Pattern Atlas: Exploring Personal Rhythm Spaces in Ableton Live with a Transformer-Based Variational Autoencoder

Alexander Lunt
Nuremberg University of Music
Nuremberg, Germany
alexander.lunt@hfm-nuernberg.de

Sebastian Trump
Nuremberg University of Music
Nuremberg, Germany
sebastian.trump@hfm-nuernberg.de

Abstract

This paper presents *Pattern Atlas*, an interactive pattern generation system for Ableton Live designed for the performative exploration of personal rhythm spaces. Combining a transformer-based variational autoencoder with a Max for Live interface, the system allows users to load their own MIDI clips as anchor patterns in a two-dimensional control space and to generate new rhythms through interpolation between them. Pattern Atlas thus connects the broader musical knowledge of a pre-trained rhythm model with the specificity of user-owned material. Its representation is velocity-sensitive and models microtiming through learned note-wise timing offsets, enabling transformations not only of rhythmic structure but also of groove-related temporal nuance. The paper outlines the conceptual motivation, model design, interactive control approach and Ableton Live implementation of the system. Furthermore the paper discusses the use of Pattern Atlas in artistic practice as a preliminary evaluation of performable rhythm-space interaction.

1 INTRODUCTION

Advances in deep learning have significantly expanded the possibilities of symbolic music generation, including the generation of melodies, harmonies, and rhythmic structures (Eck and Schmidhuber, 2002; Roberts et al., 2018; Huang et al., 2018). In rhythm generation in particular, latent generative models have become an important paradigm because they do not only reproduce musical patterns, but also provide structured spaces for variation, interpolation, and navigation (Roberts et al., 2018; Tokui, 2020). This makes them especially relevant for interactive music systems in which performers explore and transform musical material in real time. At the same time, however, the creative usefulness of such systems depends not only on their technical ability to generate plausible patterns, but on whether they provide musically rich, surprising, and controllable results in artistic practice.

A central limitation of current VAE-based rhythm generation approaches is that, while they often enable smooth interpolation between learned patterns, they can remain comparatively narrow in rhythmic richness and creative unpredictability. In Margaret Boden's distinction between combinational and exploratory creativity, such systems often operate convincingly at the level of recombination, but less strongly at the level of opening new and artistically meaningful spaces of exploration (Boden, 2004). This issue becomes particularly relevant in live and improvised performance contexts, where performers require not only stylistically coherent output, but also variation that remains musically nuanced, responsive, and open to personal artistic direction. As Yee-King argues, latent spaces can be understood not merely as technical embeddings, but as creative spaces whose usefulness depends on how they are made accessible and navigable in practice (Yee-King, 2022).

Prior work has established important foundations for this area. Early neural approaches demonstrated that sequential models could capture temporal structure in musical material (Eck and Schmidhuber, 2002). Variational autoencoders later introduced the possibility of learning continuous latent representations that support interpolation and structured variation, with MusicVAE becoming a particularly influential example for symbolic music generation (Roberts et al., 2018). Transformer architectures further improved sequence modeling by capturing longer-range dependencies and more complex temporal relations through attention mechanisms (Vaswani et al., 2017; Huang et al., 2018). For interactive rhythm generation specifically, Tokui's RhythmVAE demonstrated the artistic potential of integrating a latent rhythm generator directly into a DAW plugin workflow (Tokui, 2020). This was an important step toward democratizing AI-supported rhythm generation and making latent-space exploration available in live and production contexts.

At the same time, several challenges remain. First, earlier systems often provide limited rhythmic richness in the generated material, particularly when event structure, instrumentation, and temporal nuance are constrained. Second, interaction with latent spaces can remain comparatively opaque, especially when users have little influence over how musical material is arranged or accessed. Third, the relationship between broader pre-trained models and artist-owned musical material remains insufficiently developed. Earlier work has addressed some of these issues primarily at the level of interaction and mapping, focusing on latent-space navigation through musical action and on emergent live interfaces for AI music systems (Lunt and Trump, 2023; Trump and Lunt, 2025).

Pattern Atlas addresses these issues through an interactive pattern generation system implemented as a Max for Live device for Ableton Live. The system combines a pre-trained rhythm model with personalization through user-owned MIDI clips, which can be loaded directly into the interface and used as anchor patterns for interpolation. In this way, the approach continues concerns associated with small-data and artist-centered generative systems, including personalization and data ownership (Vigliensoni et al., 2020, 2022), while combining them with the broader musical coverage of a larger learned model. In addition, the system aims at musically richer output by explicitly modeling microtiming as learned note-wise timing offsets. This allows groove-related temporal deviations to become part of both generation and interpolation, rather than being reduced to strictly quantized patterns. The primary artistic context of the system is live performance, although it is equally relevant for exploratory composition and studio-based experimentation.

This paper presents *Pattern Atlas* as an interactive pattern generation system and Max for Live device for generative AI performance. It describes how the system works conceptually and technically, and how it can be used in artistic practice. More specifically, the paper introduces a transformer-based VAE architecture for rhythm generation that integrates improved sequence modeling with a deployable live-performance workflow; a personal control-space approach in which user-owned MIDI patterns can be encoded and explored through interpolation; and a representation that is velocity-sensitive and supports explicit microtiming control. Together, these elements aim to make latent rhythm generation more musically legible, more performable, and more artistically promising in live electronic music practice.

2 PATTERN ATLAS

2.1 Concept and Motivation

Pattern Atlas is designed as an interactive pattern generation system for live performance in Ableton Live. Its main goal is not autonomous generation in isolation, but musically usable generation within an existing artistic workflow. In this context, the key requirement is not only plausible output, but a system that is fast, understandable, and adaptable enough to support real-time artistic decision-making.

The design is motivated by the observation that existing latent rhythm interfaces can be compelling as demonstrations of interpolation, but often remain limited as performative tools in extended artistic use. RhythmVAE is an important reference point here, as it showed how variational autoencoders can be integrated into a DAW plugin for interactive rhythm generation (Tokui, 2020). For the present work, however, artistic practice suggested three additional requirements: richer rhythmic output, more flexible spatial control, and stronger integration of user-owned musical material.

The core interaction principle is a two-dimensional XY space into which users load their own MIDI rhythms as anchor patterns. These anchors serve as the basis for generating new rhythms through interpolation. As in RhythmVAE and Beat Blender, interpolation is central to the experience of the system (Tokui, 2020; Google Creative Lab and Magenta, 2018). *Pattern Atlas* extends this paradigm, however, by allowing more flexible spatial arrangement and by treating the interface as a personal control space rather than a fixed map of model relations. The learned latent space remains the generative substrate, but the XY interface becomes a performable arrangement layer through which users shape how rhythmic relations are explored in practice. This perspective also relates to earlier work that frames latent-space traversal as an interactive musical process rather than as a purely technical operation (Pati et al., 2019).

A further design motivation concerns rhythmic representation. Pattern Atlas aims for musically richer output by explicitly modeling microtiming as learned note-wise timing offsets. This is not only a technical refinement, but a musical one: rhythmic identity depends not only on onset structure, but also on timing nuance, articulation, and groove. Additionally, relating personal pattern with a pretrained model can broaden stylistic diversity in the generated output.

The integration of transformer components into the VAE architecture follows the same logic. Compared with earlier model versions, the transformer-based redesign was introduced to improve sequence understanding, rhythmic coherence, and latent-space organization (Vaswani et al., 2017; Huang et al., 2018), while avoiding the practical limitations of fully transformer-based real-time generation. Pattern Atlas is therefore conceived not as a standalone model, but as an interactive generative tool for producing new rhythms from given patterns within a familiar music-production environment.

2.2 Model Architecture and Data Representation

The generative core of Pattern Atlas is a variational autoencoder with transformer-based encoder and decoder components. The model is designed for two-bar symbolic rhythm-pattern generation while preserving the latent-space properties needed for interpolation and interactive exploration. It therefore combines the advantages of VAE-based latent representations for rhythmic variation (Roberts et al., 2018; Tokui, 2020) with transformer-based sequence modeling for capturing temporal dependencies in musical material (Vaswani et al., 2017; Huang et al., 2018). More recent symbolic music work has also combined transformer architectures with VAE objectives, suggesting that this hybrid design can support both stronger sequence modeling and controllable latent representations (Wu and Yang, 2021). An overview of the architecture is shown in Figure 1.

Given an input rhythm sequence x, the encoder maps it to a latent distribution

$$q_\phi(z \mid x) = \mathcal{N}(\mu(x), \sigma(x))$$

where μ and σ denote the mean and variance parameters predicted by the encoder. A latent vector z is sampled via the reparameterization trick and passed to the decoder, which models the conditional reconstruction distribution

$$p_\theta(x \mid z).$$

This formulation is important for interactive rhythm generation because it encourages a structured latent space in which related rhythmic patterns occupy nearby regions and can therefore be interpolated smoothly. For deterministic inference, the latent mean can be used directly, i.e., $z = \mu$.

The current architecture uses four transformer layers in the encoder and four in the decoder, each with four attention heads. Test versions with six layers per component have also been explored. The latent space has 512 dimensions, which has so far proven sufficient for representing pitch, velocity, and microtiming, and is also consistent with related systems such as MusicVAE (Roberts et al., 2018). The decoder is autoregressive and conditioned on the latent representation through cross-attention. Teacher forcing is used during training.

The transformer-based redesign is the main modification compared with earlier model versions. It was introduced

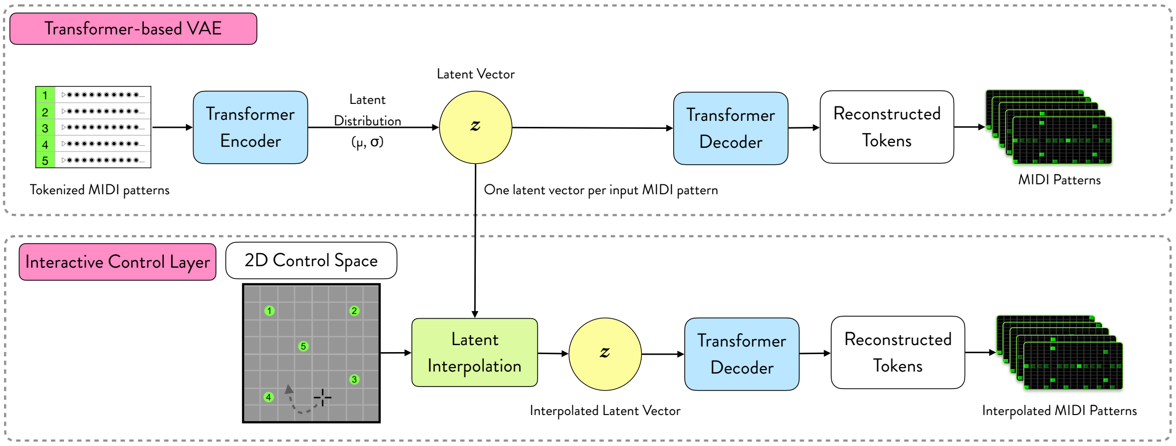

Figure 1: Overview of the Pattern Atlas architecture. The encoder maps input rhythm patterns to the parameters (μ, σ) of a latent distribution. During inference, the latent vector z is deterministically set to μ. In the illustrated example, five patterns are encoded and serve as anchor points for latent space interpolation. A two-dimensional control space provides a user-facing representation of the high-dimensional latent space, enabling intuitive navigation and control. Interpolated latent vectors are subsequently fed into the decoder to generate new rhythmic patterns.

to improve sequence understanding, generation quality, and latent-space organization, while avoiding the practical limitations of fully transformer-based real-time generation. In this sense, *Pattern Atlas* is closer to hybrid symbolic-music systems such as MuseMorphose than to purely autoregressive transformer generation, although its application domain is interactive drum-pattern generation rather than full-song style transfer (Wu and Yang, 2021).

The input representation is based on tokenized MIDI patterns derived from the Groove MIDI Dataset and drum tracks of the LAKH MIDI Dataset. Each pattern covers two bars, organized as 32 sixteenth-note steps. Tokenization follows an event-based scheme with special tokens for padding, start and end of sequence, and step boundaries. Musical attributes are represented through separate token ranges for pitch, velocity, and microtiming. In the current setup, pitch tokens cover MIDI notes 36–59, velocity is quantized into 32 bins, and microtiming is likewise represented in 32 discrete bins, corresponding to deviations in milliseconds relative to the quantized grid. As a result, interpolation affects not only onset structure and instrumentation, but also aspects of groove feel and rhythmic articulation.

The current dataset consists of approximately 25,000 tokenized sequences extracted from the Groove MIDI Dataset and the LAKH MIDI Dataset. At present, only drum and percussion material is used, with the original timing retained and no additional augmentation. Generated patterns are produced inside the device at generation time and then stored for real-time recall during interaction. This supports the broader aim of *Pattern Atlas*: combining richer rhythmic modeling with a representation that remains practical for live use.

2.3 Interactive Latent Space Control

The central interaction principle of Pattern Atlas is a two-dimensional control space in which users explore a musical space defined by their own rhythmic material. Rather than navigating the learned latent space directly, users configure a personal control space by loading MIDI clips as anchor patterns and placing them manually in an XY interface. All anchor patterns are encoded through the same encoder used for the training data. In this way, user-provided MIDI clips are mapped into the learned latent space and can serve as points of departure for generating new rhythms. The number of anchors is not fixed, though in practice the current setup is typically used with four to eight patterns. Anchor placement is fully manual and unconstrained, allowing users to organize the control space according to personal musical logic rather than according to predefined model geometry. This makes personalization central to the system: the interface is not only populated by generated content, but structured by artist-owned rhythmic material.

Interpolation between anchors is based on radial basis function weighting. In practice, this allows the system to generate musically meaningful transitions across the control surface while remaining flexible with respect to the number and placement of anchors. Compared with stricter interpolation schemes, this approach supports smoother transitions and more continuous blending of groove identities. The musical goal is not only to move between similar patterns, but to give performers control over contrast and transformation while retaining a coherent sense of rhythmic relation.

To support live use, interpolation is not computed entirely on the fly during performance. Instead, the system includes a precomputation step, triggered by an interpolation command, that generates rhythm patterns across the grid in advance. These patterns are then stored and recalled in real time during navigation. This design makes continuous exploration possible within a live-performance workflow while preserving the responsiveness of the device inside Ableton Live.

2.4 Max for Live Implementation

Pattern Atlas is implemented as a Max for Live device for Ableton Live (cf. Figure 2), with the aim of making interactive pattern generation available inside a familiar musical environment and without requiring coding. This is an important part of the system's design: rather than treating the model as an external prototype, the project embeds latent rhythm generation directly into an existing DAW workflow for performance and experimentation.

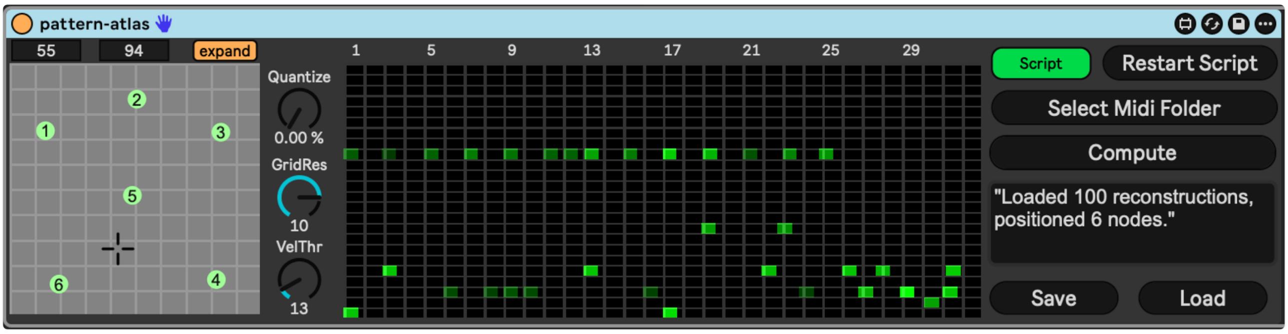

Figure 2: Pattern Atlas as a Max for Live device in Ableton Live. User-provided MIDI clips are placed as anchor patterns in a two-dimensional control space and explored through interpolation.

The interaction model described in section 2.3 is realized through a dedicated graphical user interface within the *Pattern Atlas* device, designed to support both configuration and performance-oriented use.

The central element of the interface is a two-dimensional XY control surface representing the interpolated rhythm space. The surface is displayed as a square field subdivided into a configurable grid, where each cell corresponds to a precomputed rhythmic pattern. A movable cursor, visualized as a crosshair, defines the current position within this space. This position can be controlled via mouse interaction or mapped to external MIDI controllers, enabling continuous navigation during live performance. The rhythm associated with the current cursor position is triggered in real time, allowing the performer to traverse the interpolated field as a continuous musical space.

Anchor patterns are displayed as labeled nodes within the control surface. Each node corresponds to a user-provided MIDI clip, from which a two-bar segment is extracted and encoded into the latent space. Prior to computation, users freely position these anchors within the XY space. Their spatial arrangement directly determines the interpolation behavior: regions between anchors contain rhythmic patterns that reflect weighted combinations of the corresponding latent representations. As a result, movement across the surface produces perceptible morphing between rhythmic identities.

The resolution of the control surface is defined by a grid parameter, which determines the number of precomputed patterns (e.g., $N \times N$). Since all patterns are generated during the precomputation step, changes to this parameter only take effect after precomputation. This reinforces the distinction between a setup phase, in which the structure of the space is defined, and a performance phase, in which it is explored.

In addition to spatial control, the interface provides parameters that influence the generated rhythms at playback time. A quantization control adjusts the strength with which microtiming deviations are aligned to a fixed temporal grid. This operation is applied as a post-processing step, enabling a continuous transition between expressive timing and strictly quantized rhythms. A velocity threshold parameter filters note events below a specified velocity, affecting both the visual representation and the MIDI output and thereby allowing direct control over rhythmic density.

To support perceptual feedback, the currently active rhythm is displayed in a piano-roll-style visualization. This representation shows a two-bar loop with sixteenth-note resolution and a fixed drum-oriented pitch range. Color intensity reflects velocity, and a moving playhead indicates the current playback position. The visualization is read-only and serves to make the temporal structure and dynamic variation of the generated patterns more transparent during interaction.

The right-hand section of the interface provides controls for data loading and computation. Users can load collections of MIDI files, which are then used as anchor patterns. After anchors have been positioned and the grid resolution has been set, a *Compute* command triggers the precomputation of all patterns across the control surface. Progress feedback is provided during this process, and computation can be interrupted if necessary. Once completed, the generated pattern field can be stored and reloaded in subsequent sessions.

The model is trained outside the device and deployed for inference inside Ableton Live. In the current implementation, inference runs locally through a Node/JavaScript layer within Max, while interface logic, MIDI handling, and playback control are managed in Max for Live.

This implementation strategy reflects a practical compromise between model complexity and live usability. Rather than generating every pattern from scratch during continuous movement, *Pattern Atlas* precomputes the interpolated grid in order to maintain responsiveness in performance. At the same time, the model itself runs inside the device at generation time, and the resulting patterns remain part of a unified workflow within Ableton Live. The current system is therefore best understood not only as a deployment pipeline from machine learning model to plugin environment, but as a musician-facing interface design that integrates generative rhythm control into standard DAW practice. A longer demonstration video in the style of a tutorial further documents the device workflow, including anchor loading, interpolation, and continuous navigation within the generated pattern space.[1]

At present, the device focuses on symbolic drum-pattern generation and real-time navigation of interpolated rhythmic spaces. Export functions such as writing generated patterns to MIDI clips have not yet been implemented, and the interface remains a prototype that will require further refinement for extended artistic use. Nevertheless, the current implementation already demonstrates that richer latent rhythm generation can be made accessible in a practical performance environment, and it also points toward future

[1] https://youtu.be/4V67RHrFAYE

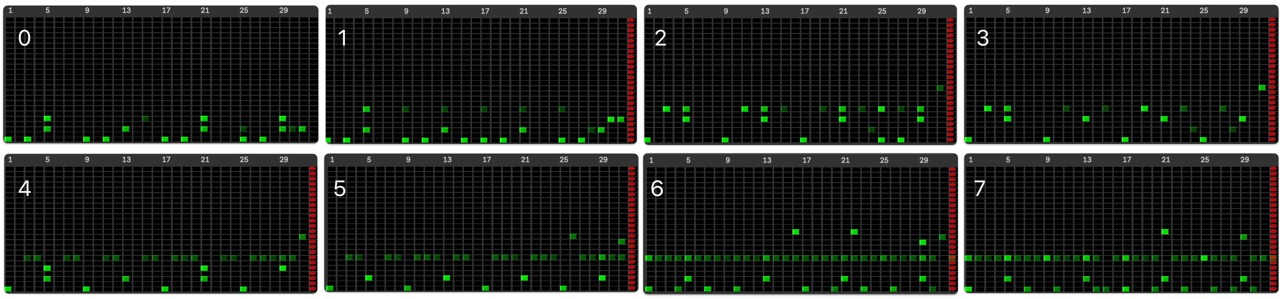

Figure 3: Eight-step interpolation between two anchor patterns in Pattern Atlas. The transformation shows how rhythmic identity changes gradually across the control space, including musically meaningful shifts in instrumentation and the appearance or disappearance of stylistically characteristic elements.

extensions, including melodic or more broadly multimodal pattern spaces.

The implementation of Pattern Atlas, including the Max for Live device and accompanying source code, is publicly available in an online repository.[2]

3 ARTISTIC WORKFLOW, EVALUATION, AND DISCUSSION

Pattern Atlas was developed primarily for artistic practice, especially for live electronic and improvisational performance contexts in which rhythmic material must be generated, transformed, and explored in musically meaningful ways. The present evaluation is therefore preliminary and combines objective reconstruction and interpolation measures with qualitative observations from artistic practice. Rather than reporting a formal user study, this section reflects on the system through its use in our own artistic workflow, supported by quantitative model evaluation, demo examples, listening-based judgment, and observation of system behavior. More broadly, the present work also relates to performance-oriented AI music systems that frame machine learning models as collaborators or co-performers in live contexts (Lorway et al., 2021).

The demo scenario used for this paper follows exactly this process. A small set of personal MIDI clips is loaded, spatially arranged, and explored through continuous movement across the interpolated field. Figure 3 documents an eight-step transition between two anchor patterns and makes the gradual transformation of rhythmic identity visible across the control space. An accompanying audiovisual example further illustrates this process online.[3] In this example, it becomes particularly clear how instrumentation changes in musically meaningful ways over the course of the interpolation, and how individual elements appear or disappear in a stylistically characteristic manner.

From both an artistic and technical perspective, three aspects are especially relevant. First, the model reconstructs input patterns with high fidelity, achieving a windowed reconstruction accuracy of 86.3%, with category-specific accuracies of 82.2% for pitch, 86.6% for velocity, and 84.3% for microtiming. These measures employ tolerance windows for small temporal and expressive deviations and therefore reflect perceptually meaningful reconstruction quality rather than exact token identity. Matches were determined using a windowed nearest-neighbor procedure

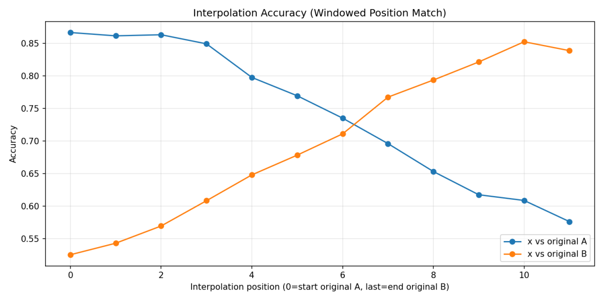

Figure 4: Windowed similarity of interpolated patterns to the two anchor rhythms across an interpolation trajectory. Similarity to anchor A decreases as similarity to anchor B increases. The minimum similarity remains above approximately 0.55, indicating substantial structural commonality between the evaluated rhythms.

with category-specific tolerances for pitch, velocity, and microtiming, allowing small temporal and expressive deviations while preventing multiple assignments of the same predicted token.

Second, interpolation between anchor patterns produces smooth and structurally coherent transitions. Across 15 interpolation pairs, generated patterns achieved a mean windowed similarity of 83.0% to the closer anchor while preserving pitch (78.2%), velocity (82.7%), and microtiming (81.9%) characteristics. As expected, similarity decreases toward the midpoint of the interpolation trajectory because these patterns are furthest from both endpoint rhythms rather than indicating a degradation of rhythmic coherence. Figure 4 illustrates this behavior for an interpolation trajectory: similarity to the first anchor gradually decreases while similarity to the second anchor increases, with both curves approaching one another near the center.

A limitation visible in Figure 4 is the comparatively high minimum similarity. Even patterns located furthest from one of the anchors retain a windowed similarity of approximately 0.55 to that anchor. This suggests that the interpolation does not move between entirely dissimilar rhythmic structures. One possible explanation is that all patterns used in this evaluation originate from the Groove MIDI Dataset and therefore share recurring characteristics in instrumentation, rhythmic vocabulary, and performance style. While this common basis may support coherent transitions, it may also constrain the stylistic distance that can be achieved within the current latent space.

[2] https://github.com/hfmnuernberg/PatternAtlas
[3] https://youtu.be/DKxJSdMcXjI?si=o0F5jZxnRk8Xvi10

Third, interpolation affects not only onset structure but also expressive performance characteristics through velocity and microtiming. Together, these quantitative observations support the listening-based impression that Pattern Atlas enables controllable yet creatively varied exploration of rhythmic spaces.

This relates closely to Yee-King's description of latent spaces as creative spaces rather than merely technical embeddings (Yee-King, 2022). It also resonates with recent work that seeks to make latent-space behavior more inspectable, understandable, and artistically usable through interface design and visual feedback (Bryan-Kinns et al., 2023). In *Pattern Atlas*, the latent representation becomes musically accessible through the combination of personal anchor patterns, spatial arrangement, and continuous navigation. The result is a more performable and musically legible form of latent-space interaction, in which performers shape and traverse a rhythmic field configured through their own material. This performative understanding of latent-space interaction also connects to recent live systems in which performers manipulate latent representations in real time as part of an improvisational setting (Shaheed and Wang, 2024).

4 CONCLUSION AND FUTURE WORK

This paper presented *Pattern Atlas* as an interactive pattern generation system and Max for Live device for generative AI performance. The system combines a pre-trained rhythm model with user-owned MIDI material, enabling performers to generate and explore new rhythmic patterns from personally selected anchors within a live DAW environment. In this way, *Pattern Atlas* addresses a central challenge of latent rhythm systems by aiming for richer rhythmic output, greater flexibility of control, and a more direct connection between learned models and artistic practice.

Pattern Atlas also reflects a broader tension between small-data personalization and larger pre-trained models. Earlier small-data approaches emphasized artist-specific training material and data ownership (Vigliensoni et al., 2020, 2022). *Pattern Atlas* continues this concern, but combines it with a broader learned latent space. Personal MIDI material remains central, not through retraining, but through anchor definition and interpolation. This allows the system to retain artistic specificity while benefiting from the wider coverage of a pre-trained model.

At a technical level, the paper described a transformer-based VAE architecture for symbolic drum-pattern generation, a velocity-sensitive representation with explicit microtiming offsets, and a control model in which user-defined anchor patterns are explored through interpolation in a two-dimensional performance space. At the level of artistic use, the system showed how latent rhythm generation can become more performable and musically legible when personal material, continuous navigation, and DAW-based interaction are brought together in a single interface.

The present implementation remains a prototype, and future work will focus in particular on refining the user interface for extended artistic use. Beyond this, several musical and technical extensions are planned. One important direction is the expansion from drum-pattern generation toward melody and more broadly multimodal musical spaces. Another is the use of a wider pitch range and more flexible pattern representations. Future work will also continue to explore how personalized musical material can be combined with broader pre-trained models, especially in relation to artistic specificity and control.

In a broader artistic context, *Pattern Atlas* may also support forms of use beyond live performance, including exploratory compositional settings and installation-based applications. In such contexts, manually arranged pattern spaces may offer further possibilities for structuring musical material through spatial placement and interpolation. Overall, *Pattern Atlas* suggests that latent rhythm generation can become a productive artistic interface when broader learned rhythmic knowledge is combined with personal material, performable control, and a familiar musical environment.

5 ETHICS STATEMENT

The development of Pattern Atlas involves the use of symbolic MIDI data from the Groove MIDI Dataset and drum tracks from the Lakh MIDI Dataset. Both datasets consist of publicly available material widely used in music information retrieval research. The Groove MIDI Dataset contains professionally recorded drum performances released by Google Magenta for research purposes. No additional personal or sensitive user data was collected during training, and therefore no formal ethics approval was required for the use of these datasets in this work.

In addition to the public datasets, the system supports the use of user-provided MIDI material as input. These user-generated patterns remain locally stored and are only processed within the application for the purpose of encoding and generating new rhythmic material. No external transmission, logging, or secondary use of user data is performed.

The system is designed as a creative tool for musical exploration rather than a generative system intended for autonomous content production. Consequently, it does not aim to replicate or imitate identifiable performances beyond stylistic rhythmic characteristics inherent in the training data.

We acknowledge that generative models for music raise broader questions regarding authorship, stylistic influence, and creative agency. Pattern Atlas positions itself within this discourse as an assistive instrument that foregrounds user control through interaction design and personal data integration.

REFERENCES

Boden, M. A. (2004). *The Creative Mind: Myths and Mechanisms*. Routledge, 2 edition.

Bryan-Kinns, N., Banar, B., Ford, C., Reed, C. N., Zhang, Y., Colton, S., and Armitage, J. (2023). Exploring XAI for the arts: Explaining latent space in generative music. *arXiv preprint arXiv:2308.05496*.

Eck, D. and Schmidhuber, J. (2002). Finding temporal structure in music: Blues improvisation with LSTM recurrent networks. In *Proceedings of the 2002 IEEE Workshop on Neural Networks for Signal Processing*, pages 747–756. IEEE.

Google Creative Lab and Magenta (2018). Beat blender. https://experiments.withgoogle.com/ai/beat-blender. Interactive web experiment.

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A. M., Hoffman, M. D., Dinculescu, M., and Eck, D. (2018). Music transformer: Generating music with long-term structure. *arXiv preprint arXiv:1809.04281*.

Lorway, N., Powley, E., and Wilson, A. (2021). Autopia: An AI collaborator for live networked computer music performance. In *Proceedings of the 2nd AI Music Creativity Conference (AIMC)*.

Lunt, A. and Trump, S. (2023). Latent space explorer: Interactive rhythm generation through musical actions. In *Proceedings of the 4th Conference on AI Music Creativity*.

Pati, A., Lerch, A., and Hadjeres, G. (2019). Learning to traverse latent spaces for musical score inpainting. In *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*.

Roberts, A., Engel, J., Raffel, C., Hawthorne, C., and Eck, D. (2018). A hierarchical latent vector model for learning long-term structure in music. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pages 4364–4373.

Shaheed, N. and Wang, G. (2024). I am sitting in a (latent) room. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pages 333–338.

Tokui, N. (2020). Towards democratizing music production with AI: Design of variational autoencoder-based rhythm generator as a DAW plugin. *arXiv preprint arXiv:2004.01525*.

Trump, S. and Lunt, A. (2025). Sonic Map Explorer – Ein emergentes Interface zur Live-Interaktion mit KI-Musiksystemen. In *Mensch und Computer 2025 – Workshopband*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*.

Vigliensoni, G., McCallum, L., and Fiebrink, R. (2020). Creating latent spaces for modern music genre rhythms using minimal training data. In *Proceedings of the International Conference on Computational Creativity (ICCC)*.

Vigliensoni, G., McCallum, L., and Fiebrink, R. (2022). R-VAE: Live latent space drum rhythm generation from small training data. *Journal of Creative Music Systems*, 6(1).

Wu, S.-L. and Yang, Y.-H. (2021). Musemorphose: Full-song and fine-grained piano music style transfer with one transformer VAE. *arXiv preprint arXiv:2105.04090*.

Yee-King, M. (2022). Latent spaces: A creative approach. In Vear, C. and Poltronieri, F., editors, *The Language of Creative AI: Practices, Aesthetics and Structures*, pages 137–154. Springer International Publishing, Cham.