From Novice to Composer: Using AI to Facilitate Music Creation with MIDI Generation and Sample Extraction

Slingerland, Erik
Responsible AI lab
Amsterdam University of Applied Sciences,
Amsterdam, The Netherlands
erik.slingerland01@gmail.com

***Abstract*—Concerns have been raised over the increased prominence of generative AI in art. Some fear that generative models could replace the viability for humans to create art and oppose developers training generative models on media without the permission of the artist. Proponents of AI art point to the potential increase of accessibility. Is there an approach to address the concerns raised by artists, while still utilizing the potential these models bring?. Current models often aim for autonomous music generation. This however makes the model a black-box that users can't interact with. By utilizing an AI pipeline combining symbolic music generation and a proposed sample creation system, trained on Creative-Commons data, a musical looping application has been created to provide non-expert music users a way to start creating their own music. First results show that it assists users in creating musical loops and shows promise for future research into the field of human-AI interaction in art.**

I. Introduction

AI has become a tool used in various art disciplines. Examples include using natural language processing to help with writer's block [1], creating paintings using old media [2], or using AI-generated melodies as the basis of a song [3]. Some fear a trend where smaller artists get replaced by AI models [4]. There are concerns over ownership since these AI applications can be trained on data without creators' approval or knowledge [2] [5]. Proponents of using AI in art cite the idea that AI can improve accessibility towards art, encourage creativity [2] and could positively impact mental health [6].

To address this situation, the Responsible-AI lab of the Amsterdam University of Applied Sciences started a new project called Hitloop. Hitloop aims to research possible applications that allow for cooperation between humans and AI in creating music. This will be addressed in two domains: Transparency and Human Autonomy. It has therefore been decided to start with a pilot version aimed at non-musical specialists, to start a discussion about how AI can be created to cooperate with artists. This pilot version will focus on assisting users in the process of looping and sampling to create Electronic Dance Music (EDM).

In music looping, fragments of different audio sources are combined and rearranged to create new music. This technique started with avant-garde composers in the 1940s like Pierre Schaeffer with "Etude aux chemins de fer" and has continued into popular music like hip hop. These musical fragments are called samples and are selected by the musicians to be played in a reoccurring pattern [7]. This involves making small musical loops of the created samples. Most musicians arrange these loops in the Musical Instrument Digital Interface (MIDI) protocol. This protocol allows different musical devices and applications to communicate rhythm, pitch and speed in a standardised set of instructions. Within EDM, music looping is a standard approach for creating songs [8]. The AI pipeline proposed in this paper will assist in creating samples and creating MIDI patterns that the user can modify further.

II. Related work

Generating MIDI patterns is a form of symbolic music generation. This task involves creating music as represented in a non-auditory medium. Within symbolic music generation, MIDI is the standard format [9]. In this form, the pitch is represented on the vertical axis and time on the horizontal axis. Figure 1 presents an example of a drum track in MIDI format.

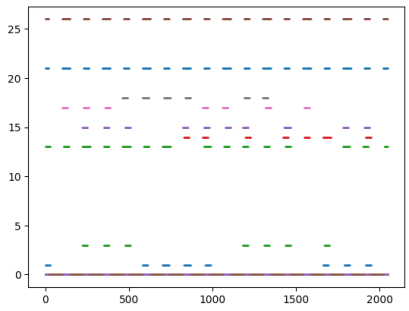


Fig. 1: Example of drums in a MIDI representation

Over the years, there have been different proposed methods to achieve music generation. Examples of applications

range from systems based on Markov chains to those using transformer models [9]. Some relevant papers that directly influenced the decisions in this paper are the Generative Adversarial Network (GAN) model MuseGAN [10], the Variational Auto-Encoder (VAE) model MusicVAE [11] and the Transformer-based Adaptive Music System [12].

Most of the models used for symbolic music generation are trained on datasets that are unsuitable for our research purpose. The majority of them are trained in classical music. One of the main reasons behind this is that this music no longer falls under copyright laws due to its age [10] [13]. For researchers who use copyrighted music, it is not always clear how much of this music can be shared due to copyright [11] [9]. Because of this, these datasets are often not shared and only used for research purposes. Our main purpose is keep a high level of transparency, and keeping the dataset publicly available is a top requirement in the context of EDM creation.

There have been no significant publications exploring the task of sample creation. This paper will therefore propose a new method. This method will be based on two discoveries during research. Firstly: Researchers have found that spectral representations of audio fragments offer machine-learning models sufficient information for the task of instrument identification [14] [15]. Secondly: When musicians use samples, they either take them from a sample library or create them by hand. The process of sample creation varies between musicians, but a standard method is extracting samples that are similar to instrument sounds [8].

III. Architectures/Proposed Pipeline

Three pipelines were employed in the creation of the Hitloop system. A MIDI dataset creation pipeline, a MIDI generation pipeline and a Sample extraction pipeline.

A. Midi Dataset creation

Previous researchers have successfully used a combination of models to create a pipeline for transforming audio into a symbolic representation [12]. This process has been emulated but with non-copyrighted EDM to create a MIDI dataset with publicly available data. As training data, Creative Commons (CC) music was used. Specifically the CC-BY, CC-BYSA, CC-BYNC, CC-BYNCSA and CC Zero licence [16]. These licences allow for remix/reuse of the audio and release of the resulting product non-commercially. Roughly 90 songs were used. The used songs are listed in the supplements.

Creating a MIDI dataset has three steps visualised in Figure 2. At first, the audio track is split into separate instrument sections using Meta's Demucs model [17]. By doing this, the different instruments can be utilised in separate channels for the next step. In the second step, the channels are transcribed into MIDI separately. The Drums are transcribed with Omnizart [18] and the pitch-based instruments with Basic Pitch by Spotify [19]. These transcribed instrument channel versions are combined into one bigger array. This created a representation of the song where each instrument channel can be addressed separately. This will allow the MIDI generator model to learn specific patterns for the instrument groups.

Finally, post-processing steps were applied, and the song was separated into its musical bars. EDM is characterised by small 1-4 bar repeating patterns where the musician adjusts the feel and danceability over time by making small adjustments during the song [8]. For this reason, the decision was made to focus on generating single bars instead of entire songs. In post-processing, the MIDI data was compressed into smaller dimensions. Because of EDM's focus on bass and drums [7], the decision was made to compress the data to the drum channels and a single bass channel. This was done to decrease the model train time and ensure users were able to run the model. This decreased the size on the vertical axis from 128x4 possible notes to 5 possible notes. The horizontal axis representing time also has decreased in size. MIDI normally represents time in a format of an array of 256 ticks for each beat. This means each beat has 256 possible on-off states. This would correspond to the possibility of using 512th notes to be distinguishable by having a rest between the shortest note. We do not require this amount of ticks because we consider anything below a 32nd note to be noise. We chose 32 ticks per beat because this allows us to distinguish up to 64th notes. Our horizontal axis, therefore, has a length from 1024 to 128.

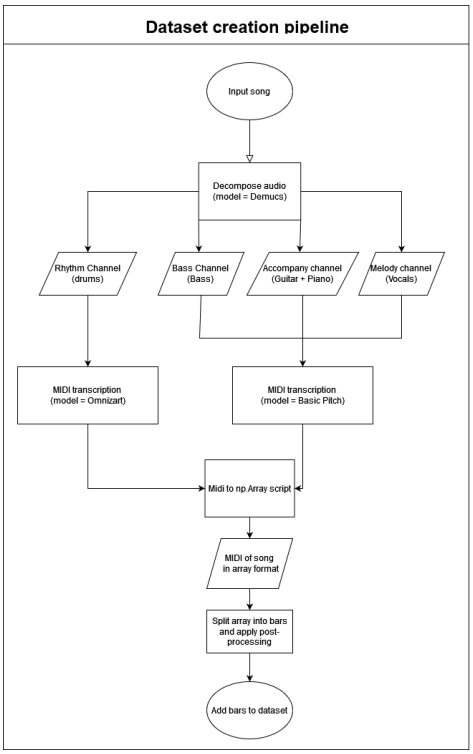


Fig. 2: Representation of the MIDI dataset creation pipeline

B. Midi Generator

There have been different proposed methods for MIDI generation. With consideration for deployment ability and quality, two architectures were tested for the MIDI generation aspect of Hitloop. The initial testing was with a GAN based on MuseGAN [10]. Secondly, experiments were conducted using a VAE based on MusicVAE [11]. Where the MIDI generator differs between MuseGAN and MusicVAE is the lack of recurrence in the model. The cited papers use recurrence to keep the music coherent between many bars. It was deemed unnecessary to implement because Hitloop only generates a single bar.

After experimentation models, three final models have been deemed candidates for human evaluation. Two models are based on the MuseGAN paper. GAN 1 was trained on 1000 epochs, and GAN 2 was trained on 500 epochs. Finally, one model based on the MusicVAE model was trained on 1000 epochs. The architectures of these models are available in the supplements.

C. Sample Creation

The proposed sample creation pipeline works by finding audio fragments within a recording that are similar to instrument sounds. This pipeline is visualized in Figure 3 This is done by first extracting auditory features from the instrument samples. The audio recording will be examined for potential samples using a sliding window. For every window, the same features get extracted. The features of the window are then compared against the features of all the instrument samples. When features have a certain similarity overall features, the audio fragment is extracted as a sample.

Different researchers have shown that spectral representations of audio fragments are sufficient features to categorize instruments [14] [15]. Therefore the decision was made to utilize spectral representations for representing audio fragments. For the sample extraction, mel-spectrograms were used. To support this, additional features were used to decrease the randomness of the process. After experimenting with different configurations, the additional features of Spectral bandwidth and Spectral contrast were selected.

The cosine distance was used to establish the similarity between the window and instrument sample. In different machine learning applications, cosine similarity is the industry standard [20]. The decision was made to implement this as a combined cosine distance where the cosine distance is calculated for each feature between the window and instrument samples. These are then added up to give a combined distance. Then to extract a sample, the combined cosine distance must be below a pre-defined threshold. By using this method, it is possible to influence the amount of- and similarity of the extracted samples. The higher the threshold, the more dis-similar samples will be extracted.

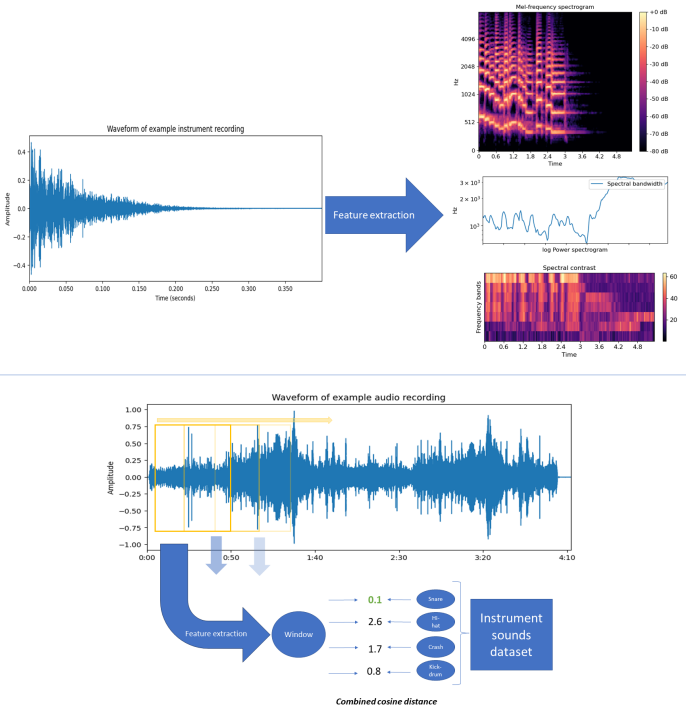


Fig. 3: Sample Extraction pipeline: Audio features get extracted for each window of the audio file. These are compared to instrument samples using cosine distance. Windows bellow the threshold get extracted

This sample extraction pipeline requires instrument samples and audio recordings to be extracted from. The Re:vive Amsterdam dataset from the Dutch Institute for Sound and Vision [21] was used as recordings to extract samples. These recordings are copyright free and intended to be used by DJs for music production. For the instrument samples: snare, hi-hat, cymbal and bass sounds were gathered from Wikimedia. The list of used files is found in the supplements. Both of these sources are copyright free and publicly available.

IV. Experimental Set-up

To assist in EDM creation, the system utilizes two AI models. A model for MIDI generation and a model for sample extraction. Both these models have the additional requirement to be transparent.

For the two models, different evaluation methods are required. The MIDI generation pipeline will be evaluated with a mix of quantitative and qualitative measures, as can be seen in Figure 4. In the first phase, quality measures are used to establish what developed model has similar characteristics to the EDM dataset. This model will be used in phase two, where user testing is conducted with non-expert users to establish whether the AI tool assisted in creating musical loops.

Three quantitative measures were used for the first phase. Firstly the average amount of unqualified notes in generated songs. We define an unqualified note as a note that is shorter than a 32nd note. This signals potential artefacts in the generated MIDI, as notes shorter than a 32nd note are rarely used in music [10]. Secondly, the ratio of empty space in the music was evaluated. This indicates what area of the generated music has no actively playing notes [10]. Usually, there are some areas of a song that are considered 'rests'. This measure has been compared to the dataset to see how similar it is to EDM. These first two measures were selected because of their ability to show possible artefacts of the AI in the form of too-short notes or unusual empty sections. Lastly, to signal the possibility of mode collapse in the MIDI generation, we devised a measure to look at the similarity of the model's outputs to signal variance in the resulting MIDI patterns. The final metric is a proposed metric to signal variation in the model output. To signal issues like mode collapse, the models generated 100 random patterns. These are all compared against each other using cosine distance. The resulting matrix from this is then plotted on a heat map. This heat map can show the variation between the generated MIDI patterns. The ideal outcome for a model here is to show a similar variation to the original data. If it has more variation, it could indicate abnormalities in the training outcome, while less variation would indicate the possibility of mode collapse.

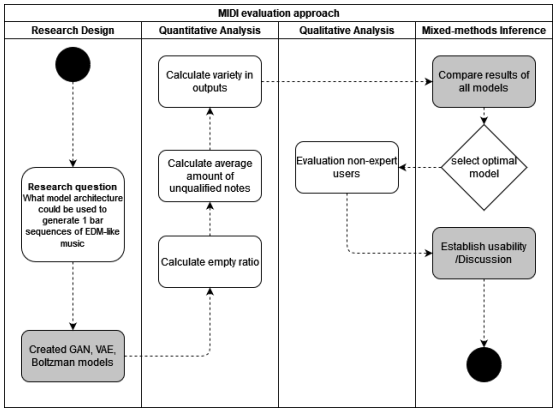


Fig. 4: MIDI evaluation pipeline

The evaluation of the sample creation pipeline only consists of one phase. Due to the lack of research surrounding sample extraction, there have also been no established quality measures. The decision was therefore made to also evaluate this aspect of Hitloop with user testing. To allow for this user testing, a prototype musical looping application was built using JavaScript and the tone.js [22] module for audio playback.

The prototype was designed to fit within the context of musical looping with non-expert users for EDM. EDM is characterized by repeating 1-4 bar patterns, where the musician adjusts the feel and danceability of the song by making small adjustments between sections[8]. For this reason, Hitloop was designed so users would utilize the components in a realistic context. The Hitloop prototype allows users to adjust a 1-bar MIDI pattern in real-time and adjust the tempo to emulate this process by EDM musicians. The focus on adjustability and 1-bar patterns was because the goal of this paper is not to create a state-of-the-art MIDI generation or sample extraction model but to approach this task from a responsible AI perspective. For the sample creation, the user currently has no control besides selecting which samples are played. Other researchers at the Responsible AI lab are experimenting with designing user-AI interactions for sample extraction.

During user testing, participants received two versions of Hitloop. Firstly a version utilizes the proposed AI systems for sample creation and MIDI generation. Secondly, a version where both systems operate randomly. The random samples were created by extracting audio fragments when a random number between 0 and 3 is below the thresholds defined for the sample creation pipeline. The random MIDI patterns are created by setting each cell in the interface to 1 or 0 with a coin flip. The users were not told that only one version uses AI. For both versions, the users received two tasks: 'Create a sequence that you enjoy when you play it repeated for 2-3 times' and 'Adjust the sequence so it feels either calmer or more energetic'. By using these two questions, we give little guidance but do ensure the user is guided towards EDM creation by making them emulate the process of creating an EDM loop and changing this for the song.

After completing the tasks, the users were tasked to answer four questions about the prototypes. These questions could be answered on a 5-point Likert scale, which went from Strongly disagree to Strongly agree. The questions are:

- I am happy with the loop I created
- The pre-made pattern in the sequencer helped me start making music
- The samples felt like they fit together
- There was enough variety in the samples

The first two questions were created to establish whether the user created a musical loop that they enjoyed and whether they thought that the generated pattern helped. The goal of this is to indicate whether the Hitloop application assisted people in creating musical loops and whether people noticed this. Between the random and AI version, it will be established whether using AI-generated MIDI patterns are necessary or whether the act of giving a starting point (even if it is random) is sufficient.

The last questions were created to establish the user's attitude towards the samples. The first question establishes whether the samples fit together. This could indicate whether the user could find a form of cohesion between the samples and if the user had an understanding of what the samples were meant to be. The second question establishes whether there was enough variety in the samples. If the user feels

like they are missing samples they want to use, the usage of Hitloop might be frustrating. Between the random and AI version, this will also indicate whether the AI-generated samples give more pleasing results than the random alternative.

V. Results

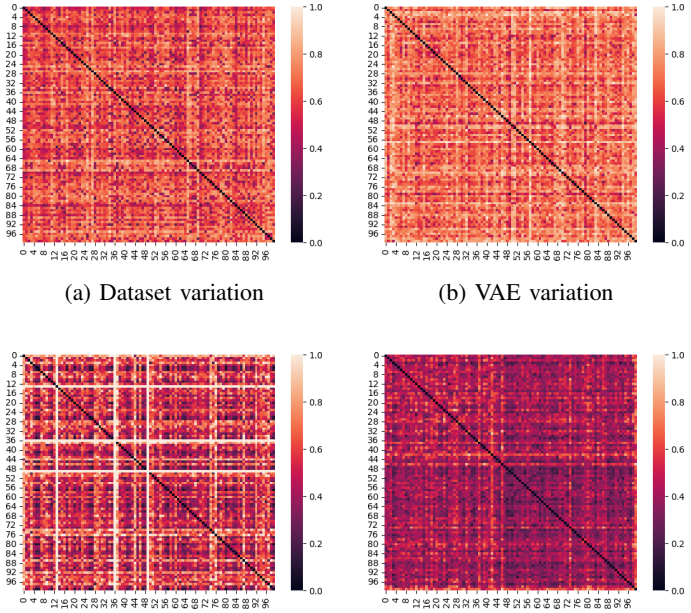
A. Quantitative test

During the quantitative test, The GAN 1, GAN 2 and VAE. The results of the Average empty ratio and Unqualified Notes are shown in Table I. The variation of the generated MIDI patterns is compared in Figure 5.

The goal of the metrics in Table I are to compare the models to the dataset with a numeric score. The score closest to the Data set, is considered best. Figure 5 displays the variation between 100 generated outputs in a heat map. When two outputs are identical, the score is 0. The best model here should display a similar level of variation as the dataset.

Model	Average empty ratio	Average Unqualified notes
Dataset	**0.287**	**0.056**
GAN 1	0.358	0.473
GAN 2	0.666	0.239
VAE	0.188	2.063

TABLE I: MIDI evaluation metrics



(a) Dataset variation (b) VAE variation

(c) Gan1 variation (d) Gan2 variation

Fig. 5: Variation in MIDI patterns between dataset and models

By looking at the outcomes, it can be established that in terms of the Average empty ratio, GAN 1 behaves most like the dataset. On the subject of unqualified notes, GAN 2 performs best. In the variation heat map, the VAE performs most like the dataset. This means every model performed best on a different test. Because a single model was required to be selected for the human evaluation, it was decided to use GAN 1. This decision was made because of two reasons. Firstly, GAN 1 performed best on the empty ratio, which means it has similar rests as the dataset. This could be influential rhythmically. It also has the second lowest amount of average unqualified notes and still displays variation in the output. Therefore it does not perform best on all tests but is the most consistently high ranking. At the same time, the other models excel in one test (VAE in variation and GAN 2 in unqualified notes) but perform worst in the other tests.

B. User test

18 users evaluated the random and AI pipeline of Hitloop. The average score on the four questions are listed in figure 6.

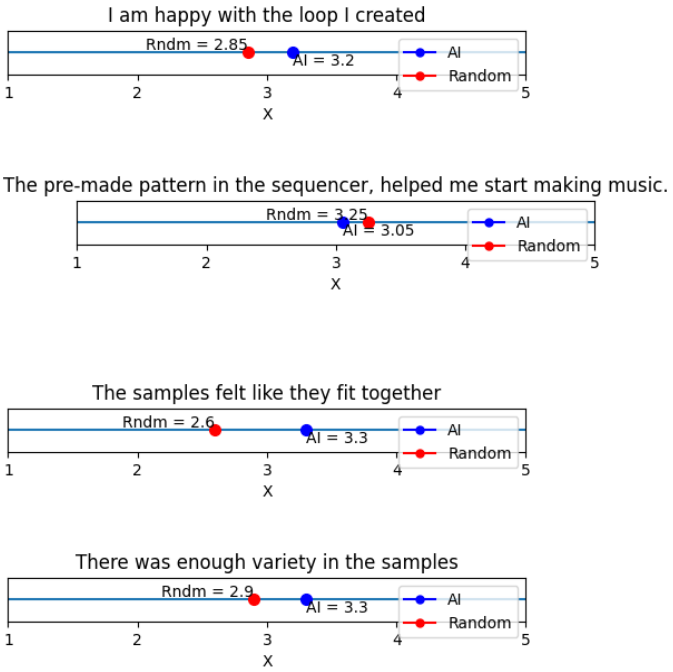


Fig. 6: Average scores of the random and AI pipeline of Hitloop

VI. Discussion

User testing revealed that AI-generated MIDI patterns might not be the superior solution for a human-in-the-loop EDM application. Although users preferred the created loop during the AI pipeline, they did not find the AI-generated patterns inspiring upon initial exploration. Therefore it can be questioned whether the MIDI generation component is essential. Because the goal is to prompt a non-expert user to create musical loops, this could be achieved with more low-tech solutions like a random system or going back to pre-deep learning rule-based systems for music generation [23]. The results seem to suggest these would perform comparable to or better than the proposed pipeline while not requiring training data. This could be a way to address the societal concerns raised during the introduction.

The sample creation pipeline was evaluated more favourably. The AI pipeline outperformed the random baseline in perceived cohesion and variety. It is also possible this could indicate that a perceived higher quality of samples results in the users creating a better musical loop. Because the AI pipeline scored higher while receiving worse pre-made patterns. To conclusively determine this, more research is required.

Out of the current pipelines, the AI pipeline seems to be preferable to the random pipeline. It, however, could benefit from utilizing the random MIDI generator. For now, the proposed pipeline can be seen as the baseline for further versions of Hitloop. With users scoring their created loop on average a 3.2, this means the pipeline assists users in creating musical loops while addressing Transparency and Human Autonomy.

VII. Conclusion

The proposed pipeline for Hitloop has achieved it's goal of addressing Transparency and Human Autonomy questions whilst assisting users in creating musical loops. While it still has room to improve, it can be considered a first step in exploring possible applications that allow for cooperation between human and AI in the creation of music.

A demonstration of the Hitloop application can be accessed at https://hitloop.responsible-it.nl/.

VIII. Future work

During development, each aspect of the proposed pipeline had at least one signalled potential improvement. Due to the scope of the research, it was unfeasible to implement them within the time frame. Research trying to emulate the proposed pipeline of Hitloop could consider these alterations to increase the results.

- For the dataset, a different drum transcription model could be utilised. During the research, Omnizart [18] was used because it was the only model that could be deployed. It, however, has the drawback that it only recognises very specific drum beats. Because drums are a major aspect of EDM [7], the model's inability to recognise more alternative drum sounds will result in the dataset not containing all rhythmic information of the original EDM track used for the dataset. This could impact the data quality. The next iteration could explore alternative options that are deploy able on current software.
- The sample creation pipeline currently extracts a sample if its cosine distance to a sample is below a certain threshold. Researchers have used a K-Nearest Neighbour model to classify instrument samples [24]. This type of model can be configured to use cosine distances to classify whether windows are close to a sample. This method would no longer require the developer to set manual thresholds. This would allow the system to train on more samples and perhaps improve the quality of the samples.
- During the training of the GAN used in Hitloop, the generator and discriminator had to be balanced by decreasing the discriminator size. This allowed the generator to learn well at the early stages but could result in a situation where the discriminator is not powered fully enough to detect the generated content of a better generator. To mitigate this, the discriminator could be increased in size. This, however, would lead to the imbalance again. Researchers have succeeded in kick-starting the generator by giving it the weights of a partially trained Variational Auto encoder. This

has resulted in GANs converging faster [25]. By applying this to the GAN for Hitloop, it could allow the discriminator to be returned to its original size without preventing the generator from learning at the start.

IX. Acknowledgments

Throughout the writing of this paper, I have received a great deal of support and assistance from various individuals at the Responsible AI lab.

At first I would like to express my gratitude to dr. Marcio Fuckner for his guidance and support throughout this research. His expertise has greatly influenced the quality of this paper and the development of Hitloop.

I also thank Yuri Westplat for guiding the Hitloop project and providing valuable input on the design and necessary components for facilitating human-AI interaction.

Lastly, I would like to mention fellow graduate students at the Responsible AI lab, Amsterdam University of Applied Sciences, for their valuable support and insightful discussions.

References

[1] AI art: I'm using a language model called GPT-2 to write my next novel - vox. [Online]. Available: https://www.vox.com/future-perfect/2019/8/30/20840194/ai-art-fiction-writing-language-gpt-2

[2] E. Cetinic and J. She, "Understanding and creating art with AI: Review and outlook," issue: arXiv:2102.09109. [Online]. Available: http://arxiv.org/abs/2102.09109

[3] Musicians are using AI to create otherwise impossible new songs. [Online]. Available: https://time.com/5774723/ai-music/

[4] J.-W. Hong and N. M. Curran, "Artificial intelligence, artists, and art: Attitudes toward artwork produced by humans vs. artificial intelligence," vol. 15, no. 2, pp. 58:1–58:16. [Online]. Available: https://doi.org/10.1145/3326337

[5] M. Senftleben, "A tax on machines for the purpose of giving a bounty to the dethroned human author – towards an AI levy for the substitution of human literary and artistic works," issue: 4123309 Place: Rochester, NY. [Online]. Available: https://papers.ssrn.com/abstract=4123309

[6] D. Williams, V. J. Hodge, and C.-Y. Wu, "On the use of AI for generation of functional music to improve mental health," vol. 3. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frai.2020.497864

[7] T. Rodgers, "On the process and aesthetics of sampling in electronic music production," vol. 8, no. 3, pp. 313–320. [Online]. Available: https://www.cambridge.org/core/journals/organised-sound/article/abs/on-the-process-and-aesthetics-of-sampling-in-electronic-music-production/236F868DD4AF8926152E6182C005E78E

[8] A. Behr, K. Negus, and J. Street, "The sampling continuum: musical aesthetics and ethics in the age of digital production," vol. 21, no. 3, pp. 223–240. [Online]. Available: https://doi.org/10.1080/14797585.2017.1338277

[9] J.-P. Briot, "From artificial neural networks to deep learning for music generation – history, concepts and trends," issue: arXiv:2004.03586. [Online]. Available: http://arxiv.org/abs/2004.03586

[10] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," issue: arXiv:1709.06298. [Online]. Available: http://arxiv.org/abs/1709.06298

[11] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," issue: arXiv:1803.05428. [Online]. Available: http://arxiv.org/abs/1803.05428

[12] G. A. C. d. Santos, A. Baffa, J.-P. Briot, B. Feijó, and A. L. Furtado, "An adaptive music generation architecture for games based on the deep learning transformer mode," issue: arXiv:2207.01698. [Online]. Available: http://arxiv.org/abs/2207.01698

[13] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, "Deep learning techniques for music generation – a survey," issue: arXiv:1709.01620. [Online]. Available: http://arxiv.org/abs/1709.01620

[14] L. Haidar-Ahmad, "Music and instrument classification using deep learning technics."

[15] K. Racharla, V. Kumar, C. B. Jayant, A. Khairkar, and P. Harish, "Predominant musical instrument classification based on spectral features," in *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 617–622. [Online]. Available: http://arxiv.org/abs/1912.02606

[16] About CC licenses. [Online]. Available: https://creativecommons.org/about/cclicenses/

[17] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," issue: arXiv:2211.08553. [Online]. Available: http://arxiv.org/abs/2211.08553

[18] Y.-T. Wu, Y.-J. Luo, T.-P. Chen, I.-C. Wei, J.-Y. Hsu, Y.-C. Chuang, and L. Su, "Omnizart: A general toolbox for automatic music transcription," vol. 6, no. 68, p. 3391, publisher: The Open Journal. [Online]. Available: https://doi.org/10.21105/joss.03391

[19] R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal, and S. Ewert, "A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[20] P. Sitikhu, K. Pahi, P. Thapa, and S. Shakya, "A comparison of semantic similarity methods for maximum human interpretability," issue: arXiv:1910.09129. [Online]. Available: http://arxiv.org/abs/1910.09129

[21] Re:vive. [Online]. Available: https://revivethis.org/

[22] Tonejs/tone.js: A web audio framework for making interactive music in the browser. [Online]. Available: https://github.com/Tonejs/Tone.js
[23] D. Herremans, C.-H. Chuan, and E. Chew, "A functional taxonomy of music generation systems," vol. 50, no. 5, pp. 1–30. [Online]. Available: http://arxiv.org/abs/1812.04186
[24] Y. Su, "Instrument classification using different machine learning and deep learning methods," vol. 34, pp. 136–142.
[25] H. Ham, T. J. Jun, and D. Kim, "Unbalanced GANs: Pre-training the generator of generative adversarial network using variational autoencoder," issue: arXiv:2002.02112. [Online]. Available: http://arxiv.org/abs/2002.02112

X. Supplements

Generator

TABLE II: GAN Generator 3.967,617 params

Layer (type)	Output Shape
dense (Dense)	(None, 8192)
batch_normalization	(None, 8192)
leaky_re_lu	(None, 8192)
reshape	(None, 8, 2, 512)
dropout	(None, 8, 2, 512)
conv2d_transpose	(None, 16, 4, 256)
batch_normalization_1	(None, 16, 4, 256)
leaky_re_lu_1	(None, 16, 4, 256)
dropout_1	(None, 16, 4, 256)
conv2d_transpose_1	(None, 32, 4, 128)
batch_normalization_2	(None, 32, 4, 128)
leaky_re_lu_2	(None, 32, 4, 128)
conv2d_transpose_2	(None, 128, 20, 128)
batch_normalization_3	(None, 128, 20, 128)
leaky_re_lu_3	(None, 128, 20, 128)
conv2d (Conv2D)	(None, 128, 20, 1)

Discriminator

TABLE III: GAN Discriminator 1.840,065 params

Layer (type)	Output Shape
dense (Dense)	(None, 8192)
batch_normalization	(None, 8192)
leaky_re_lu	(None, 8192)
reshape	(None, 8, 2, 512)
dropout	(None, 8, 2, 512)
conv2d_transpose	(None, 16, 4, 256)
batch_normalization_1	(None, 16, 4, 256)
leaky_re_lu_1	(None, 16, 4, 256)
dropout_1	(None, 16, 4, 256)
conv2d_transpose_1	(None, 32, 4, 128)
batch_normalization_2	(None, 32, 4, 128)
leaky_re_lu_2	(None, 32, 4, 128)
conv2d_transpose_2	(None, 128, 20, 128)
batch_normalization_3	(None, 128, 20, 128)
leaky_re_lu_3	(None, 128, 20, 128)
conv2d (Conv2D)	(None, 128, 20, 1)

VAE Decoder (Encoder is mirrored) [h]

TABLE IV: VAE Decoder: 4.382.493 params

Layer (type)	Output Shape #
input_2 (InputLayer)	(None, 128)
dense (Dense)	(None, 4096)
batch_normalization_6 (Batch	(None, 4096)
leaky_re_lu_6 (LeakyReLU)	(None, 4096)
reshape (Reshape)	(None, 4, 2, 512)
dropout_5 (Dropout)	(None, 4, 2, 512)
conv2d_transpose (Conv2DTran	(None, 8, 2, 256)
batch_normalization_7 (Batch	(None, 8, 2, 256)
leaky_re_lu_7 (LeakyReLU)	(None, 8, 2, 256)
dropout_6 (Dropout)	(None, 8, 2, 256)
conv2d_transpose_1 (Conv2DTr	(None, 16, 2, 256)
batch_normalization_8 (Batch	(None, 16, 2, 256)
leaky_re_lu_8 (LeakyReLU)	(None, 16, 2, 256)
dropout_7 (Dropout)	(None, 16, 2, 256)
conv2d_transpose_2 (Conv2DTr	(None, 32, 2, 128)
batch_normalization_9 (Batch	(None, 32, 2, 128)
leaky_re_lu_9 (LeakyReLU)	(None, 32, 2, 128)
dropout_8	(None, 32, 2, 128)
conv2d_transpose_3	(None, 64, 10, 64)
batch_normalization_10	(None, 64, 10, 64)
leaky_re_lu_10	(None, 64, 10, 64)
dropout_9 (Dropout)	(None, 64, 10, 64)
conv2d_transpose_4 (Conv2DTr	(None, 128, 20, 32)
batch_normalization_11 (Batc	(None, 128, 20, 32)
leaky_re_lu_9 (LeakyReLU)	(None, 128, 20, 32)
conv2d_transpose_5	(None, 128, 20, 1)

TABLE V: Instrument Samples used in dataset

Internal sample name	Link to audio file
Kick-1	https://commons.wikimedia.org/wiki/File:Punch_Kick.wav
Kick-2	https://commons.wikimedia.org/wiki/File:Hardstyle_kick.wav
Hihat-1	https://commons.wikimedia.org/wiki/File:Hi-Hat_Abierto.ogg
Hihat-2	https://commons.wikimedia.org/wiki/File:Hi-Hat_Cerrado.ogg
Snare-1	https://commons.wikimedia.org/wiki/File:Redoblante.ogg
Snare-2	https://commons.wikimedia.org/wiki/File:Cajon_Peruano_Parche.ogg
Bass-1	https://commons.wikimedia.org/wiki/File:Bass_loop_1_(Carrai_Pass).wav (1st note)
Bass-2	https://commons.wikimedia.org/wiki/File:Bass_loop_2_(Carrai_Pass).wav (1st note)

TABLE VI: Songs used in dataset

Song Name	Artist	CC-License
Higher Than The Sky	Infraction	http://creativecommons.org/licenses/by-nc-sa/3.0/
Off Guard	Anitek	http://creativecommons.org/licenses/by-nc-sa/3.0/
Feel My Energy	Infraction	http://creativecommons.org/licenses/by-nc-sa/3.0/
RnB Chill Trap	Sevennotes	http://creativecommons.org/licenses/by/3.0/
Soulful Clouds	LEXMusic	http://creativecommons.org/licenses/by-nc/3.0/
Walking Bird	Anitek	http://creativecommons.org/licenses/by-nc-sa/3.0/
Immersion	Vulpey	http://creativecommons.org/licenses/by/3.0/
Wind Starts	Lysergic Tempo	http://creativecommons.org/licenses/by-nc-sa/3.0/
The Forbidden Door Pt.2	Woochia	http://creativecommons.org/licenses/by-nc-sa/3.0/
back to abnormal	Delectable Mosquito	http://creativecommons.org/licenses/by-nc-sa/3.0/
Stomp and Claps	Alexiaction	http://creativecommons.org/licenses/by-nc-sa/3.0/
Nothing	kiyo	http://creativecommons.org/licenses/by-nc-sa/3.0/
Whisper	Ramp	http://creativecommons.org/licenses/by-nc-sa/3.0/
Melody	J.L.T	http://creativecommons.org/licenses/by-nc-sa/3.0/
Bad Chick	The.madpix.project	http://creativecommons.org/licenses/by-nc-sa/3.0/
Bird of Paradise	Capashen	http://creativecommons.org/licenses/by-nc-sa/3.0/
Cool Man	Marco Margna	http://creativecommons.org/licenses/by-sa/3.0/
Trance	Chaz Robinson	http://creativecommons.org/licenses/by-sa/3.0/
I Dont Cry	A Virtual Friend	http://creativecommons.org/licenses/by/3.0/
Breathe	AxlampArth	http://creativecommons.org/licenses/by/3.0/
The light of my life	DID	http://creativecommons.org/licenses/by-nc-sa/3.0/
Who's my Baby	Capashen	http://creativecommons.org/licenses/by-nc-sa/3.0/
Superimposed	Peyo	http://creativecommons.org/licenses/by-nc-sa/3.0/
I'm Bringing My Cat	Mark Skinner	http://creativecommons.org/licenses/by-nc
Bust That Bust This	Professor Kliq	http://creativecommons.org/licenses/by-nc-sa/3.0/
Erotic Robotics	The Polish Ambassador	http://creativecommons.org/licenses/by-nc-sa/3.0/
Electro funk	Index	http://creativecommons.org/licenses/by-sa/3.0/
Dream atmosphere	MKHLSDRV	http://creativecommons.org/licenses/by-sa/3.0/
Virus	Infraction	http://creativecommons.org/licenses/by-nc-sa/3.0/
Gold Sequence	Sound Creator	http://creativecommons.org/licenses/by-nc-sa/3.0/
The Deep	Anitek	http://creativecommons.org/licenses/by-nc-sa/3.0/
Dont Look Back	Infraction	http://creativecommons.org/licenses/by-nc-sa/3.0/
Syberia	GroovyVoxx	http://creativecommons.org/licenses/by-nc-sa/3.0/
Mumbai	Soundrider Dope	http://creativecommons.org/licenses/by/3.0/
Ouest West	Mr.Ju	http://creativecommons.org/licenses/by-sa/3.0/
Roentgen	Sonic Radiation	http://creativecommons.org/licenses/by-nc-sa/3.0/
NINJASLIPDECOMBAT	Mister Electric Demon	http://creativecommons.org/licenses/by-nc/3.0/
DJ Alvin	Gianluca Luppi	http://creativecommons.org/licenses/by-nc-sa/3.0/
Come On Join In	Anydoll	http://creativecommons.org/licenses/by-nc-sa/3.0/
Visual Bastards	Pablo Hardway	http://creativecommons.org/licenses/by-sa/3.0/
Dirty angel	The Phase	http://creativecommons.org/licenses/by-sa/3.0/
My World	WE ARE FM	http://creativecommons.org/licenses/by-nc-sa/3.0/
SDRNR Original Version	The Easton Ellises	http://creativecommons.org/licenses/by-nc-sa/3.0/
Day	Sunwill	http://creativecommons.org/licenses/by-nc-sa/3.0/
Mad Man	Ortoped	http://creativecommons.org/licenses/by-nc-sa/3.0/
Bust This Bust That Second Movement	Professor Kliq	http://creativecommons.org/licenses/by-nc-sa/3.0/
Dance with me	SAMSHARA	http://creativecommons.org/licenses/by-nc-sa/3.0/
Blink Of An Eye	Shon Tha Phenom	http://creativecommons.org/licenses/by-sa/3.0/
F The King	F The KingDouble	http://creativecommons.org/licenses/by-nc-sa/3.0/
Internorthernalz	Internorthernalz	http://creativecommons.org/licenses/by/3.0/
Its Alright	Its Alright Elephant Funeral	http://creativecommons.org/licenses/by/3.0/
F The King	Double	http://creativecommons.org/licenses/by-sa/3.0/

FREE21	MAGMA	http://creativecommons.org/licenses/by-nc-sa/3.0/
Been Waiting feat. Rebekka Salomea	Jaief amp Asong	http://creativecommons.org/licenses/by-nc-sa/3.0/
On the Come Up	Wordsmith	http://creativecommons.org/licenses/by/3.0/
Mellow Dreams	Mindfull	http://creativecommons.org/licenses/by-sa/3.0/
No Es Capricho	El Corma	http://creativecommons.org/licenses/by-sa/3.0/
summertimewav	Emcee Lynx	http://creativecommons.org/licenses/by-nc-sa/3.0/
Movie Chiappe	P.Yo	http://creativecommons.org/licenses/by-nc-sa/3.0/
The Tester	Dave Foster	http://creativecommons.org/licenses/by-sa/3.0/
Love Me Notts	Hazernomical	http://creativecommons.org/licenses/by-sa/3.0/
Pourin Moe by The East Side Hustlas	Cartel	http://creativecommons.org/licenses/by-sa/3.0/
Collective	Branton	http://creativecommons.org/licenses/by-nc-sa/3.0/
El Cambio Definitivo ft. Dj Rihla	Picatoste	http://creativecommons.org/licenses/by-nc-sa/3.0/