed time limits when designing similar studies.

7 CONCLUSION & OUTLOOK

In this paper, we presented a new GUI-based approach to both providing structure and the ability to interact with input latent spaces for GANs in a creative and musical context. Furthermore, we presented the results of a user study in which we evaluated two distinct GUI designs (Vector Manipulation Modules or VMMs) which were integrated into a WaveGAN-based drum sequencer, one using three and the other using two spatial dimensions. From investigating user behavior, self-assessment questionnaires, and open feedback we conclude that usability and pragmatic qualities of the 2D-version (Slider Array) were heavily appreciated by participants, while the emphasized visual aesthetics of the 3D-version (Fibonacci Sphere) seemed to be less relevant for study participants. Further, we found that participants used the convergent search functions that our VMMs provide more often than purely randomized/divergent search methods, indicating that users appreciated having the possibility to directly interact with a GAN on a detailed level.

In the future, we aim to research more methods and concepts for communicating affordances for latent space vectors. Further, we plan to explore the potential of immersive technologies such as Virtual or Augmented Reality to assess the applicability of the three-dimensional approach by using hardware that is built for interaction in three spatial dimensions. Also, we aim to conduct future studies that investigate the potential of our approach in environments and use cases that are solely designed to support creativity.

ACKNOWLEDGMENTS

We would like to thank Chris Donahue for the permission to use the code of the WaveGAN-based drum sequencer as well as for providing valuable feedback for this project.

REFERENCES

[1] Matthew L Bernacki, Timothy J Nokes-Malach, and Vincent Aleven. 2015. Examining self-efficacy during learning: Variability and relations to behavior, performance, and learning. *Metacognition and Learning* 10, 1 (2015), 99–117.

[2] Axel Berndt, Nadia Al-Kassab, and Raimund Dachselt. 2015. TouchNoise: A New Multitouch Interface for Creative Work with Noise. In *Proceedings of the Audio Mostly 2015 on Interaction With Sound*. 1–8.

[3] Philip Bontrager, Wending Lin, Julian Togelius, and Sebastian Risi. 2018. Deep interactive evolution. In *International Conference on Computational Intelligence in Music, Sound, Art and Design*. Springer, 267–282.

[4] Philip Bontrager, Aditi Roy, Julian Togelius, Nasir Memon, and Arun Ross. 2018. Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 1–9.

[5] Jean-Pierre Briot and François Pachet. 2017. Music generation by deep learning-challenges and directions. *arXiv preprint arXiv:1712.04371* (2017).

[6] Mihaly Csikszentmihalyi. 2000. *Beyond boredom and anxiety.* Jossey-Bass.

[7] Palle Dahlstedt. 2001. Creating and exploring huge parameter spaces: Interactive evolution as a tool for sound generation. In *ICMC*. Citeseer.

[8] Chris Donahue, Julian McAuley, and Miller Puckette. 2018. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208* (2018).

[9] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. 2015. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1538–1546.

[10] Rebecca Fiebrink and Baptiste Caramiaux. 2016. The machine learning algorithm as creative musical tool. *arXiv preprint arXiv:1611.00379* (2016).

[11] Rebecca Anne Fiebrink. 2011. *Real-time human interaction with supervised learning algorithms for music composition and performance.* Citeseer.

[12] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *arXiv preprint arXiv:1406.2661* (2014).

[13] Jeff Gregorio and Youngmoo E Kim. 2019. Augmenting Parametric Synthesis with Learned Timbral Controllers.. In *NIME*. 431–436.

[14] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546* (2020).

[15] Marc Hassenzahl, Robert Kekez, and Michael Burmester. 2002. The importance of a software's pragmatic quality depends on usage modes. In *Proceedings of the 6th international conference on Work With Display Units (WWDU 2002)*. ERGONOMIC Institut für Arbeits-und Sozialforschung Berlin, 275–276.

[16] Maximos Kaliakatsos-Papakostas, Aggelos Gkiokas, and Vassilis Katsouros. 2018. Interactive control of explicit musical features in generative lstm-based systems. In *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*. 1–7.

[17] Takuhiro Kaneko, Kaoru Hiramatsu, and Kunio Kashino. 2018. Generative adversarial image synthesis with decision tree latent controller. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6606–6615.

[18] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4401–4410.

[19] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and evaluation of a user experience questionnaire. In *Symposium of the Austrian HCI and usability engineering group*. Springer, 63–76.

[20] Jan Lee. 1981. Transcript of question and answer session. In *History of programming languages I (incoll)*, Richard L. Wexelblat (Ed.). ACM, New York, NY, USA, 68–71. https://doi.org/10.1145/800025.1198348

[21] Silvan Mertes, Alice Baird, Dominik Schiller, Björn W Schuller, and Elisabeth André. 2020. An Evolutionary-based Generative Approach for Audio Data Augmentation. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 1–6.

[22] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[23] Dan Morris, Ian Simon, and Sumit Basu. 2008. Exposing parameters of a trained dynamic model for interactive music creation. (2008).

[24] Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. 2019. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 4610–4617.

[25] Falko Rheinberg and Regina Vollmeyer. 2003. Flow-Erleben in einem Computerspiel unter experimentell variierten Bedingungen. (2003).

[26] Hannes Ritschel, Ilhan Aslan, Silvan Mertes, Andreas Seiderer, and Elisabeth André. 2019. Personalized synthesis of intentional and emotional non-verbal sounds for social robots. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 1–7.

[27] Jeff Snyder and Danny Ryan. 2014. The Birl: An Electronic Wind Instrument Based on an Artificial Neural Network Parameter Mapping Structure.. In *NIME*. 585–588.

[28] Richard P Stanley. 1975. The Fibonacci lattice. *Fibonacci Quart* 13, 3 (1975), 215–232.

[29] Robert Tubb and Simon Dixon. 2014. The Divergent Interface: Supporting Creative Exploration of Parameter Spaces.. In *NIME*. 227–232.

[30] Vanessa Volz, Jacob Schrum, Jialin Liu, Simon M Lucas, Adam Smith, and Sebastian Risi. 2018. Evolving mario levels in the latent space of a deep convolutional generative adversarial network. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 221–228.

[31] Nicola Zaltron, Luisa Zurlo, and Sebastian Risi. 2020. Cg-gan: An interactive evolutionary gan-based approach for facial composite generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 2544–2551.