MAICO: A Visualization Design Study on AI-Assisted Music Composition

Simeon Rau, Frank Heyen, Benedikt Brachtel, and Michael Sedlmair *Member, IEEE*

***Abstract*—We contribute a design study on using visual analytics for AI-assisted music composition. The main result is the interface MAICO (Music AI Co-creativity), which allows composers and other music creators to interactively generate, explore, select, edit, and compare samples from generative music models. MAICO is based on the idea of visual parameter space analysis and supports the simultaneous analysis of hundreds of short samples of symbolic music from multiple models, displaying them in different metric- and similarity-based layouts. We developed and evaluated MAICO together with a professional composer who actively used it for five months to create, among other things, a composition for the Biennale Arte 2024 in Venice, which was recorded by the Munich Symphonic Orchestra. We discuss our design choices and lessons learned from this endeavor to support Human-AI co-creativity with visual analytics.**

***Index Terms*—Music composition, human-AI, comparative visualization, glyphs, similarity, design study.**

I. Introduction

PROFESSIONAL and hobbyist artists alike are increasingly integrating generative algorithms into their creative work [11], [34], [55]. Generative music models, for instance, can take a primer melody together with parameters or a prompt and then create variations of this melody, propose continuations of it, suggest harmonic/polyphonic extensions, or even construct entire songs [26], [32], [35], [49], [57]. For most composers, however, a fully automated generation of entire songs is not the goal, as it strongly contrasts the human creative process of composition, lacks personality, and often misses the intention of the composer [23], [81]. Instead, generative models are used as tools [18], [51] that can provide music snippets, foster inspiration, assist the composer with tedious tasks at different stages of the workflow [56], or provide externalized music knowledge for beginning composers [62].

We argue that generative artificial intelligence (AI) should not replace human creativity but instead could serve as a complementary tool in a composer's toolbox, fostering inspiration and providing alternative perspectives. Many musicians are looking for new sounds and techniques, which has continuously contributed to the increasing variety of music in the past. Consider, for instance, the digital revolution in music through sampling and digital audio workstations (DAWs) [21], which offered composers a plethora of new possibilities. AI promises to be the next big step in this development. Although there are several risks regarding the power of AI to generate whole songs and the resulting worry of a human replacement, use of copyrighted art, and biases (for example, toward Western music) [21], composers are eager to incorporate these new tools into their composition workflows. Additionally, current law does not allow the copyright of AI-generated pieces if there is no substantial human contribution, with little to no human-in-the-loop. Instead, our goal is to always leave the composer in control and only assist in steps in the composition pipeline through adequate user interfaces with the AI [43].

Other user interfaces already allow composers to leverage generative models by repeatedly generating single snippets, such as the one by Bazin et al. [4]. However, unsatisfying results lead them to repeatedly re-generate further snippets [31], and properly steering models and integrating them into artistic and creative work remains an open challenge. To this end, it would be beneficial to look at many samples from different models and parameter settings simultaneously and more systematically. Otherwise, composers might not know which AI model to use, which parameters to change, and whether the next generations will produce better or worse samples.

Generating larger numbers of samples with AI can help answer these questions, but analyzing many samples manually and sequentially is a tedious and time-consuming task. To better support this analysis, we combine the concepts of visual parameter space analysis [75] and notational audiation [8] (visually imagining music). Visual parameter space analysis uses visual overviews and interactive exploration for analyzing models and model ensembles (fig. 2). Similar approaches have shown to be useful in many application areas that need to systematically explore and understand models and simulations, for instance, when judging different scenarios of flood simulations [90] or parameterizing classification models [28]. Little work [62], however, exists on how this approach can be leveraged for exploring the output of generative music models.

While visualization cannot fully replace listening, it has always played an important role in composition: in the form of sheet music notations. With training, it is possible to understand (i.e., "hear") music simply by visually reading it [8], an indispensable approach in professional composition [63]. We aim to use this listening/reading dualism and investigate how more sophisticated visual analytics approaches can support composers working with generative music AI.

Toward better understanding this area and supporting composers in their workflow, we contribute the design and evaluation of MAICO (Music AI Co-creativity), which allows creators to generate a collection of hundreds of samples from different generative music models at once. Previous work [62] has already helped beginners with visualization to process a small number of samples simultaneously. We extend this work with a focus on professionals and their workflow with larger sample sizes from multiple different models. Our goal is to assist composers in efficiently generating and systematically

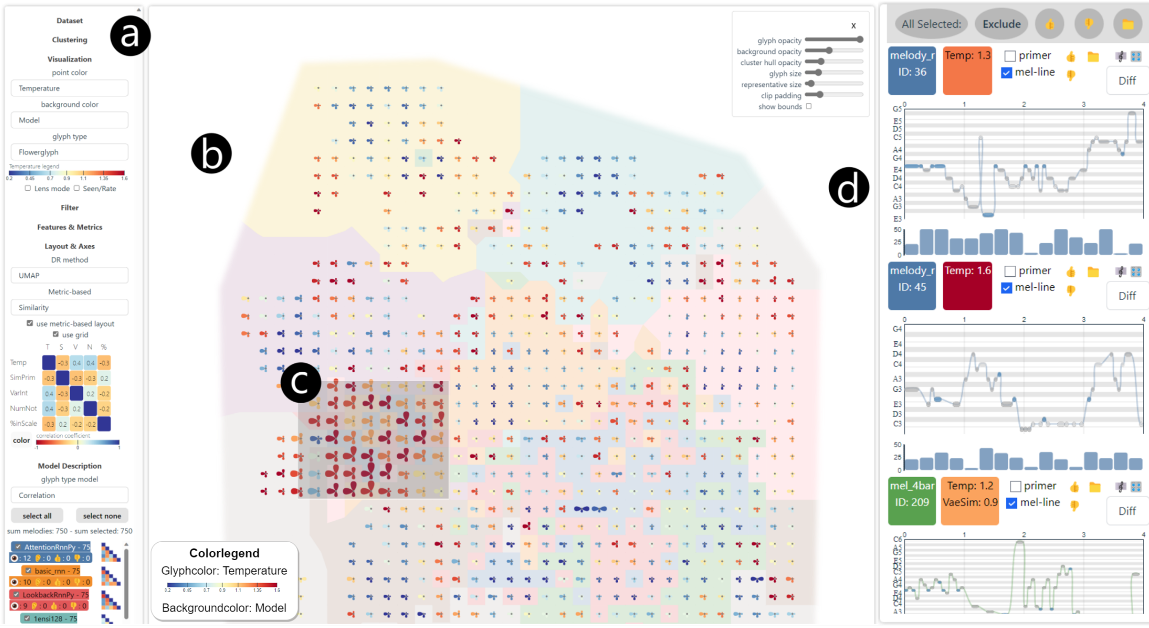


Fig. 1: MAICO's main visualizations to explore generative music models: **a)** Menu for data generation & model selection, clustering & filtering, and layout & encoding options. **b)** Overview visualization with glyphs placed by similarity; glyphs and their background color-code parameters and metrics. **c)** Brushed selection of samples, for which more details are shown in **d)** a detail panel with piano rolls and meta information.

exploring this large sample collection visually to find samples that fit a composition. Furthermore, we want to support the comparison and analysis of models to help composers learn about how and when to use which model.

We specifically focus on models that generate symbolic music and output discrete notes (like notation) instead of audio – which allows composers to first focus on sample selection and adjustment and then assign instruments separately. During MAICO's design process, we closely followed design study methodology [78] and worked together with the professional composer Benedikt Brachtel, who we included as a co-author due to his substantial contributions. MAICO provides visual overviews to get an approximate image of the space of possible outputs along with their parameters and interactions to narrow the space down to essential samples. For generated samples, we explore different glyphs [5] that accelerate readability in the overview. Using different musical metrics, our visualizations show clusters of music samples using similarity or correlation and provide aggregations and summaries.

We first evaluated our initial design with usage scenarios, followed by the design study and interviews with music experts from different backgrounds. Brachtel generally preferred our approach over the manual sampling baseline, and MAICO assisted him in creating two commissioned works. These results demonstrate MAICO's potential for processing large numbers of samples, interacting with them, and thereby assisting composers in elaborating on motifs more efficiently.

Our supplemental material contains more extensive versions of some of the following sections, a video showcasing functionality and user interaction, examples from sessions and one of the commissioned works, as well as the source code and a web-based demo of our prototype[2].

In summary, we contribute 1) a design study on AI-assisted composition with a professional composer, resulting in interactive visualizations that help investigate behaviors of and differences between music-generating models, the influence of parameters, and each model's samples for music creation, 2) an evaluation through usage scenarios, a multi-dimensional in-depth long-term case study (MILC [80]), and expert interviews, as well as 3) a web-based prototype (MAICO)[2].

II. Related Work

We review related work on interfaces for algorithmic music composition, as well as the more general areas of visualization of musical data and visualization of machine learning models that underlie the ideas of our work.

A. Algorithmic Music Composition

While algorithmic composition has been used since the 1980s [27], advances in deep learning brought a variety of new methods [7] for audio [49] and symbolic music generation [35], including recurrent neural networks [26], [57], variational autoencoders [59], [66], and generative adversarial networks [19]. Other work contributes combinations of the surveyed methods [17], [35], transformers [32], [45], agent-based and heuristic algorithms [41]. Recent research explores large language models for music generation [46], affective music generation through Markov models and rule-based systems [16], and better controllability [61], [91].

[2]https://github.com/visvar/MAICoV2/

Interfaces for Algorithmic Composition: Controllable models together with respective user interfaces can serve to augment artists' workflows and creativity [81] while the artist retains authorship and freedom of expression [31]. Although interfaces that provide more control perform better, offer more authorship, and are therefore preferred by users [42], [43], research in this area is sparse compared to AI development [12]. While others used visual interfaces for audio synthesis with, e.g., corpus-based methods for generation [74], [86], they mostly followed) an audio-focused end-to-end approach. In contrast, we focus on a symbolic approach, which has a clear separation between note information and sound, allowing for a more faceted control in the composition process. Applying human-AI interaction design guidelines for co-creative systems [18], [65], some human-in-the-loop interfaces allow creating a piece step-by-step [4], [25], [45], [60], [62], [94]. In our own previous work [62], we proposed an interface for AI-supported composition for novice users. The main idea was to use visualization to guide the choice between a few generated alternatives and edit them more efficiently. Building on these initial ideas, MAICO takes them to the next level by targeting professional composers as the intended user group. To do so, it allows for a systematic analysis of a much larger sample space, giving more control during generation, exploring multiple models and parameterizations in more detail, and comparing models to improve understanding and control.

Embedding into DAWs: Algorithmic composition tools have also found their way into widely-used digital audio workstations. The DAW Ableton Live, for instance, allows integrating Magenta Studio [67] to use models like MusicRNN/MelodyRNN and MusicVAE [66] for generating a few samples directly as clips with limited steering possibilities. In comparison, we incorporate more user control with filters and allow for simultaneous assessment of many more samples through visualization. In fact, we use the same models and, therefore, the same generative quality but further allow integrating similar models as plugins. While integrating MAICO into Ableton Live is out of our scope, future work could turn it into a plugin to make it available to a broader audience.

User Evaluation: Algorithmic composition approaches have been evaluated in several musical settings before. In the context of an AI song contest [31], for instance, Micchi et al. [51] incorporated AI into songwriting by generating single samples repeatedly until one worth working with appeared. Deruty et al. [18] worked together with musicians and tested various AI tools (such as NONOTO [4]) in contemporary popular music production, showing that adequate interfaces/tools are needed in order to make these tools interesting for artists. To this line of work, we contribute a longitudinal in-depth evaluation with a professional composer who used our tool for commissioned compositions.

B. Visualization of Musical Data

Visualization has been used for different types of musical data before. Typical visualization techniques like charts, maps, and piano roll views have been used for different data types like musical works, collections, musicians, or instrument data.

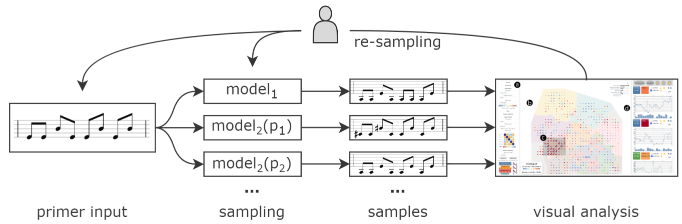


Fig. 2: In visual parameter space analysis [75], the models' output gets sampled for different parameterizations (p_i) and then visualized. Iterative re-sampling allows the analyst to further explore interesting areas of the parameter space. This analysis provides insights into the model's behavior and enables more effective usage.

As a full coverage of this area is beyond the scope of this paper, we point the interested reader to two comprehensive surveys by Khulusi et al. [33] and Lima et al. [40]. Khulusi et al. [33] show the recurring combinations of maps for musical collections and piano rolls for musical works, two relevant techniques and representations for our work. Lima et al. [40] focus more on the specific techniques of music representation and feature encodings, including glyphs, which also play a major role in our design. By classifying approaches according to goals and users, they also identified a lack of papers that target music composers, which is the focus of our work.

Focusing on analysis tasks in musicology, visual analytics has been used to investigate music corpora [53] and semantic structures of sequences [10], or to augment music notation [52]. Fourney and Fels [22] explored how visualization can help deaf consumers get a feeling for the music without listening to it, a topic related to our ideas of leveraging notational audiation. Wu et al. [92] contributed a visual interface to control modes during live performance, comparable to our interface for controlling AI models. However, little research looked at visualization and analytics for generative AI in a musical context [62], which is the focus of our work. We use common approaches like maps for a sample collection [33], enhanced with metrics and glyphs [40], supporting the exploration of an AI-generated sample space in the context of music composition, rather than for purely analytical tasks.

C. Visualization of Machine Learning Models

One goal of our work is to help composers better understand the models they use, a task that is shared by other work on machine learning visualization. A recent survey [89] on visual analytics for machine learning focused on data types and tasks and showed a gap for generative models, even outside of generative AI for music. Many visual analytics tools examine other AI models at different stages of the machine learning pipeline [68], most of which focus on the development of new techniques. In contrast, comparing and selecting models, which is highly impacted by user's needs [9], is under-researched [29]. A few works, such as by Heyen et al. [28] and others [2], [64], analyze the performance and accuracy of classifiers in order to help choose the best classifier. Abstractly, our approach follows a similar intention, but we focus on a different domain with different tasks – music composition. Our goal is to support the individual and subjective analysis of music-generating models, as a fully objective and automatic evaluation is usually insufficient for supporting human artistic creativity [24], [27], [93].

III. Design

In the following, we discuss our design process, user and data characteristics, and the resulting design with its visualizations and metrics.

A. Process

Following design study methodology [78], we closely cooperated with a professional composer, Benedikt Brachtel[1], as a domain expert. He studied classical composition and works as a freelance composer and music producer since 2010. He regularly encounters contemporary means of compositional creation and developed a tool (dodecaphony enforcer/avoider) in 2019 that turns music automatically into a 12-tone series based on Schoenberg's theory [71]. His clients included the Bavarian State Opera and the Burgtheater Vienna.

We started with an initial prototype focusing on model comparison, which we evaluated in semi-structured interviews with four musicians with different levels of musical expertise. Despite an initial learning curve – which was expected as MAICO is an expert tool – all participants offered highly positive feedback on the visual approach. Our design helped them to get an overview of large spaces with many different samples. They found that visualization is very important to find specific samples and makes them like to explore multiple options and analyze what fits best. Using model glyphs (density piano roll), one participant found that *"many samples are the same, and there is a bit of variation"*, showing the usefulness of these glyphs to filter models (more details on this initial study can be found in the supplemental materials).

After we had implemented their feedback, we showed the prototype to our expert in an initial meeting using a pair analytics approach [3] to analyze the requirements for MAICO. He was immediately interested in using it and curious about what he could do with it and how it could help him with his work, leading us to start our in-depth collaboration.

Over the course of five months, we met weekly to iteratively extend and adapt MAICO's design, resulting in a total of 15 video-recorded sessions. During these five months, Brachtel used our prototype for his own work, and we tracked his interactions and generated samples. In our meetings, he gave us feedback on the existing features and the needs that occurred during the prior week. Together, we developed solutions to address these needs, implemented them, and analyzed their impact during the next meeting or after he used them. With this iterative design and regular updates, we were able to incorporate his ideas and demands immediately and verify MAICO's applicability.

[1]https://www.benibrachtel.com

B. Users & Tasks

Our primary target audience is professional composers and songwriters who work with symbolic music and desire inspiration from generated content or want to evaluate or integrate it. A composer's typical workflow consists of these steps:

- S1) identify themes and requirements of the work,
- S2) find musical motifs,
- S3) elaborate on motifs and bring them into context,
- S4) find suitable instrumentation for the arrangement,
- S5) write down the score, and eventually
- S6) record or perform the work.

We intend for composers to use MAICO at the beginning of the workflow (S2, S3) as inspiration for finding motifs and/or elaborating on these motifs, which is the most challenging composition step. For the other steps, good tools already exist. As the composer's goal is to generate and find suitable samples (S2, S3) for their personal composition efficiently, we identified the following high-level tasks:

- T1) steer models (using T4, T5) toward ideas, generate samples,
- T2) systematically explore the space of samples and thin it out,
- T3) find musical variations and characteristics in sets of samples,
- T4) analyze the influence of model parameters on the generated output and find appropriate values, and
- T5) compare multiple models and select the best-fitting one(s) for a specific use case.

C. MAICO Overview

To support these tasks, we leverage a visual parameter space analysis approach as illustrated in figure 2. After selecting a primer input, we generate the samples using different models and parameters. These samples are then prepared and displayed visually for further analysis. By gaining insights throughout this analysis, the composer can narrow down the search and exploration space and make increasingly more effective choices for inputs, models, and parameters before re-sampling and repeating the analysis on the new output.

In our case, the primer input and generated data are musical samples, where each data point is one symbolic snippet. At the heart of our approach is a visual overview, which shows a similarity map of the samples (fig. 1b), akin to many other visual parameter space analysis approaches [28], [44], [69]. This map offers a large space to organize samples, form clusters, and show outliers [6].

Because sequential analysis is not possible – listening to all samples one after another takes too long – we need metrics to organize and visualization to allow composers to explore the sample space. To cater to the creative, subjective, and ill-defined nature of composing, we need a broad range of different metrics, like quantitative metrics that describe the musical aspect, as well as a similarity metric to define the similarity between two samples.

Along these main components, an interactive environment is needed to dynamically support the composer's workflow and tasks. In the following, we describe these different design components in detail.

D. Data & Sampling

Our primary data are the samples generated by the selected models, each representing a short motif/snippet of symbolic music, i.e., a set of discrete notes temporally ordered by a rhythm. A sample can either be monophonic, that is, a single melody/voice with no notes being played simultaneously; or polyphonic (in our case counterpoints), with two or more simultaneously played melodies/voices, adding potential harmonies to the composition.

Monophonic Samples: To focus on motifs while keeping a manageable scope, we started with models that generate monophonic samples – only one note at each time. The models we chose have one parameter called *temperature* that – put simply – controls the randomness of the generated samples. We use pre-trained models (Google's MusicRNN/MelodyRNN and MusicVAE [66]). Our selection of models is exchangeable, as we integrate them through a plugin system that allows us to use any model with similar parameters and outputs hosted on the internet. In doing so, we intend to be flexible toward the needs of different composers. In our case, using some notes to prime a model (primer input), we sample up to 10 models *simultaneously* and up to 15 different temperature values to represent each model's output diversity. Primer, models, and temperature values are all chosen by the composer.

Polyphonic Samples: Because polyphony is interesting to many composers, we generate novel, polyphonic music samples by systematically combining generated monophonic samples. In doing so, we support a more experimental approach to composition and sidestep the rather repetitive replications of existing content that occur in polyphonies directly generated by AI [12]. Using a systematic and exchangeable rule, we create polyphonic two-, three-, or four-voice samples by combining monophonic samples that have no overlap of notes at any time. Therefore, one monophonic sample symbolizes one "voice" of the polyphonic sample, which is a counterpoint, as these are distinct melodies rather than harmonics. This approach contrasts classical compositions because it allows the generation of polyphonic samples that are rather distant from the music one typically composes, but still uses some variation of the primer melody.

E. Metrics

In order to judge and compare the generated samples, composers need to know about their characteristics. For example, composers might have certain characteristics in mind when looking for samples, varying from lower-level statistics like "number of notes" to higher-level concepts like "brightness". We, therefore, include metrics that allow us to correlate, compare, and visualize samples and visually place them in space. These metrics, for instance, allow us to represent a monophonic or polyphonic sample compactly in our overview visualization (T3) (fig. 5b-d). After considering a large set of potential metrics, which we evaluated, we implemented the

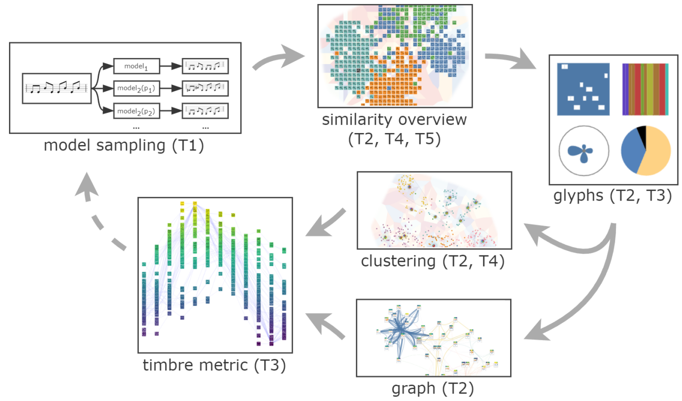


Fig. 3: The workflow that Brachtel uses to ultimately find interesting samples from hundreds of generated ones. He starts off by generating samples using different models and parameters (top left), then gets an overview of the space using a similarity-based layout with glyphs and clustering, refines the selection by looking at musical harmonic information in order to export the right samples, and starts over again with increasingly better primers, models, and parameters.

following metrics to represent different statistical and music-theoretical aspects of the samples. While these metrics are specifically chosen for Brachtel and his ongoing composition projects, other compositions or composers may need different metrics due to different individual requirements.

Statistical Metrics: With the first set of metrics, we intend to quantify different statistical properties that are relevant to musically characterize samples or reveal information about input settings for AI generation. The *number of notes* and the percentage of time that is not occupied by rests (*ratio of non-silent beats*) directly count characteristics of samples, while the *mean note duration* and *mean pitch* (how high a note sounds) directly average the characteristics of the notes. In order to apply suitable instruments later, a composer usually needs to know the *pitch range* of a sample's notes, which is why we calculate the *maximum and minimum pitch* and the range in between. For monophonic samples, the *variance of pitch intervals*, meaning the pitch difference between two consecutive notes, reveals whether a sample is lively with many different intervals or monotone. For polyphonic samples, we calculate the *intervals* between two neighboring voices, showing their relationship. In order to learn more about a model's behavior and improve steering in the future, composers can look at the input setting, that is, the provided *model* and the *temperature* value that was used to generate the sample.

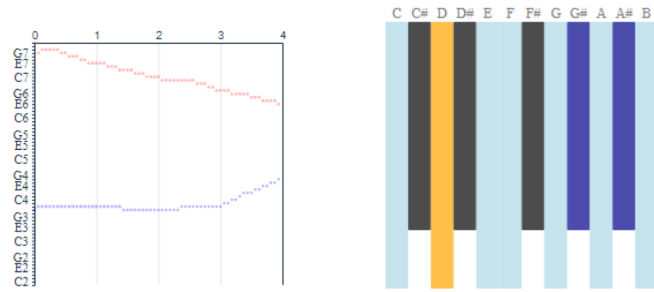
Music Theory Metrics: To allow for more musical reasoning, we include metrics that take music theory into account: For each sample, we determine the *percentage of notes that are offbeat* (not on the counts of quarters), *rhythmic complexity* (whole and per beat, by taking rhythmic changes and offbeat notes into account) [84], [85], as well as the *direction of the monophonic sample* – whether it is rising (more positive intervals than negative) or falling (more negative than positive intervals). For intervals between consecutive notes, we take the *percentage of perfect intervals like fourth, fifth, and* *octave* [83]. In the initial prototype, we used the musical *key* (tonic plus major or minor), the percentage of notes that fit this key (*in-scale notes*), and collected harmonic information, for example, by counting *tonic, dominant, and subdominant* notes [83]. To detect the key, we tried three different algorithms: one from the tonaljs (github.com/tonaljs) library and two by Krumhansl and Schmuckler [36] and Temperley [82]. Through our interviews, we noted that using musical keys is sometimes ambiguous, especially for short monophonic samples, due to the amount of special cases that can occur on randomly generated samples. To address this ambiguity, we allowed composers to manually select a key.

Talking to experts, we found that using the key was not beneficial, as they could tell the key by looking at samples better than our algorithms, often leading to disagreements or multiple possibilities. Therefore, we replaced the key with a *timbre* metric Brachtel proposed, which uses the circle of fifths and overcomes certain biases of Western music theory. Instead of fixing a key, the timbre metric indicates whether a sample implies a bright or dark harmonic. We use a root note, which functions as a musical reference point, and rotate the circle of fifths such that the root appears at the top. For each note in the sample, we determine whether it appears on the right side of the circle, which indicates brightness, or the left for darkness. Because the bottom note can be added to both sides, we add their occurrences toward the leading side and finally derive a value between zero for dark and one for bright samples. Since this metric relies on a root note, which is hard to detect in some scenarios and can, therefore, be misleading, we allow composers to manually select one.

Similarity Metrics: In order to relate samples directly to each other, we also need dedicated similarity metrics. Since other similarity metrics for symbolic music [58] lack controllability and composers are often more interested in either rhythmic *or* harmonic structures, we use a weighted approach for a normalized similarity metric that takes in rhythmic and harmonic information independently and works for both monophonic and polyphonic symbolic samples. This metric allows us to compare two samples in order to display similar intent or use pairwise similarities for visual placement. Together with the interviewed musicians (section III-A), we analyzed the metric's validity, and they all approved the similarity between samples. Music commonly expresses emotion, so we combine statistical (e.g., number of notes) and music-theoretical (e.g., key, but not timbre) metrics to create an emotion-based similarity metric [13], [39], [50], [70], [73].

F. Visualization Design

In this section, we describe the visualizations and our design choices along Brachtel's workflow throughout the project (fig. 3). Our visual interface (fig. 1) consists of three main parts: 1) a menu for data generation via selected models or algorithms, visualization options like layouts and encodings, and model summaries, 2) an overview that represents samples as glyphs positioned by parameters/metrics or similarity, and 3) a detail panel for selected samples.



(a) (b)

Fig. 4: Two filters allow steering model sampling (a) by pitch range per step and (b) by allowed chroma. In this example, (a) we narrow the range toward the end to get a controlled and more concentrated ending. Also, (b) we allow any chroma but C♯, D♯, and F♯ (gray), while we select D as our root note (orange) to look for samples that are highly related to the D minor key using our other visualizations, as Brachtel did in his commissioned work.

Generating Samples: Brachtel's workflow begins with uploading a primer melody and querying all selected models using their parameters and our filters. Alternatively, we allow importing an existing dataset of samples from earlier sessions. As Brachtel usually already has rough ideas in mind, he wants to steer models by limiting the range of pitch as well as the melody direction (rising/falling) of samples and the ability to enforce tonality (T1). Using common visualization designs for music, in the form of piano rolls and piano keys, he can define a pitch range for each timestep in the sample (fig. 4a) by drawing two lines that the notes have to be between. We chose piano rolls over sheet music, as they avoid biasing to stay within a key, allowing for more musical freedom. To enforce any wanted tonality, composers can use the piano keys to select keys that should be allowed in the results, highlighted in blue, and to select a root note, highlighted in orange (fig. 4b). If a generated sample does not pass the filters, we automatically move the conflicting notes up or down by octaves or single steps until they fit. As composers might change their minds later, we allow them to adjust samples using these filters after generation as well. With the above tools, they can "steer" a model's output (T1) even if the model itself is does not provide such features or has limited controllability.

Getting an overview: To get an overview of the samples, Brachtel then explores the space (T2) with a similarity-based layout to visualize all samples as dots simultaneously. We compute this layout based on our similarity metrics (section III-E), the result of which we feed into dimensionality reduction to get two-dimensional positions (fig. 6). Concretely, we allow choosing between MDS [37], [38] and UMAP [48] from DruidJS [14], as these optimize for local and global similarity respectively. As an alternative, we also allow choosing two metrics for the scatterplot axes.

In order to provide a meaningful overview without a biased selection by the composer, the similarity-based layout (fig. 6a) places similar samples close together, and dissimilar ones are

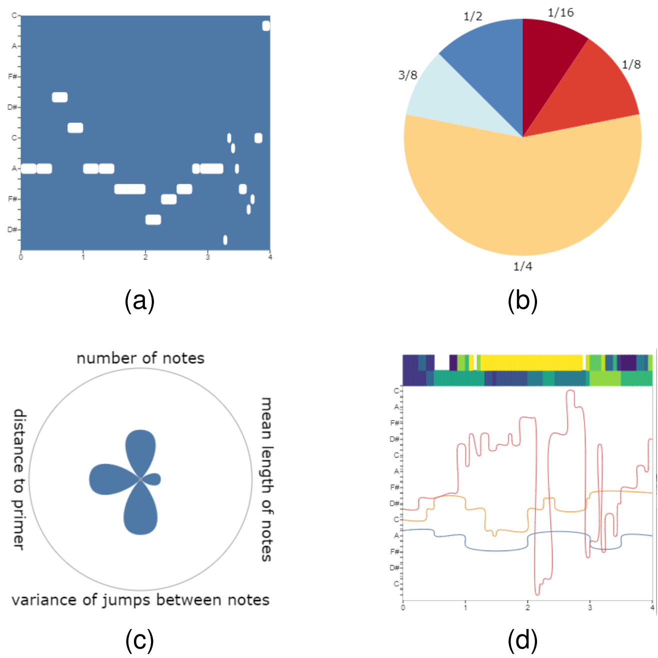


Fig. 5: The different types of glyphs that represent samples: **(a)** piano rolls show the raw data, **(b)** rhythm pie charts show distribution of note durations, **(c)** metric flower glyphs show different pitch metrics, **(d)** voice lines show voices and their movement and additionally display the distances between voices at each step through color at the top.

far away. Brachtel easily finds outlier samples – those totally different from the primer or other generated samples – as they are placed far away from most others.

In some cases, multiple samples share roughly the same position. This makes displaying and investigating all of them difficult, as the samples' representations overlap, reducing the ability to make fast notational audiation and visual overview judgments. We thus apply gridification [15] to produce an occlusion-free layout, with the drawback of losing some accuracy in the layout as positions are shifted and distances might change (fig. 6b).

In order to explore the space systematically (T2), a composer needs more information about the characteristics of a sample to find interesting regions and thin out the space effectively. Therefore, we color the dots using a composer-selected metric (T3, T4, T5). We additionally use Voronoi cells to assign each sample a separate background color that encodes another chosen metric and allows composers to look for patterns regarding two metrics at once. Because overly large cells at the outskirts of the plot visually bias findings, we use a convex hull to cut off background colors.

By displaying the model and temperature information as colors, Brachtel can keep track of models and temperatures that work well for him (T5), gaining knowledge about their behavior (T4). This knowledge helps him to reduce the number of samples by excluding models he dislikes and improves his choice of models and parameters for more efficient generation next time (T1).

Representing Samples with Glyphs: In our initial overview with dots, we encountered the problem that color-coding only two metrics was often not enough to find interesting regions based on musical characteristics. Therefore, we replaced the dots with different types of glyphs [5] that reveal more information about a sample (T2, T3).

To encode a set of metrics compactly, we use *flower glyphs* (fig. 5c), which are more readable than the commonly used star glyphs [87]. We display the rhythmic variety of samples as *pie charts* (fig. 5b) of the distributions of note duration, allowing for normalized comparison of samples with different numbers of notes. To represent the effect of a note's duration, we display short notes in red, as shorter notes often show activity, while long notes are colored blue, indicating calmness.

Talking to experts, we noticed that the above glyphs do indeed help but are not effective enough to display the music itself, making it hard to find regions of interesting samples. Although these metrics can help characterize samples regarding their liveliness and pitch range, imagining the motif through notational audiation is not possible with them. For this use case, we had already implemented piano rolls and later added voice lines. These show the data of the sample itself (T3) and became Brachtel's go-to glyphs to use.

Our most detailed representation is a simplified *piano roll* (fig. 5a). Piano rolls represent notes as rectangles according to their start time (x-axis), pitch (y-axis), and duration (width). This representation is easy to understand, common in music software, and shows the complete raw data of a sample.

Because polyphonic samples have multiple voices, where notes belong to one voice each, previous glyphs did not allow spotting whether samples have similar voices (T3). Thus, we use all notes from one voice to interpolate a line to show the "flow" and approximate the perception of Brachtel, which is why we did not use linear interpolation. We use the mean pitch of each voice to determine their order (lowest, middle, and highest for three voices). Each voice is then visualized as a line on the same layout as the piano roll glyph, always using the same color, which we selected due to separability, for the same position of the voice, for example, blue for the lowest. These *voice line* glyphs (fig. 5d) reveal the potential overlap of voices, are easier to read at first glance than piano rolls, and improve the comparability of samples. As the distance between voices indicates the musical energy, we add a colored bar that uses yellow for high energy and large intervals and purple low energy and small intervals. Therefore, the color indicates the distance of neighboring voices for each step, which allows composers to, for example, quickly spot samples where two voices are similar but shifted by an octave.

Investigating Clusters: When using previous visualizations and thinning out the set of interesting samples (T2), experts were missing information about regions in some cases to exclude them. Therefore, we provide a clustering visualization based on hierarchical clustering and our similarity metric that highlights similar samples with a representative glyph of the region. This visualization allows for selecting and comparing regions and provides aggregated information from all samples in a cluster. First, we show a histogram that depicts the distribution of occurring models (fig. 7a) (T4). Second,

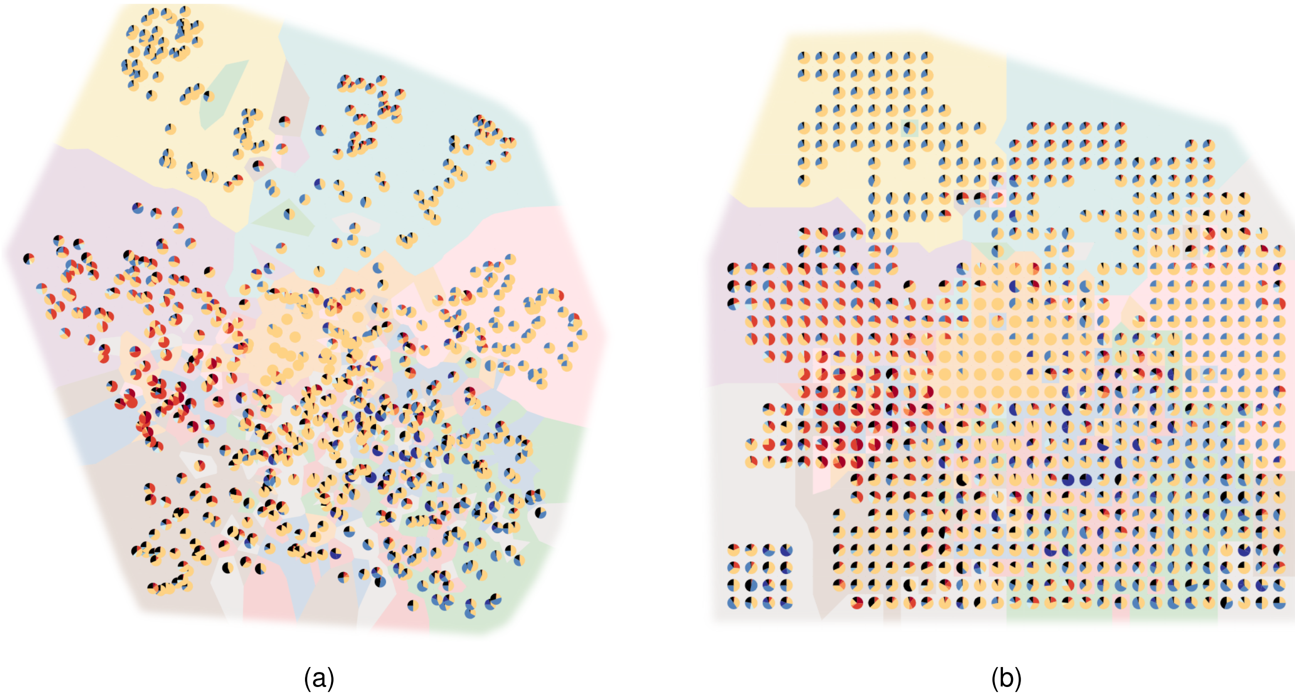


Fig. 6: Two options for a similarity-based layout with rhythm pie glyphs (fig. 5b). The glyphs are laid out by **a)** melodic similarity, **b)** optionally gridified to avoid overlap. In this example, each glyph's background color encodes the source model. While some areas are clearly occupied by a single model (purple area in the left), there are some overlaps (single green sample amongst yellow, top left). The glyphs confirm that neighboring samples have similar rhythmic characteristics, many from the yellow model look almost identical (top of **b)**). Composers can quickly spot outliers or regions of samples with, for example, only short notes (large red parts of glyphs).

a density piano roll displays common and different note occurrences and allows analyzing or comparing the variety of samples in selected clusters (fig. 7b). This especially highlights pitches that occur more or less at specific timings. Last, we show harmonic information (fig. 7c) in the form of occurrence percentages of tonic, dominant, subdominant, in-scale, and out-of-scale notes regarding the selected root note. The tonic, dominant, and subdominant are musical terms and indicate notes that sound well together – their occurrence tells how standard or unique the samples are and how strongly a region follows the rules of music theory.

As polyphonic samples played a big role in one of the compositions, we needed another possibility to explore the space and quickly exclude regions (T2): Due to our method of producing polyphonic samples, the same single voice can occur in many samples, allowing composers to exclude regions if the voice is not interesting. We, therefore, connect every pair of samples that shares the same monophonic sample with a colored line (fig. 8). We use a set of differentiable colors to indicate the ten biggest clusters, allowing composers to find many interesting samples faster. Excluding a big cluster thins out the space more efficiently than excluding small ones. To avoid clutter in dense regions, we apply edge bundling [30].

Analyzing Harmonics: After limiting the space to fewer interesting samples, Brachtel is interested in musical harmonics for multiple root notes (T2, T3). He worries that selecting only one root already biases the representation when not looking for a specific musical key. As previous visualizations cannot show samples for multiple roots, we calculate the timbre metric (section III-E) for each root note (C, C♯, D . . . , B) and display them simultaneously. We use a parallel coordinate plot, where the root notes are used as features and the timbre as value to determine the positions. Thus, each sample occurs 12 times (instead of once like in other layouts) and is colored by timbre from dark (purple) to bright (yellow) – encoding timbre brightness as color brightness. This plot allows composers to follow the sentiment of a single snippet over multiple root notes, as well as select interesting regions based on timbre, ultimately selecting samples that could fit a sentiment or intention. We position the root notes in the sequence of the circle of fifths, as this represents its calculation best and reveals interesting patterns (fig. 9).

Representing Samples in Detail: After selecting samples with the previous visualizations, further investigations can be done via our detail panel, which displays meta information and an extended piano roll (fig. 1d). In this view, Brachtel can also mark samples to export them later. The piano roll shows the raw data – all notes – of the sample and the underlying line of the voice. For monophonic samples, we encode the faithfulness to its timbre in the rectangles' color when a root note is selected: Blue marks a note that contradicts the timbre to highlight outliers that are interesting to him, combining advantages of common staff notation (seeing musical patterns) and piano rolls (all notes have the same visual importance).

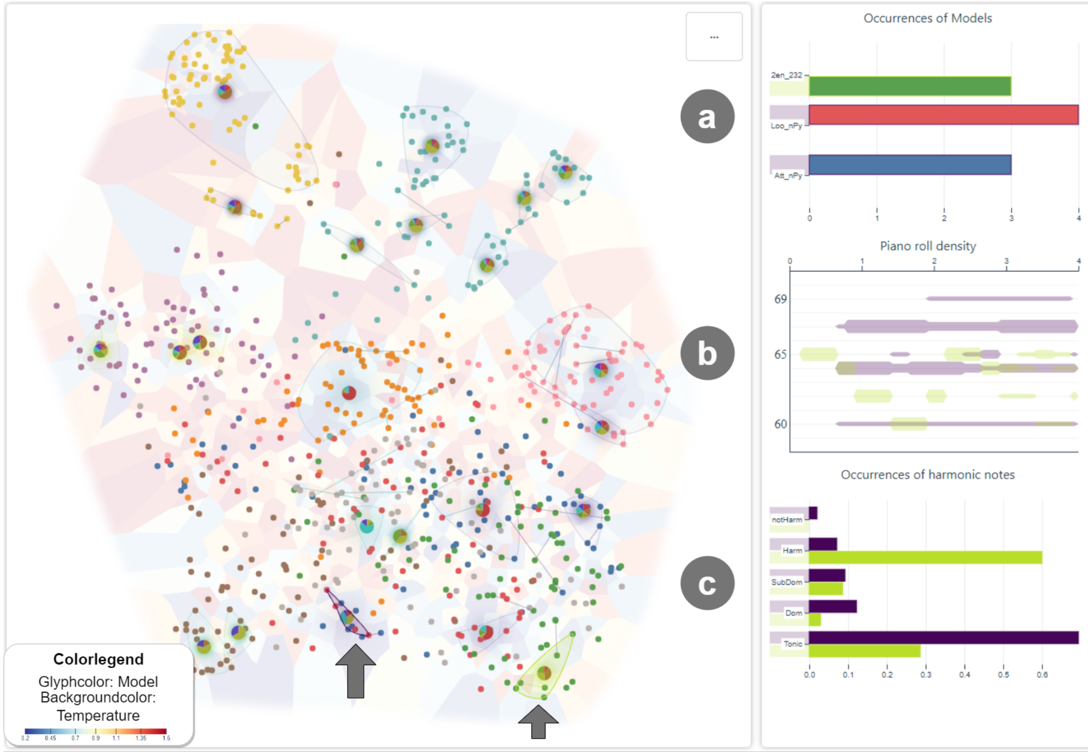


Fig. 7: Similarity-based clustering with two selected clusters at the bottom (arrows). Visualizations on the right show the distribution of a) models, c) harmonic information, and b) the density of piano rolls for each cluster (purple and light green refer to the selected clusters; green, red, and blue to models). Additionally, the heatmap reveals that all samples of the green cluster start identically, while the purple cluster shows more variety but sticks to five different pitches.

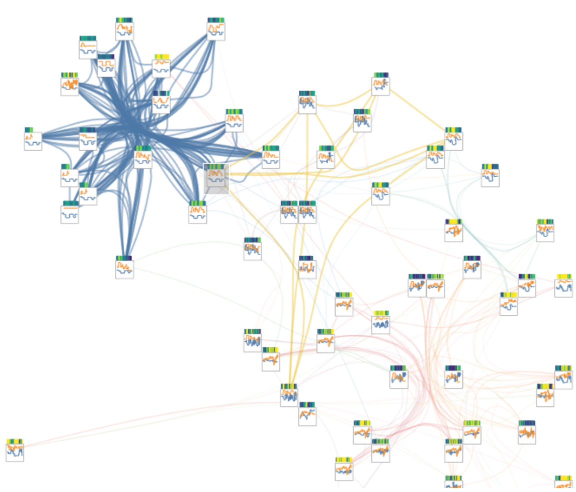


Fig. 8: This representation reveals three big clusters in blue, red, and orange, showing that around 70% of all polyphonic samples share one of these three monophonic samples. Clearly visible is a big cluster of polyphonic samples that share the same voice (blue links), which can all be excluded if their voice is not interesting. The voice line glyphs show interesting samples with clearly separated voices (top left) and overlapping/crossing voices (bottom right)[3].

At the end of each session, he exports all marked samples and can feed some of them back into MAICO as a primer in future sessions to explore their variations.

IV. Evaluation

As our application is highly creative, subjective, and ill-defined, there are no simple quantitative measures to evaluate their success and efficiency. These characteristics are typical for visual analytics evaluation [72], and we thus follow a primarily qualitative evaluation reporting anecdotal evidence from expert users [54]. Below, we first report usage scenarios, which we derived from our initial expert interviews (section III-A). We then reflect on the experience from our five-month multi-dimensional in-depth long-term case study (MILC) [80], followed by interviews with experts from different music backgrounds. We present a summary of the findings.

A. Usage Scenarios

The following scenarios illustrate how MAICO is used to support two common composition tasks in our visual parameter space analysis framework: Variation identification with some diversity (T3, T4, T5) and analysis of rhythmic complexity (T2, T4, T5).

[3]Figures scaled down. In our studies, all visualizations were full-size and readable without any problems. See supplemental material for larger versions.

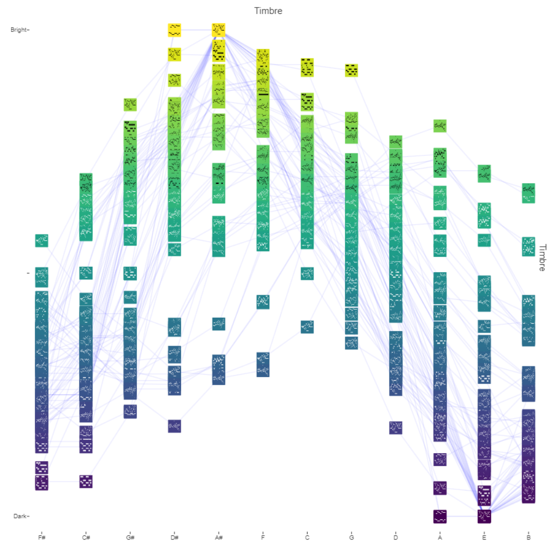


Fig. 9: Harmonic visualization for polyphonic samples using the timbre brightness metric. This example shows the brightest samples at A♯ while E shows the complete opposite, as it is the parallel key. Most samples for C have a bright sentiment[3].

1) Scenario 1 – Variations that Fit the Primer: A typical composition task is finding interesting variations of a motif, in this case, the primer melody (model input). In our case, this task means identifying model outputs that are similar to the initial primer melody and sound interesting. To do so in MAICO, one can first filter samples that are close to the primer. In figure 10a, for instance, doing so reveals that the pink model (melody_rnn) brings up the largest number of similar samples, as indicated by the background color. Overall, we can see that most samples are generated with a lower temperature values, since most glyphs are blue.

Going forward, a composer can then change the layout to encode *similarity to the primer* in the y-position and temperature on the x-position (fig. 10b). The background color shows that the pink model produces the closest samples at the top but sometimes only repeats the primer (similarity is 1), especially at low temperatures. An interesting sample is the right-most one: It is similar to the primer even though it has a high temperature – contradicting the above findings that similar samples usually have low temperatures.

When inspecting the detailed piano roll of this sample (fig. 10c), we can see that it has the same first two bars as the primer but contains a variation in the second part while keeping the rhythm. This sample also uses a F♯, which contradicts the sample's timbre, as indicated by the blue color (in fact, it is the tritone interval to C – long described as the 'devil's interval' – which can add interesting tension and variation to compositions).

Overall, while the pink model generates useful variations at any temperature value, lower values tend to produce rather conservative changes to the primer, and composers can, therefore, prefer higher values for more varied patterns.

2) Scenario 2 – Rhythmic Complexity: Figure 11 shows an example where we want to find samples that have a complex rhythm and investigate the rhythmic capacities of different

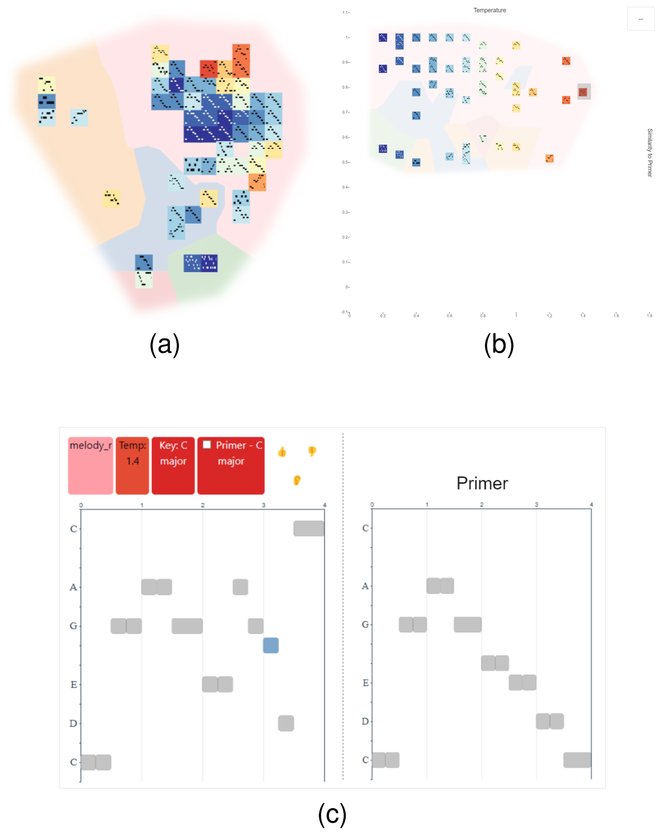


Fig. 10: **Case Study 1:** A step-by-step search for variations of the primer. **(a)** Filtering only samples with the same number of notes and a high similarity (> 0.5) to the primer. Similarity-based layout with piano rolls. Glyph- and background color encode temperature and model. **(b)** The same samples and colors as in a), but with x- and y-axes for temperature and similarity to the primer melody. We can estimate a weak negative correlation between temperature and this similarity. **(c)** We select the right-most sample from (b), which has a similarity to the primer of 0.8. The variation shows a different ending, compared to the primer, including a note that does not benefit the timbre regarding the root note C (blue note).

models in that regard. The different models are encoded by the glyph colors in this example. We can see that the blue model has more diverse characteristics, as indicated by very different glyph shapes for rhythm complexity metrics. In contrast, the yellow and turquoise models at the top show similar glyph patterns across all samples. This overview also shows that the purple model often produces samples with considerable rhythmic changes (right petal), which occurs much more often than in other models. In addition, some samples from the blue and red models (green rectangle in figure 11) contain many off-beat notes (left petal), possibly because they were all produced with a high temperature, as indicated by the reddish background color. Although a high temperature induces more random patterns by design, the visualization shows that the impact of a high temperature differs for different models. In figure 11, only red and blue glyphs react with rather extreme

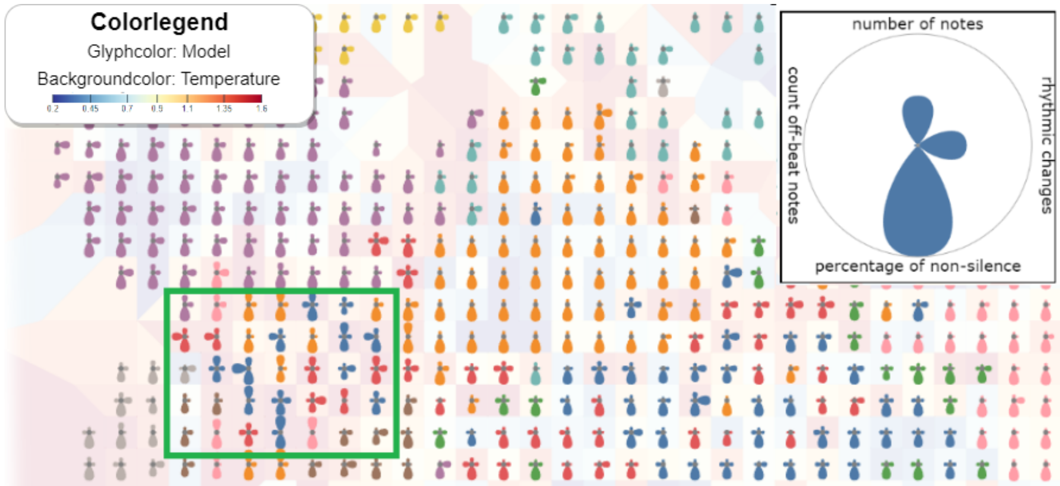


Fig. 11: **Case Study 2:** Flower glyphs with rhythm complexity metrics show regions for different models in a similarity-based layout. These glyphs are colored by model; background colors encode temperature (red for higher). The green rectangle marks an area that contains a few samples with many off-beat notes (larger left petals than most other samples).

changes compared to the grey model. These insights can help to generate specific samples with complex rhythms using the blue model or ones that stick to a rhythmic scheme (turquoise model at the top right).

We discussed these usage scenarios with Brachtel, who used similar approaches in his work with the tool. He confirmed the ecological validity of our scenarios and provided further comments. In this scenario, for instance, he was specifically curious to look at the green-marked region and investigate whether this yields interesting samples due to the extreme rhythm characteristics indicated.

B. Multi-Dimensional In-Depth Long-Term Case Study

Using the guidelines from Shneiderman and Plaisant [80], we conducted a long-term case study (MILC) with Brachtel. Over the course of five months, he incorporated MAICO into his workflow to create two commissioned works: a composition and a theatre piece called "Mosi". After completion, the composition was recorded by the Munich Symphonic Orchestra and it premiered in Venice at the Biennale Arte on April 17th, 2024. "Mosi", on the other hand, was performed live for the first time at the Residenztheater in Munich on April 27th, 2024.

For the Biennale composition, Brachtel continuously used MAICO throughout the composition process, overall, in 47 sessions for a total time of over 31 hours. He generated a total of over 30,000 monophonic samples that represent around 122 hours of music material. Due to technical difficulties, we could only record the data on polyphonic generations for seven sessions, in which around 4,300 polyphonic samples (around 19 hours) were produced using 6,400 monophonic samples. By extrapolating these values, we get around 20,200 polyphonic samples (around 81 hours) for a total of over 200 hours of music material generated, processed, and analyzed in 31 hours – which is only possible with a visual approach.

The final recorded composition is 30 minutes long. Software that was used for other steps in his composition workflow included the DAW Logic Pro (workflow step S4), and the music notation software Sibelius (S5). MAICO was specifically used for finding and elaborating on musical motifs (S2 and S3, section III-B). Brachtel used the motifs and snippets identified with MAICO as inspiration for the composition, adapted and combined them with other ideas, and sometimes also directly used them in the composition.

The play "Mosi" encompassed 75 minutes, 50 of which contained music. In contrast to the previous composition, this piece was not recorded but rather played with live instrumentation during the performance (S6, section III-B), using a monophonic synthesizer (S4) and monophonic MIDI data. To better compensate for limited time and last-minute changes, MAICO was used to produce and select a multitude of variations efficiently, which Brachtel exported as long sequences of monophonic variations with a few "out of the box" samples (S2 and S3), and further added his own ideas. While the composition for the Biennale focused on polyphonic counterpoints, "Mosi" showed a use case of monophonic samples, validating their support in MAICO.

Brachtel initially needed time to get a feeling for using the AI parameters and MAICO but he quickly became accustomed. By *"not changing too many [parameters] at the same time"*, he learned what he did (not) want from our models and how to manipulate them with our filters. After using our tool for around three hours and generating many rounds of samples, he found several interesting samples, which changed his perspective on the content. He found two favorite and one least favorite model, which allowed focusing more on this selection in the future. The main reason for favoring one model was that *"it often produces innovative samples that try something different but also keep elements from the primer"*.

In order to process a generated space even with an increasing amount of data, Brachtel highlighted the importance of visualization multiple times during our meetings to keep a fast and efficient workflow. Because he *"would never listen to more than 20 samples in one session"*, (visually) filtering and excluding samples is needed to thin out the space to the most important regions. To foster this need, he emphasized the value of notational audiation in order to *"hear with your eyes"*. Notational audiation is the ability to tell how a melody sounds simply by looking at their visual notation (sheet music, piano roll). It is a core skill that is taught in the education of professional composers, and Brachtel illustrated it to us many times by easily recognizing melodies of different complexity from their piano roll notation. We further supported fast access to a large exploration space through our different music-driven metrics and visual encodings. Looking at the representation in figure 8, for instance, allowed him to quickly scan the layout and spot interesting samples with large intervals, indicated by the yellow bars, or samples where voices (lines) intersect, which can lead to interesting tonal colors.

His go-to settings were the gridified similarity-based layout with a direct representation of the samples, where the voice line glyphs were superior to the piano roll glyph and the timbre coloring in the background. These settings also allowed him to quickly skim regions for interesting samples using the *"easy-to-read glyphs"*, which are more important than the more precise positions without gridification. These settings allowed for an easy start to every session and provided a good overview. Still, temperature and model coloring were more interesting at the beginning to learn how the models work and which provided the best samples. While his go-to settings were used most of the time, the other visualizations were used situational after thinning out or when analyzing harmonics.

When we asked Brachtel about MAICO's influence on the compositions, he continuously praised the *"great inspiration"* he got from it and the positive impact it had on the composition. Our visualizations helped him, for instance, to select many samples that were *"worth adjusting for specific instruments and putting them to paper"*. He mentioned that two of nine parts of the final Biennale composition explicitly resulted from using MAICO, *"but essentially, the entire composition was [also] influenced by the work with the tool"*. As the final result is a mixture of AI and human influences, he could not tell exactly how much influence AI had and which part was influenced precisely how much by the tool.

As our design got more complex over time, it *"completely fulfilled [Brachtel's] hopes from when we started half a year ago"*. He mentioned that MAICO was not just an experiment to him but a tool that he, from now on, will add to his personal repertoire to use for future compositions.

In conclusion, we verified that visualization can help composers process a large sample space through notational audiation and that our *"approach serves as a tangible, never-ending source of inspiration and helps organize large numbers of samples effectively"*. Brachtel told us that MAICO fits his workflow and *"this will definitely not be the last time [he works] with MAICO"*, but it is important to keep in mind that requirements and needs could be different for other composers. Nevertheless, other artists we talked to about our project mentioned their interest, showing the potential of our approach to be applied to other cases as well.

C. Expert Interviews

An interesting question in design studies relates to the transferability of the resulting design to a broader set of application cases [78]. To test this aspect, we conducted semi-structured interviews with five other composers from different music backgrounds. We contacted 12 professionals, but only five found time during the two-month frame in which the interviews took place. Each of the experts had substantial composition experiences in different genres (techno/electronic music, metal, jazz, theater, and opera). They studied music and have worked professionally as composers, directors, DJs, teachers, and/or musicians for 8 to 23 years. During the interviews, we first showed and explained our prototype, then let them test it and asked for their thoughts on the application, usage scenarios, and requirements to integrate it into their own workflow. Four of the five interviews were conducted online, while the fifth took place in person. The interviews lasted 51 minutes on average. We expected mixed results as composition workflows are substantially different between individuals, and MAICO is thoroughly tailored to Brachtel's workflow. Therefore, we are interested in what features of MAICO work well in a broader context and what would need to change for other composers.

When asked about the application of MAICO in their own work, one expert said he would use it as is, three would use it with further minor adaptations, while only one required more fundamental changes. He would work with it, *"if this program would be an Ableton Live plugin and melodies would be moved directly into the active clip"* (P3).

In terms of how to integrate MAICO into their composition activities, all participants thought of several different approaches, some similar and some different from Brachtel's approaches. Similar to Brachtel's approach, P2 normally uses an iterative approach to create a melody and likes to integrate experimental aspects. As such, he stated that MAICO would suit this style by exploring variations of a motif. Others (P3, P4, P5), however, rather avoid an experimental approach and stick to small variations of good ideas. P5, for instance, wanted to use MAICO to find simple and good motifs to then incorporate musical ornaments on his own or take the ideas a step further: *"I would take this part and repeat it later again"* (P5). On a more general note, P4 stated that a music tool has to be precisely designed for a single step in the workflow. In that sense, all participants mentioned that MAICO would be especially helpful in exploring and finding variations, matching the focus we set out in our user and tasks analysis (section III-B). The main criticism mentioned was on a technical level: *"Switching [between programs] is too big a hurdle"* (P3), and P3 would, thus, rather prefer a direct implementation in the DAW Ableton Live. Integrating new solutions into existing environments is a known challenge in design study research that is often outside the scope of a research project – as is also the case for us [76].

While MAICO uses visualization approaches that are not common in the music domain, every participant understood the general idea and visualizations quickly, even without a prior visualization background. All participants found the similarity-based layout the most helpful for getting an overview, ordering the space, and sorting out unwanted suggestions to avoid going over every sample manually. We found that participants used a combination of visual selection and listening to samples in order to process the generated space, while four of them could imagine how a sample would sound based on visual representations (notational audiation). Especially for polyphonic samples, P2 found the edges between samples (fig. 8) with shared voices helpful to work through the space efficiently. As emotions play a big role in music, P4 found the timbre metric a good start in that direction but would like to have more specific options to create and find a certain emotion. P5 had a clear direction in mind and wanted a dark mood, where he used the timbre metric and the parallel coordinate plot to explore the corpus. He followed certain melodies over different root notes, understood the metric, and ultimately found several dark samples that would fit his envisioned composition. Similar to our previous findings, all participants preferred a direct representation (piano roll or voice line glyphs) of the melody, but *"statistics could also be a possibility to compose new music from"* (P2). Although all participants understood the visualizations, two of them were a bit *"overwhelmed"* (P1) by this *"extensive project"* (P2) and the number of features and representations in the beginning. Both mentioned that it might take some time to learn how to use it efficiently.

As MAICO supports investigating the AI models and their parameters, P2, who worked with generative tools before, underlined the importance of knowing the input-output relation to reduce the frustration of using a black box. All of the participants mentioned that they first have to get a feeling for using the generative models in order to be efficient later on by creating precise results. The features that were noted as most helpful were the colors that helped better understand parameters and the filters, which were mentioned as a good basis for more control. P4 suggested additional rhythm-based filters if the AI does not allow controlling for that.

While polyphonic samples are generally helpful to *"give context to a single voice"* (P5), all participants found that the way of creating polyphonic samples in MAICO is experimental and differs from other common approaches in music theory. One suggestion by P4 was to add polyphonic AI models or stricter rule-based combinations to also support a more traditional way of composition. However, not surprisingly, every participant had their own ideas and came up with their own different approaches. This finding illustrates the individuality in music composition: every musician has different personal preferences, styles, and needs, making it hard to find a good, one-size-fits-all solution.

Generally, all participants found MAICO and its visual approach helpful and praised MAICO's potential to support experimental composition. To better integrate MAICO into their own work – aside from DAW integration – they wished for additional features like better AI control and *"adjusting parameters live during inference"* (P3). Other wishes were improvements to listening to samples like a sequencer, recommendations on interesting samples, and different options to create polyphonic samples with AI or different rules.

V. Discussion

Throughout our design study on AI-assisted music composition, we found increasing evidence that using visual analytics can be a well-suited approach for this application area. AI-assisted music composition is an ill-defined, subjective, and creative process. Therefore, there is no single representation that will cover all aspects of space and music in every case. It is not a process that can be automated in a multi-objective optimization fashion but instead needs to offer flexibility to composers to interact with models and explore the rich output space. These characteristics resemble well the goals and strengths of visual analytics [20].

In addition, we learned that notational audition is a powerful skill that most professional composers exhibit and that offers great potential for visualization approaches. For instance, our voice line and the piano roll glyphs were the most successful representations, as they allow composers to visually "hear" samples without actually listening to them, processing them much faster than by listening.

Another lesson we learned is that many contemporary composers are fine with complex interfaces and are used to feature-rich interfaces with notation software and digital audio workstations. Although some participants from our expert interviews were initially overwhelmed by the number of new concepts, our long-term study showed that learning the interface can lead to effective and successful work. From the beginning, Brachtel was excited about what MAICO's richness and fully supported the complexity that comes with this focus: *"it is so delightful to see so many music samples at once"*. A rich and complex tool gives him new opportunities and allows for the agency needed to articulate compositional creativity. Even with an initial learning curve, he explained that composers are willing to use new complex interfaces outside their comfort zone since *"nowadays, as a composer, you can't get around using complex systems"*.

We also encountered common challenges in working with experts in design studies [77]. It was, for instance, already hard to even start a collaboration with professional composers, as they are often already booked out for months or even years in advance. Therefore, the time one can get with them for long-term studies is also limited, and collaborating only works if a new approach can add immediate benefit.

While we focus on generative music models, the data source is independent of our visualizations. Our approach could similarly work with any other algorithm, ways of creating polyphonic samples, or even with human data like existing compositions sliced into snippets. Except for the model steering (fig. 4) and the connection of polyphonic samples (fig. 8), visualizations can also be used to browse and analyze (T2, T3) a collection of snippets with the same features from any data source. However, parameters like temperature and the tasks related to them (T1, T4, T5) are specific to generative models and the resulting visualizations.

Although our metrics and glyphs are specific to symbolic music snippets, they might generalize to longer symbolic or audio samples of music, and our layouts and visual encodings could also apply to other musical data. Furthermore, the close integration of generative systems and steerability allowed Brachtel to use generative AI in the first place, as switching between too many systems is often a hindrance for professional composers.

Despite instrumentation playing a big role in music, we consciously did not implement different instruments, as they appear in different steps of the workflow and we wanted to avoid biases. We are aware that this might also be a personal preference, and others would combine these steps in their workflows. Therefore, we support the playback of samples on a (hardware or software) synthesizer via MIDI.

A natural limitation of our work is that we heavily focus on a case study with a single expert. Therefore, our work is potentially limited in that specific metrics and visualizations might only be beneficial for this person. Our primary goal is realism through an in-depth investigation of a case study. We do not seek statistical generalizability to larger populations as postulated by controlled experiments [47] and instead echo the many previous calls for in-depth qualitative research for design/case studies like ours [72], [78], [80]. We think that especially the in-depth collaboration with a professional helped us to gain insights and develop the metrics and visualizations that might also benefit other professional users in the long run. In our case, fully-fledged long-term case studies with multiple composers with this level of expertise would not have been feasible, as practices differ substantially between individual composers, and their availability is highly limited. As such, as a next step, it would be interesting to see how MAICO could be transferred to the workflows of other composers. We think that *"everyone might use it differently because the principles are not the usual ones"*, proclaiming interesting characteristics to be studied in the future. Toward the next step of transferability, we have already received additional feedback from four musicians (section III-A) and five different experts (section IV-C). We plan further in-depth investigations with other users in the future.

VI. Conclusion

We propose MAICO, the result of a design study on using visual analytics for AI-assisted music composition to interactively generate, explore, select, edit, and compare samples from generative music models. MAICO's design builds on statistical and musical metrics, based on which we designed similarity-based layouts and glyph representations. Our usage scenarios, MILC study with the professional composer Brachtel, and expert interviews demonstrated the utility of our design. He has adopted MAICO into his toolbox and continues to use it in commissioned compositions for inspiration and efficient sample selection. For him, MAICO is a professional software, a *"new component of his toolchain, not just an experiment"*, and he *"hope[s] it stays online like this forever"*. His work with MAICO showed that visualization, via notational audiation, can contribute to music composition, an ill-defined and subjective application area with a data-rich process.

Future extensions could add other models, including polyphonic models and the integration of more powerful prompt- [1], description- [88], or theme-based [79] models, that would improve precise generation for composers. Extensions for manual labeling and action analysis could benefit recommendations on interesting samples or models to enhance exploration, interactivity, and efficiency. This would allow for a personalized version of MAICO that generates better-matching samples for individual composing styles. To explore the flexibility of MAICO, we plan to do a study with further composers and projects to investigate other workflows and usage scenarios. As MAICO focuses on specific tasks and steps in the workflow, one could create further utensils for artists' toolboxes or extend MAICO, respectively. Such extensions could support other musical features, such as dynamics, instrumentation, or composition techniques, like spectralism.

Acknowledgments

Funded by the Ministry of Science, Research and the Arts Baden-Wuerttemberg in the Artificial Intelligence Software Academy (AISA) and the Cyber Valley Research Fund.

References

[1] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, et al. MusicLM: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

[2] S. Amershi, M. Chickering, S. M. Drucker, B. Lee, et al. ModelTracker: Redesigning performance analysis tools for machine learning. In *Proc. Conf. Human Factors in Computing Systems (CHI)*, pages 337–346, 2015.

[3] R. Arias-Hernandez, L. T. Kaastra, T. M. Green, and B. Fisher. Pair analytics: Capturing reasoning processes in collaborative visual analytics. In *Proc. Hawaii International Conf. System Sciences (HICSS)*, pages 1–10, 2011.

[4] T. Bazin and G. Hadjeres. NONOTO: A model-agnostic web interface for interactive music composition by inpainting. In *Proc. International Conf. Computational Creativity (ICCC)*, pages 89–91, 2019.

[5] R. Borgo, J. Kehrer, D. H. S. Chung, E. Maguire, et al. Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. In *Eurographics (State of the Art Reports)*, pages 39–63, 2013.

[6] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner. Visualizing dimensionally-reduced data: interviews with analysts and a characterization of task sequences. In *Proc. Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV)*, page 1–8, 2014.

[7] J.-P. Briot, G. Hadjeres, and F.-D. Pachet. *Deep Learning Techniques for Music Generation*. 2020.

[8] W. Brodsky, Y. Kessler, B.-S. Rubinstein, J. Ginsborg, et al. The mental representation of music notation: notational audiation. *Journal of Experimental Psychology: Human Perception and Performance*, pages 427–445, 2008.

[9] D. Cashman, S. R. Humayoun, F. Heimerl, K. Park, et al. A user-based visual analytics workflow for exploratory model analysis. *Computer Graphics Forum (CGF)*, pages 185–199, 2019.

[10] B. Chang, G. Sun, T. Li, H. Huang, et al. MUSE: Visual analysis of musical semantic sequence. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, pages 4015–4030, 2022.

[11] L.-Y. Chiou, P.-K. Hung, R.-H. Liang, and C.-T. Wang. Designing with ai: An exploration of co-ideation with image generators. In *Proc. ACM Designing Interactive Systems Conference (DIS)*, page 1941–1954, 2023.

[12] M. Civit, J. Civit-Masot, F. Cuadrado, and M. J. Escalona. A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends. *Expert Systems with Applications*, 2022.

[13] M. Costa, P. Fine, and P. Ricci Bitti. Interval distributions, mode, and tonal strength of melodies as predictors of perceived emotion. *Music Perception (MP)*, pages 1–14, 2004.

[14] R. Cutura, C. Kralj, and M. Sedlmair. Druid$_{JS}$ – a JavaScript library for dimensionality reduction. In *Proc. IEEE Visualization Conference (VIS)*, pages 111–115, 2020.

[15] R. Cutura, C. Morariu, Z. Cheng, Y. Wang, et al. Hagrid —- gridify scatterplots with Hilbert and Gosper curves. In *Proc. International Symp. Visual Information Communication and Interaction (VINCI)*, pages 1–8, 2021.

[16] A. Dash and K. Agres. AI-based affective music generation systems: A review of methods, and challenges. *ACM Computing Surveys*, pages 1–34, 2024.

[17] R. T. Dean and J. Forth. Towards a deep improviser: A prototype deep learning post-tonal free music generator. *Neural Computing and Applications*, pages 969–979, 2020.

[18] E. Deruty, M. Grachten, S. Lattner, J. Nistal, et al. On the development and practice of AI technology for contemporary popular music production. *Trans. International Society for Music Information Retrieval (TISMIR)*, pages 35–50, 2022.

[19] 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. In *Proc. Conf. Artificial Intelligence (AAAI)*, pages 34–41, 2018.

[20] A. Endert, M. S. Hossain, N. Ramakrishnan, C. North, et al. The human is the loop: New directions for visual analytics. *Journal of Intelligent Information Systems*, pages 411–435, 2014.

[21] C. Flick and K. Worrall. The Ethics of Creative AI. In *The Language of Creative AI: Practices, Aesthetics and Structures*, pages 73–91. 2022.

[22] D. W. Fourney and D. I. Fels. Creating access to music through visualization. In *Proc. International Conf. Science and Technology for Humanity (TIC-STH)*, pages 939–944, 2009.

[23] A.-M. Gioti. From artificial to extended intelligence in music composition. *Organised Sound*, page 25–32, 2020.

[24] A.-M. Gioti. Artificial intelligence for music composition. *Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity*, pages 53–73, 2021.

[25] R. Guo, I. Simpson, C. Kiefer, T. Magnusson, et al. MusIAC: An extensible generative framework for music infilling applications with multi-level control. In *Proc. International Conf. EvoMUSART (EvoStar)*, pages 341–356, 2022.

[26] G. Hadjeres and F. Nielsen. Anticipation-RNN: Enforcing unary constraints in sequence generation, with application to interactive music generation. *Neural Computing and Applications*, pages 995–1005, 2020.

[27] C. Hernandez-Olivan and J. R. Beltrán. Music composition with deep learning: A review. *Advances in Speech and Music Technology: Computational Aspects and Applications*, pages 25–50, 2023.

[28] F. Heyen, T. Munz, M. Neumann, D. Ortega, et al. ClaVis: An interactive visual comparison system for classifiers. In *Proc. International Conf. Advanced Visual Interfaces (AVI)*, pages 1–9, 2020.

[29] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions Visualization and Computer Graphics (TVCG)*, pages 2674–2693, 2019.

[30] D. Holten and J. J. Van Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum (CGF)*, pages 983–990, 2009.

[31] C. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinculescu, et al. AI song contest: Human-AI co-creation in songwriting. In *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, pages 708–716, 2020.

[32] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, et al. Music Transformer: Generating music with long-term structure. In *Proc. International Conf. Learning Representations (ICLR)*, pages 1–13, 2019.

[33] R. Khulusi, J. Kusnick, C. Meinecke, C. Gillmann, J. Focht, and S. Jänicke. A Survey on Visualizations for Musical Data. *Computer Graphics Forum (CGF)*, pages 82–110, 2020.

[34] S. Knotts and N. Collins. A survey on the uptake of music AI software. In *Proc. International Conf. New Interfaces for Musical Expression (NIME)*, pages 499–504, 2020.

[35] E. S. Koh, S. Dubnov, and D. Wright. Rethinking recurrent latent variable model for music composition. In *Proc. IEEE International Workshop Multimedia Signal Processing (MMSP)*, pages 1–6, 2018.

[36] C. L. Krumhansl. *Cognitive foundations of musical pitch.* Oxford University Press, 1990.

[37] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, pages 1–27, 1964.

[38] J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, pages 115–129, 1964.

[39] M. V. Lakshitha and K. Jayaratne. Melody analysis for prediction of the emotions conveyed by Sinhala songs. In *Proc. IEEE International Conf. Information and Automation for Sustainability (ICIAfS)*, pages 1–6, 2016.

[40] H. B. Lima, C. G. R. D. Santos, and B. S. Meiguins. A Survey of Music Visualization Techniques. *ACM Computing Surveys*, pages 1–29, 2021.

[41] O. Lopez-Rincon, O. Starostenko, and G. A.-S. Martín. Algoritmic music composition based on artificial intelligence: A survey. In *Proc. International Conf. Electronics, Communications and Computers (CONIELECOMP)*, pages 187–193, 2018.

[42] R. Louie, A. Coenen, C. Z. Huang, M. Terry, et al. Novice-AI music co-creation via AI-steering tools for deep generative models. In *Proc. Conf. Human Factors in Computing Systems (CHI)*, page 1–13, 2020.

[43] R. Louie, J. Engel, and A. Huang. Expressive communication: A common framework for evaluating developments in generative models and steering interfaces. *arXiv preprint arXiv:2111.14951*, 2021.

[44] J. Lu, W. Chen, Y. Ma, J. Ke, et al. Recent progress and trends in predictive visual analytics. *Frontiers of Computer Science*, pages 192–207, 2017.

[45] J. A. T. Lupker. Score-Transformer: A deep learning aid for music composition. In *Proc. International Conf. New Interfaces for Musical Expression (NIME)*, 2021.

[46] R. McGee. Using artificial intelligence (AI) to compose a musical score for a Tai Chi tournament routine: A ChatGPT experiment. *SSRN*, pages 1–11, 2023.

[47] J. E. McGrath. Methodology matters: Doing research in the behavioral and social sciences. In *Readings in human–computer interaction*, pages 152–169. 1995.

[48] L. McInnes, J. Healy, N. Saul, and L. Großberger. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software (JOSS)*, page 861, 2018.

[49] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, et al. SampleRNN: An unconditional end-to-end neural audio generation model. In *Proc. International Conf. Learning Representations (ICLR)*, 2017.

[50] A. Micallef Grimaud and T. Eerola. An Interactive Approach to Emotional Expression Through Musical Cues. *Music & Science*, 2022.

[51] G. Micchi, L. Bigo, M. Giraud, R. Groult, et al. I keep counting: An experiment in human/AI co-creative songwriting. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, pages 263–275, 2021.

[52] M. Miller, D. Fürst, H. Hauptmann, D. A. Keim, et al. Augmenting digital sheet music through visual analytics. *Computer Graphics Forum (CGF)*, pages 301–316, 2022.

[53] M. Miller, J. Rauscher, D. A. Keim, and M. El-Assady. CorpusVis: Visual analysis of digital sheet music collections. *Computer Graphics Forum (CGF*, pages 283–294, 2022.

[54] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions Visualization and Computer Graphics (TVCG)*, pages 921–928, 2009.

[55] A. Nairn and J. Matthews. Exploring the integration of art and technology. *Journal of Pervasive Media (JPM)*, pages 3–10, 2023.

[56] S. Nicholls, S. Cunningham, and R. Picking. Collaborative artificial intelligence in music production. In *Proc. Audio Mostly (AM)*, pages 1–4, 2018.

[57] S. Oore, I. Simon, S. Dieleman, and D. Eck. Learning to create piano performances. In *Proc. NIPS Workshop on Machine Learning and Creativity*, 2017.

[58] S. Park, T. Kwon, J. Lee, J. Kim, et al. A cross-scape plot representation for visualizing symbolic melodic similarity. In *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, pages 423–430, 2019.

[59] A. Pati, A. Lerch, and G. Hadjeres. Learning to traverse latent spaces for musical score inpainting. In *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, pages 343–351, 2019.

[60] N. Privato, O. Rampado, and A. Novello. A creative tool for the musician combining LSTM and Markov chains in Max/MSP. In *Proc. International Conf. EvoMUSART (EvoStar)*, page 228–242, 2022.

[61] Y. Qiu, J. Zhang, H. Ren, Y. Shan, et al. Humming2music: Being a composer as long as you can humming. In *Proc. International Joint Conf. Artificial Intelligence (IJCAI)*, pages 7163–7166, 2023.

[62] S. Rau, F. Heyen, S. Wagner, and M. Sedlmair. Visualization for AI-assisted composing. In *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, pages 151–159, 2022.

[63] R. Redgate. Do you hear what I hear? audiation and the compositional process. *Principles of Music Composing*, pages 19–27, Nov. 2019.

[64] D. Ren, S. Amershi, B. Lee, J. Suh, et al. Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE Transactions Visualization and Computer Graphics (TVCG)*, pages 61–70, 2017.

[65] J. Rezwana and M. L. Maher. Designing creative AI partners with COFI: A framework for modeling interaction in human-AI co-creative systems. *Trans. Compututer-Human Interaction (TOCHI*, pages 1–28, 2023.

[66] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, et al. A hierarchical latent vector model for learning long-term structure in music. In *Proc. International Conf. Machine Learning (PMLR)*, pages 4364–4373, 2018.

[67] A. Roberts, C. Kayacik, C. Hawthorne, D. Eck, J. Engel, M. Dinculescu, and S. Nørly. Magenta studio: Augmenting creativity with deep learning in ableton live. In *Proc. International Workshop on Musical Metacreation (MUME)*, 2019.

[68] D. Sacha, M. Kraus, D. A. Keim, and M. Chen. Vis4ml: An ontology for visual analytics assisted machine learning. *IEEE Transactions Visualization and Computer Graphics (TVCG)*, pages 385–395, 2019.

[69] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, et al. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Transactions Visualization and Computer Graphics (TVCG)*, pages 241–250, 2017.

[70] E. Schellenberg, A. Krysciak, and R. Campbell. Perceiving emotion in melody: Interactive effects of pitch and rhythm. *Music Perception (MP)*, pages 155–171, 2000.

[71] A. Schoenberg. *Style and Idea: Selected Writings of Arnold Schoenberg*. Univ of California Press, 1975.

[72] J. C. Scholtz. *User-Centered Evaluation of Visual Analytics*. Oct. 2017.

[73] E. Schubert. Modeling perceived emotion with continuous musical features. *Music Perception (MP)*, pages 561–585, 2004.

[74] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton. Real-time corpus-based concatenative synthesis with catart. In *Proc. Conf. Digital Audio Effects (DAFx)*, 09 2006.

[75] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, et al. Visual parameter space analysis: A conceptual framework. *IEEE Transactions Visualization and Computer Graphics (TVCG)*, pages 2161–2170, 2014.

[76] M. Sedlmair, P. Isenberg, D. Baur, and A. Butz. Information visualization evaluation in large companies: Challenges, experiences and recommendations. *Information Visualization*, pages 248–266, 2011.

[77] M. Sedlmair, P. Isenberg, D. Baur, and A. Butz. Information visualization evaluation in large companies: Challenges, experiences and recommendations. *Information Visualization*, pages 248–266, 2011.

[78] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions Visualization and Computer Graphics (TVCG)*, pages 2431–2440, 2012.

[79] Y.-J. Shih, S.-L. Wu, F. Zalkow, M. Muller, et al. Theme transformer: Symbolic music generation with theme-conditioned transformer. *IEEE Transactions on Multimedia (TOM)*, pages 1–1, 2022.

[80] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proc. Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV)*, page 1–7, 2006.

[81] M. Suh, E. Youngblom, M. Terry, and C. J. Cai. AI as social glue: Uncovering the roles of deep generative AI during social music composition. In *Proc. Conf. Human Factors in Computing Systems (CHI)*, pages 1–11, 2021.

[82] D. Temperley. What's key for key? the Krumhansl-Schmuckler key-finding algorithm reconsidered. *Music Perception (MP)*, pages 65–100, 1999.

[83] W. Thompson. Intervals and scales. *The Psychology of Music*, pages 107–140, 2013.

[84] E. Thul. *Measuring the Complexity of Musical Rhythm*. PhD thesis, McGill University, 2008.

[85] G. T. Toussaint and K. Trochidis. On measuring the complexity of musical rhythm. In *Proc. Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 753–757, 2018.

[86] P. A. Tremblay, G. Roma, and O. Green. Enabling programmatic data mining as musicking: The fluid corpus manipulation toolkit. *Computer Music Journal*, pages 9–23, 2021.

[87] C. van Onzenoodt, P.-P. Vázquez, and T. Ropinski. Out of the plane: Flower vs. star glyphs to support high-dimensional exploration in two-dimensional embeddings. *IEEE Transactions Visualization and Computer Graphics (TVCG)*, pages 1–15, 2022.

[88] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hoffman. FIGARO: generating symbolic music with fine-grained artistic control. In *Proc. International Conf. Learning Representations (ICLR)*, 2023.

[89] J. Wang, S. Liu, and W. Zhang. Visual analytics for machine learning: A data perspective survey. *IEEE Transactions Visualization and Computer Graphics (TVCG)*, pages 1–20, 2024.

[90] J. Waser, R. Fuchs, H. Ribičič, B. Schindler, et al. World lines. *IEEE Transactions Visualization and Computer Graphics (TVCG)*, pages 1458–1467, 2010.

[91] S. Wei, Z. Wang, W. Gao, and G. Xia. Controllable music inpainting with mixed-level and disentangled representation. In *Proc. International Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

[92] Y. Wu, L. Zhang, N. Bryan-Kinns, and M. Barthet. Open Symphony: Creative Participation for Audiences of Live Music Performances. *IEEE MultiMedia*, pages 48–62, 2017.

[93] Z. Xiong, W. Wang, J. Yu, Y. Lin, and Z. Wang. A comprehensive survey for evaluation methodologies of ai-generated music. *arXiv preprint arXiv:2308.13736*, 2023.

[94] Y. Zhang, G. Xia, M. Levy, and S. Dixon. COSMIC: A conversational interface for human-AI music co-creation. In *Proc. International Conf. New Interfaces for Musical Expression (NIME)*, 2021.

VII. Biography Section

Simeon Rau received his master's degree in computer science in 2021 from the University of Stuttgart and is currently working toward a PhD degree. His research focuses on the intersection of information visualization and visual analytics, artificial intelligence, and music composition, ultimately giving music composers access to novel technology and visual feedback.

Frank Heyen finished his master's degree in computer science in 2019 from the University of Stuttgart and is currently pursuing a PhD degree. He explores how visualization can support musicians. His research includes visually encoding sheet music structure, instrument practice data visualization on screen or in augmented reality, and interactive composition through visualization of AI output.

Benedikt Brachtel studied jazz guitar and jazz composition as well as classical composition and has been a freelance composer and music producer since 2010. His clients include the Bavarian State Opera, the IFA (Institut für Auslandsbeziehungen), and the Burgtheater Vienna. Brachtel regularly encounters contemporary means of compositional creation and musical performance due to the versatility of his artistic projects.

Michael Sedlmair is a professor of Computer Science at the University of Stuttgart and leads the research group for Human-Computer Interaction. His research interests focus on visualization, augmented and virtual reality, and interaction design. Michael has been an active musician for more than 30 years, including seven years of teaching guitar as a tutor at a music school.