ReStyle-MusicVAE: Enhancing User Control of Deep Generative Music Models with Expert Labeled Anchors

Damjan Prvulovic
TU Wien, Faculty of Informatics
Vienna, Austria
damjan.prvulovic@tuwien.ac.at

Richard Vogl
TU Wien, Faculty of Informatics
Vienna, Austria
richard.vogl@tuwien.ac.at

Peter Knees
TU Wien, Faculty of Informatics
Vienna, Austria
peter.knees@tuwien.ac.at

ABSTRACT

Deep generative models have emerged as one of the most actively researched topics in artificial intelligence. An area that draws increasing attention is the automatic generation of music, with various applications including systems that support and inspire the process of music composition. For these assistive systems, in order to be successful and accepted by users, it is imperative to give the user agency and express their personal style in the process of composition.

In this paper, we demonstrate ReStyle-MusicVAE, a system for human-AI co-creation in music composition. More specifically, ReStyle-MusicVAE combines the automatic melody generation and variation approach of MusicVAE and adds semantic control dimensions to further steer the process. To this end, expert-annotated melody lines created for music production are used to define stylistic anchors, which serve as semantic references for interpolation. We present an easy-to-use web app built on top of the Magenta.js JavaScript library and pre-trained MusicVAE checkpoints.

CCS CONCEPTS

• **Applied computing** → **Sound and music computing**; • **Human-centered computing** → **Collaborative interaction**.

KEYWORDS

music generation, user control, variational auto encoder

ACM Reference Format:
Damjan Prvulovic, Richard Vogl, and Peter Knees. 2022. ReStyle-MusicVAE: Enhancing User Control of Deep Generative Music Models with Expert Labeled Anchors. In *Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '22 Adjunct), July 4–7, 2022, Barcelona, Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3511047.3536412

1 INTRODUCTION

Music composition is nowadays most commonly taking place in fully digital environments, so-called digital audio workstations (DAW). Often performances are directly recorded in a symbolic format (usually MIDI) and then manually edited or modified to meet the composer's creative ideas. Coming up with new musical ideas can, however, be a challenging task, especially when experiencing writer's block. In such situations, creative tools, i.e. automatic music generation systems—more precisely systems for *assisted music generation* [3], may help a composer find inspiration or speed up the workflow.

In this work, we deal with deep generative models for music generation, which are a trending topic in artificial intelligence. The objective of such generative music techniques is to automatically learn musical styles and to generate new musical content based on the given learning material [2]. As opposed to handcrafted models, the benefit of using deep learning methods to generate music is their flexibility. The system can automatically learn a musical style from a corpus and create new musical content, without explicitly being taught music theoretical rules or structure. In addition to generating qualitative musical content, current challenges concern automatic curation, user control, and personalization.

We present ReStyle-MusicVAE, a system for human-AI co-creation in music composition. More specifically, ReStyle-MusicVAE combines the automatic melody generation and variation approach of MusicVAE (see Section 2) and adds further semantic control dimensions to steer the generation process toward stylistic anchors. This adds a lightweight manipulation mechanism, giving the user a higher degree of control. We make use of expert-annotated melody lines created for music production to define the stylistic anchors. Furthermore, the user can extend the set of semantic anchors with their own provided examples, providing additional flexibility and adaptation.

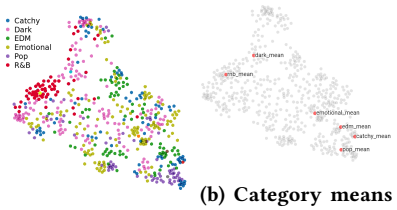
2 RELATED WORK

Over the last years, generative music systems have increasingly drawn interest, driven by major developments that reflect the trends in neural network developments, most notably, MusicVAE [11] which embeds melodies in a latent space, allowing generation by means of interpolation; Music Transformer [7], built upon self-attention mechanisms and allowing for generating longer musical sequences by modeling structure, and OpenAI Jukebox [4], which generates music in the raw audio domain. At the same time, there is a rising interest from HCI researchers in trying to investigate what people need in a human-AI co-creation process and how intelligent user interfaces can be designed to empower people to realize their creative goals. The first findings show that users encounter problems such as information overload and non-deterministic output. They desire greater agency, control, and a sense of authorship vis-a-vis the AI during co-creation [6, 9, 10, 12].

One promising deep generative model for assisted music generation is the variational autoencoder (VAE). Roberts et al. [11] already show that their MusicVAE provides some control or personalization

UMAP '22 Adjunct, July 4–7, 2022, Barcelona, Spain

ACM ISBN 978-1-4503-9232-7/22/07.
https://doi.org/10.1145/3511047.3536412



(a) All embeddings. (b) Category means only.

Figure 1: 2D t-SNE projection of the 2-bar embeddings. Points: 545 | Dimension: 256 - Perplexity: 50 | Learning rate: 10

mechanisms to steer the generation of new musical content. This works mainly by manipulating the latent vector through various operations like interpolation and averaging or attribute modification.

Although a big advantage of deep generative models is their generality, i.e. the ability to use the same system for various musical genres or styles, it usually requires retraining to be able to get meaningful style differences in the generated content. Retraining a model from scratch is, however, very expensive and time-consuming due to hyperparameter tuning, and requires a large volume of quality data. These circumstances pose a challenge for non-technical users when they want to control or personalize a deep generative model. As a response, systems aim at adapting pre-trained systems to avoid these drawbacks while enabling personalization (MidiMe [5]) or at keeping the process simple by restricting content generation to particular voices, giving more control when composing (Cococo [9]).

Similar to MidiMe, the aim of this paper is to provide a method for controlling and personalizing a pre-trained VAE music model. Dealing this way with pre-trained VAE models brings limitations: since model structure optimization, loss function modification, or hyperparameter tuning is not an option, the remaining controlling point represents the given latent space. Therefore, the goal was to analyze if and how musical styles or genres can be embedded in pre-trained MusicVAE latent spaces, to further control the generation of new melodic lines with it. Embeddings can form semantically meaningful clusters in the latent space, cf. [8]. The underlying hypothesis is that the MusicVAE model will capture useful musical patterns from a style-labeled dataset and place the embeddings in style-separated clusters. In contrast to MidiMe and Cococo, we do not attempt to reduce the complexity of the embedding space through another projection or focus on control of individual voices, resp., but rather aim for incorporating information from expert annotated ground truth to provide a flexible number of anchor points and steering of the composition.

3 METHODOLOGY AND IMPLEMENTATION IN USER INTERFACE

For this work, we are interested in embedding and localizing musical style/genre in a pre-trained MusicVAE latent space. To this end, an expert-labeled melody dataset, namely Niko's MIDI Pack,[1]

[1] https://shop.pianoforproducers.com/niko-midi-pack-2

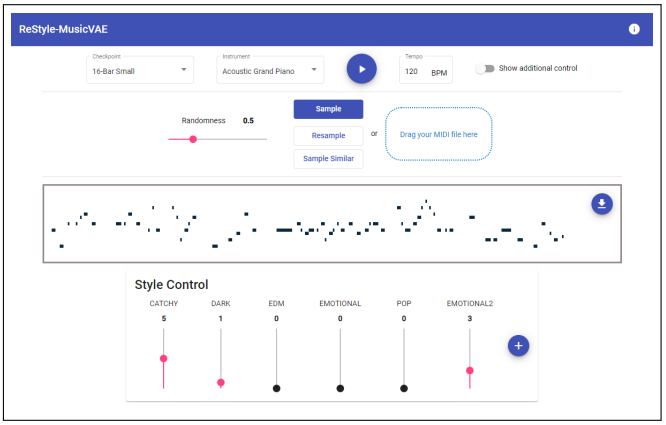


Figure 2: ReStyle-MusicVAE's user interface for real-time human-AI co-creation: Users first start with an initial melody, then tweak the style control sliders to modify the melody. The melody gets dynamically adjusted as the sliders are moved.

is embedded in 4 different MusicVAE checkpoints that mainly differ in melody sequence length.[2] For the experiments, we ignore included accompaniments and make use of melodies in the key of C Major or A Minor (relative minor) as stylistic information is considered key independent. All melodies are in 4/4 time. The MIDI dataset contains 84 unique melodies in total and is categorized in 6 styles/genres: Catchy, Dark/Hip Hop/Trap, EDM, Emotional, Pop, R&B/Neosoul. Since MusicVAE has a fixed input and output length, each melody sequence is split into smaller chunks prior to embedding, in order to fit the corresponding checkpoints bar-length. Because the embedding space is high-dimensional, i.e. in our case 256-dimensional, we use t-SNE [13] to manually inspect its structure. Figure 1 shows a 2D projection of the 2-bar embeddings colored by category.

Based on the findings from the embedding space analysis, ReStyle-MusicVAE was developed into a user interface that can be run as a web app (see Figure 2 and https://restyle-musicvae.web.app/ for a deployed instance).[3] Users start by setting an initial melody by either *a)* sampling a random sequence from the model or by *b)* uploading a melody. The initial melody can be tweaked using the *Style Control* sliders, resulting in dynamic adjustment of the melody. Additionally, users can upload their own monophonic melodies (preferable in C Major or A Minor) and let the system extract their style to create additional personalized style sliders (+ button), change the model checkpoint (which mainly sets the length of the melody sequence), select the used instrument samples, and set the tempo for the playback. In more detail, the technical aspects of each UI element are as follows:

[2] mel_2bar_small, mel_4bar_med_q2, mel_4bar_med_lokl_q2, and mel_16bar_small_q2. A full list of all available Magenta.js music checkpoints can be found at: https://goo.gl/magenta/js-checkpoints

[3] The user interface is built on top of the Magenta.js JavaScript library (https://github.com/magenta/magenta-js/tree/master/music), which is in turn powered by TensorFlow.js; Angular v10.2.0; Angular Material v10.2.7 (UI Elements); and Magenta Music v1.23.1. The source code of the interface is available at https://github.com/damjanprvc/style-embedded-musicvae.

- *Sample* samples a new random 256-dimensional latent vector from a unit Gaussian and decodes it.
- The *Randomness* slider controls the stochastic softmax output layer of the decoder (useful for the resampling functionality). The amount varies between 0.0 to 2.0 in steps of 0.1. 0.0 sets the softmax output layer to act deterministically, i.e. the notes with the highest probability get chosen strictly and the same latent vector will produce always the same output. This functionality is provided by Magenta.js as *Temperature*, however, we renamed it to be better understandable by non-technical users.
- *Resample* decodes the current latent vector again (works best in combination with the 'Randomness Slider').
- *Sample Similar* creates similar melodies by sampling a new latent vector from a unit Gaussian and interpolating between the current melody (latent vector) and the newly sampled latent vector. An interpolation near the current latent vector is chosen to preserve the similarity of the current melody.
- The *Style Control* sliders represent the main control mechanism. Each style slider represents the arithmetic mean of the respective melody snippets ("anchor") corresponding to the musical style (see above). Style manipulation is done by gradually adding the 256-dimensional anchor vector to the existing melody sequence, which is represented as a 256-dimensional embedding vector as well. The amount varies between 0.0 to 1.0 in steps of 0.1 and is controlled by the user via the sliders. For the UI this factor is multiplied by 10.

4 INTERPRETATION AND DISCUSSION

The initial assumption when using expert labeled and prototypical melodies was that the embeddings will form visibly separated clusters based on their style. With a few exceptions, this is not the case. The embeddings are generally more scattered than organized in style clusters. This is especially visible with the 2-bar and 4-bar checkpoints. Three reasons (at least) could be responsible for this: 1) defining the style or genre a melody belongs to is a rather subjective task. So even when the melodies are labeled or created by an expert, one cannot say with certainty that the given ground truth of the melodies i.e. the given style categorization is "correct". Melodies or music, in general, can usually be categorized into more than one genre or style, which often results in expert disagreement [1]. In addition, there is a semantic gap between the human perceived musical qualities and the low-level music representation suitable for a deep learning model [14], esp. when dealing with a symbolic representation of melody lines only. The one-hot representation used for this model is only capable of capturing mid-level musical information as notes played, note length, note density, range, scales, etc.; 2) 2-bar and 4-bar music snippets often cannot capture meaningful stylistic information, because of the short length; and 3) the pre-trained models may not be trained with data corresponding to the styles used in the dataset.

5 USER STUDY

Following the suggestion by Briot et al. [2, p. 247], we qualitatively evaluate our assistive music generation system based on the satisfaction of *composers*. Thus we conducted a pilot study with expert composers recruited via the public Discord server *"Make Music Income"*. Participants ($n = 4$, all male) have been asked to test the composing tool and then fill out an online questionnaire consisting of 8 questions to be answered via a 5-point Likert scale (cf. Table 1), 6 open-ended questions, and a final closed-ended question on if the participant would keep using the tool.

#	Question	mean ± std
1	The composing tool was easy to understand.	4.20 ± 0.50
2	Interacting with the composing tool was easy.	4.00 ± 0.82
3	The tool was performant (i.e. fast).	4.00 ± 0.82
4	This tool helped me come up with new melodic ideas.	3.00 ± 0.82
5	I would use this tool.	2.75 ± 1.26
6	This tool would speed up my composing workflow.	3.25 ± 0.96
7	The 'Style Controls' changed the melodies adequate & meaningful.	3.25 ± 0.50
8	I experienced a sense of control vis-a-vis the composing tool.	4.00 ± 0.82

Table 1: Likert scale questions with the average score and standard deviation for each question where 1 represents "strongly disagree" and 5 represents "strongly agree".

In brief, we see strengths of the interface and tool in terms of usability and sense of control, and individual difficulties when it comes to the interpretability and expectation of the semantic sliders. Main take-aways from the pilot study are:

(1) the UI enables good interaction with the composition process and gives participants a sense of control;
(2) the style controls do not provide the expected outputs, but to some feel rather *"random"* (P2);
(3) the tool may help composers come up with new ideas or be used as an "inspiration tool";
(4) one out of four participants would keep using the tool in its current form;
(5) the overall feedback regarding AI music generation tools is positive (note that this might stem from a selection bias): *"love the topic and am looking forward to more AI trends to come"* (P2).

With the sample of the pilot study being very small and comprising subjects not acquainted with the workflow referred to, the meaningfulness of the study is limited. However, we see the underlying idea of the interface supported, with further potential in advancing the procedure to (semi-)automatically derive semantic labels and anchors.

6 CONCLUSION AND FUTURE WORK

An interactive melody composing tool for human-AI co-creation with semantic controls was presented. As a basis, a method for inexpensive music style modification using embedding vector manipulation was proposed. Despite the prototypicality of the used expert labels, clear clusters could not be consistently identified when using the pre-trained embeddings by MusicVAE. This mismatch led to some less meaningful style controls. In future work, individual assignment and user preference could play a more pronounced role over the expert labels. The proposed method has, however, shown to be valuable for real-time human-AI interaction. It gives users a sense of control vis-a-vis the AI, and helps composers come up with new (unconventional) musical ideas.

ACKNOWLEDGMENTS

This research received support by the Austrian Science Fund (FWF): P33526.

REFERENCES

[1] Jean-Julien Aucouturier and François Pachet. 2003. Representing Musical Genre: A State of the Art. *Journal of New Music Research* 32, 1 (2003), 83–93. https://doi.org/10.1076/jnmr.32.1.83.16801 arXiv:https://www.tandfonline.com/doi/pdf/10.1076/jnmr.32.1.83.16801

[2] Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. 2020. *Deep Learning Techniques for Music Generation*. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-70163-9_1

[3] Filippo Carnovalini and Antonio Rodà. 2020. Computational Creativity and Music Generation Systems: An Introduction to the State of the Art. *Frontiers in Artificial Intelligence* 3 (2020). https://doi.org/10.3389/frai.2020.00014

[4] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A Generative Model for Music. *arXiv preprint arXiv:2005.00341* (2020). https://doi.org/10.48550/ARXIV.2005.00341

[5] Monica Dinculescu, Jesse Engel, and Adam Roberts. 2019. MidiMe: Personalizing a MusicVAE model with user data. In *Workshop on Machine Learning for Creativity and Design, NeurIPS*.

[6] Florian Grote, Kristina Andersen, and Peter Knees. 2015. Collaborating with Intelligent Machines: Interfaces for Creative Sound. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI EA '15)*. Association for Computing Machinery, New York, NY, USA, 2345–2348. https://doi.org/10.1145/2702613.2702650

[7] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. 2018. Music Transformer. *arXiv preprint arXiv:1809.04281* (2018). https://doi.org/10.48550/ARXIV.1809.04281

[8] Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, Vol. 2. 302–308. https://doi.org/10.3115/v1/P14-2050

[9] Ryan Louie, Andy Coenen, Cheng Zhi Huang, Michael Terry, and Carrie J Cai. 2020. Novice-AI music co-creation via AI-steering tools for deep generative models. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3313831.3376739

[10] Ryan Louie, Jesse Engel, and Anna Huang. 2021. Expressive Communication: A Common Framework for Evaluating Developments in Generative Models and Steering Interfaces. arXiv:2111.14951 [cs.HC] https://arxiv.org/abs/2111.14951

[11] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. 2018. A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 4364–4373. https://proceedings.mlr.press/v80/roberts18a.html

[12] Minhyang Suh, Emily Youngblom, Michael Terry, and Carrie J Cai. 2021. AI as Social Glue: Uncovering the Roles of Deep Generative AI during Social Music Composition. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3411764.3445219

[13] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[14] Òscar Celma, Perfecto Herrera, and Xavier Serra. 2006. Bridging the Music Semantic Gap. In *ESWC 2006 Workshop on Mastering the Gap: From Information Extraction to Semantic Representation*. http://hdl.handle.net/10230/34294