EXPLORING THE INNER MECHANISMS OF LARGE GENERATIVE MUSIC MODELS

Marcel A. Vélez Vásquez* **Charlotte Pouw*** **John Ashley Burgoyne**
Willem Zuidema
Institute for Logic, Language and Computation, University of Amsterdam
{m.a.velezvasquez,c.m.pouw,j.a.burgoyne,w.h.zuidema}@uva.nl

ABSTRACT

Generative models are starting to become very good at generating realistic text, images, and even music. Identifying how exactly these models conceptualize data has become crucial. To date, however, interpretability research has mainly focused on the text and image domain, leaving a gap in the music domain. In this paper, we investigate the transferability of straightforward text-oriented interpretability techniques to the music domain. Specifically, we examine the usability of these techniques for analyzing how the generative music model MusicGen constructs representations of human-interpretable musicological concepts. Using the DecoderLens, we gain insight into how the model gradually composes these concepts, and using interchange interventions, we observe the contributions of individual model components in generating the sound of specific instruments and genres. We also encounter several shortcomings of the interpretability techniques for the music domain, which underscore the complexity of music and need for proper audio-oriented adaptation. Our research marks an initial step toward understanding generative music models, *fundamentally*, paving the way for future advancements in controlling music generation.

1. INTRODUCTION

Generative AI systems for music have become mainstream in the past year, and have become a popular application for consumers, an eye-catching product for AI engineers and companies, and a key research topic for researchers. The most successful of these systems are built on top of recent advances in deep learning for text and audio encoding, and add a large *text-to-music* model, using the Transformer-architecture [1], to allow users to generate music from a text and/or audio prompt [2–5].

These systems are typically trained end-to-end, and present us with the infamous *black box problem*: it is extremely difficult to understand what is happening in the billions of mathematical operations between input and generated output. This severely limits the ability of users to influence the generated output (other than by just trying a different prompt), of companies to trace an individual output to examples from the training set (and give credit where credit is due), of engineers to diagnose shortcomings and improve the system (other than by retraining on a better dataset or bigger model) and of music researchers to relate the behavior of these models to the large body of existing theoretical and empirical work on how music works.

'Opening the black box' of generative music models is therefore a key new area of research. In this paper, we build on advances with interpretability techniques for generative text models. Although there are many important differences between text and music (including their discrete versus continuous nature, and the temporal resolution needed to build good models), we find that those techniques can be adapted to the music domain and indeed give us insights into the inner mechanisms. We focus on one representative, open-source generative music model, MusicGen [5], and on two representative human-interpretable concepts: *musical instrument* and *genre*. We ask: can we localize and manipulate those concepts in MusicGen? We report success on these tasks, and discuss in the final parts of the paper how these initial steps might be extended to the full toolbox needed to successfully address the negative consequences of the black box problem.

2. RELATED WORK

Interpretability research is relatively sparse in the music domain. Previous work has analyzed neural models trained on symbolic music representations (MIDI) using probing classifiers [6, 7], visual inspection of the embedding space [8], listenable explanations for classification models [9, 10], or post-hoc explanations in the form of highlighted parts of a piano roll [11]. To the best of our knowledge, no previous work has tried to interpret generative music models trained on raw audio data.

In the text domain, the Transformer architecture is dominating the field [1]. Recent advancements in interpretability methods are built on a key characteristic of Transformer models: their use of residual connections across layers. Typically, each layer of the Transformer contains an attention component and a Multilayer Perceptron (MLP), both

[*] These authors contributed equally to this work.

Attribution: M. A. Vélez Vásquez, C. Pouw, J. A. Burgoyne, and W. Zuidema, "Exploring the inner mechanisms of large generative music models", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.,* San Francisco, United States, 2024.

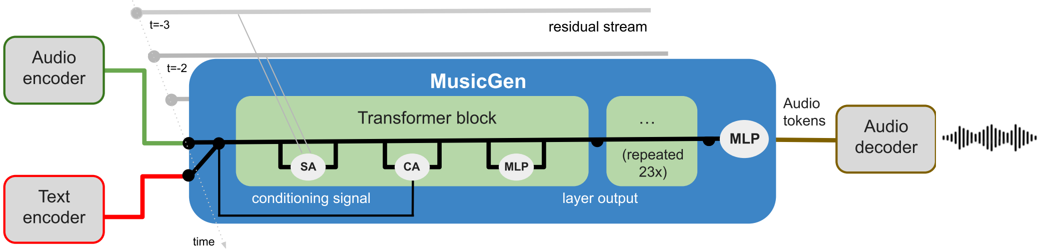

Figure 1: Architecture of MusicGen. SA = self-attention, CA = cross-attention, MLP = Multi-Layer Perceptron. The self-attention allows for communication between audio tokens; the cross-attention allows for communication between the audio tokens and the conditioning signal (consisting of an encoded text prompt and optionally an encoded melody prompt).

of which interact with the residual stream. This arrangement ensures that information in the residual stream remains accessible throughout all layers, facilitating the development of stable representations.

One method that exploits this characteristic is the **DecoderLens** [12]. It is an adaptation of the LogitLens [13], which was designed to interpret intermediate representations of decoder-only Transformer models. It applies the unembedding matrix to intermediate layer outputs to obtain a logit distribution over the vocabulary for each intermediate layer. The DecoderLens was designed to interpret the intermediate representations of encoder-decoder models. It applies the decoder to intermediate encoder outputs, providing insight in what information that can be decoded from earlier layers. In the original DecoderLens study, the authors use the DecoderLens to analyze how encoder–decoder models build meaningful representations for tasks like machine translation and question answering.

Additionally, **interchange interventions** have been used to identify model components that are causally involved in specific behavior, such as *greater-than* reasoning [14], pronoun resolution [15], and gender bias [16, 17]. By systematically altering model inputs or components and observing resultant changes in behavior, researchers have gained valuable insights into the underlying mechanisms driving model performance and decision making.

3. EXPERIMENTAL SETUP

We conduct two interpretability experiments, one using the DecoderLens and one using interchange interventions, for interpreting the inner workings of a popular generative music model, MusicGen [5].

MusicGen is an open-source, Transformer-based music generation model, built by researchers at Meta [5]. Its architecture is sketched in Figure 1. The model generates discrete audio tokens, optionally conditioned on a text prompt and/or a melody. Text prompts are first processed by a Transformer-based text encoder; music prompts by a 32-kHz EnCodec [18] tokenizer sampled at 50 Hz. MusicGen is autoregressive, and transforms its input over successive layers through multi-head *self-attention* (integrating information across timesteps) and MLP components, while incorporating information from the text and/or melody prompt through *cross-attention* to the respective encoders. The generated audio tokens are then decoded into a waveform by an EnCodec decoder.

We analyze three model sizes: **MusicGen-small** (300M parameters, 1024 dimensions), **MusicGen-medium** (1.5B parameters, 1536 dimensions), and **MusicGen-large** (3.3B parameters, 2048 dimensions). We set the generation duration to 4 seconds and keep the rest of the MusicGen parameters at their default values.*

In all of our experiments, we only condition MusicGen on text prompts, not on melody prompts. We constructed the following template for our text prompts: *Compose a [MOOD] [GENRE] piece with a [INSTRUMENT] melody. Use a [TEMPO] tempo.* We only modify the components in between brackets and keep the remaining context fixed.

3.1 Experiment 1: DecoderLens

In our first experiment, we use the DecoderLens to globally examine how MusicGen builds up representations of musicological concepts across its Transformer layers. This involves extracting intermediate representations from each layer and using the EnCodec decoder to map these to audio. We examine the representation of *musical instrument* and *genre*. We select four instruments and six genres (listed in Table 1) and construct 100 text prompts per category using our predefined template. We feed these prompts to MusicGen and use the DecoderLens to obtain 100 music outputs for each of the 24 Transformer layers.

We evaluate the **recognizability** of our selected musicological concepts within the intermediate music outputs by employing an audio classifier that was among the top-ranking classifiers of the 2021 HEAR challenge [19]. This multi-label classifier, trained on AudioSet [20], provides a logit distribution across 527 audio classes, including both musicological concepts and other sounds such as speech and environmental noises. We run each intermediate music output through the audio classifier and compute the **normalized discounted cumulative gain (NDCG)** [21], a metric commonly used in information retrieval to measure ranking quality. We consider this to be a proxy for the recognizability of a specific concept.

For each concept, we establish an 'ideal ranking' of the audio classes by assigning relevant labels (listed in Table

*For our code and listening examples, see our GitHub page: https://github.com/Marcel-Velez/musicgen-mech-interp

Category	Selection	Relevant Labels
Instrument	Guitar	Guitar, Acoustic Guitar, Electric Guitar, Bass Guitar, Plucked String Instrument
	Piano	Piano, Electric Piano, Keyboard (musical)
	Trumpet	Trumpet, Brass Instrument
	Violin	Violin/Fiddle, String Section, Bowed String Instrument
Genre	Classical	Classical Music
	Jazz	Jazz, Rhythm and Blues
	Pop	Pop Music
	Rock	Rock Music, Rock and Roll, Progressive Rock, Punk Rock
	EDM	Electronic Dance Music, Electronic Music, Techno, Drum and Bass, Dubstep, House Music
	Hip Hop	Hip Hop Music

Table 1: Relevant labels of the external audio classifier [19] for each category.

1) a relevance score of 1, while assigning all other labels a score of 0. This methodology facilitates a comparison between the predicted ranking generated by the audio classifier and our predefined ideal ranking. NDCG returns a high score when the relevant labels from our ideal ranking are ranked high by the audio classifier, with a score of 1.0 indicating a perfect predicted ranking.

3.2 Experiment 2: Interchange Interventions

In our second experiment, we use interchange interventions to identify the crucial model components responsible for generating specific musical instrument and genre sounds. The workflow for performing these interventions, which we apply for every permutation of two categories in Table 1, is as follows.

1. Construct two sets of text prompts: one for a concept such as *guitar* (henceforth the **original concept**), and one for a contrasting concept such as *piano* (henceforth the **desired concept**).

2. Run both sets of prompts through MusicGen and save the output of each individual component within the MusicGen Transformer (these model components are further explained in section 3.2.1). This leaves us with two activation caches: one for the original concept, and one for the desired concept.

3. While running MusicGen on the original concept prompts again, replace the output of a specific model component with the average output of that model component across all desired concept prompts. After the intervention, the forward pass continues as normal, but yields an **intervened music fragment**. Repeat this step for all model components.

4. To evaluate the effect of each individual intervention, run a classifier on the original and intervened music fragment, and assess how the odds for the original and desired concept labels changed (we use the same audio classifier that we used for our DecoderLens experiments). If the intervention was effective, the odds for the original concept should have decreased, and the odds for the desired concept should have increased.

3.2.1 Intervention techniques

We explore two intervention techniques: **replace** and **adjust**. With the "replace" technique, we entirely substitute an activation from an individual *original concept* prompt with the average activation of 100 *desired concept* prompts. With the "adjust" technique, we first subtract the average activation of 100 *original concept* prompts from an individual *original concept* activation. Then, we add the average activation of 100 *desired concept* prompts to that result. The latter technique is inspired by the idea that, in language models, semantic properties of words can be adjusted by adding or subtracting specific word vectors, e.g., *king – man + woman = queen* [22], or in our case, *music with guitar – guitar + piano = music with piano*.

We perform the interchange interventions across all 24 Transformer layers of MusicGen. Each layer consists of a self-attention block, a cross-attention block, and a Multi-Layer Perceptron (MLP). Thus, each intervention consists of swapping the output of one of these three components per layer individually. For a single text prompt, this adds up to 24 layers × 3 layer components = 72 interventions.

3.2.2 Within-category vs. cross-category interventions

We investigate intervention effects on both **instrument** prompts and **genre** prompts. For each instrument and genre category listed in Table 1, we construct 100 text prompts based on the template outlined in Section 3. We then perform **within-** and **cross-category** interventions.

In within-category interventions, we interchange model activations between two sets of instrument prompts, or between two sets of genre prompts. For instance, we introduce *piano* activations during a forward pass intended for *guitar*, or we introduce *jazz* activations during a forward pass intended for *classical*.

In cross-category interventions, we interchange model activations between a set of instrument and a set of genre prompts. For example, we introduce *piano* activations during a *classical* forward pass, or we introduce *jazz* activations during a *guitar* forward pass.

3.2.3 Evaluating Intervention Effects

An ideal intervention removes the original concept and introduces the desired concept, but does not change anything about the rest of the music. We therefore evaluate the interventions along two axes: **intervention effectiveness** and **intervention precision**.

We evaluate intervention effectiveness using a metric that quantifies the impact on both the original and desired concept, inspired by the metric used in [23]. Specifically, we calculate:

$$\log \frac{\text{odds}(\text{original}_{\text{before}})}{\text{odds}(\text{original}_{\text{after}})} - \log \frac{\text{odds}(\text{desired}_{\text{before}})}{\text{odds}(\text{desired}_{\text{after}})} \quad (1)$$

A high score indicates that the odds of the original concept decreased as a result of the intervention, or that the odds

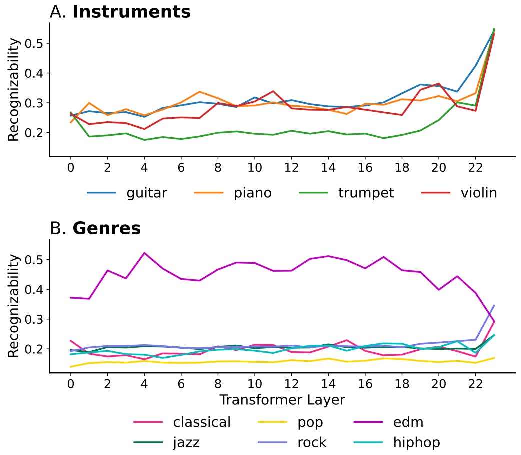

Figure 2: Results for Experiment 1 (**DecoderLens**): Average recognizability of instruments (A) and genres (B) across Transformer layers in MusicGen-small (as measured by the NDCG).

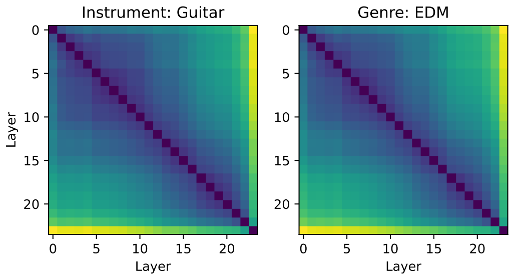

Figure 3: Self-similarity matrices (Euclidean distance) of intermediate layer outputs of the Transformer block within MusicGen-small, when conditioning the model on text prompts containing the instrument *guitar* (left) or containing the genre *EDM* (right). Both matrices are averaged over 100 prompts. A dark color indicates high similarity, a light color indicates low similarity.

of the desired concept increased. We calculate odds by applying a softmax function over the logit distribution of our external audio classifier.*

We evaluate intervention precision using the **Kullback-Leibler (KL) divergence**, which quantifies the overall shift in softmax distribution across all audio labels. This metric gauges how much the intervened music fragment differs from the original. Ideally, our intervention only has an effect on the odds for the original and desired concept labels, leaving the odds for the others unchanged. This means that low KL scores are desirable, but for ease of interpretation, we reverse them to make higher scores better.

4. RESULTS

4.1 Results Experiment 1: DecoderLens

Figure 2 shows the average recognizability of our selected instruments and genres across the Transformer layers of MusicGen-small. For instruments, we observe relatively stable recognizability in layers 0-19, followed by a gradual increase in layers 20-23. This indicates that MusicGen gradually builds up the representation of individual musical instruments across layers, with the final layers playing a crucial role. The pattern for genres is different. Except for EDM, all genres exhibit the same recognizability across layers, with a slight increase in the final layer. In contrast, EDM exhibits high scores across layers, with a gradual decline in the final layers. This pattern could be attributed to the genre distribution in MusicGen's training data: EDM is disproportionately represented [5], possibly leading to the model overfitting to EDM characteristics. It could be that most Transformer layers are tuned to generate music reminiscent of EDM, and only the final layer has learned to integrate genre information from the input text prompt.

An alternative explanation is that the DecoderLens is currently not optimized for music. Upon listening to DecoderLens outputs, we noted instances of distortion and disorganization. While these outputs predominantly resemble the EDM genre when compared to other genres like *classical* or *jazz*, they may simply reflect artifacts of the EnCodec decoder, trained specifically for decoding representations from the final Transformer layer. The representations of earlier layers may be harder to decode, as they may follow a different representational distribution.

To explore this alternative hypothesis, we examined the similarity of intermediate layer outputs within the Transformer block of MusicGen-small. We re-ran the model with the same 100 text prompts for each concept and extracted the output of each intermediate Transformer layer. We averaged these layer outputs across time and then computed the Euclidean distance between all combinations of layers. Figure 3 displays the results for *guitar* and *EDM*, but similar patterns were observed for the other instruments and genres listed in Table 1. We indeed observe that the final layer (24) is highly dissimilar to the other layers.

Further exploration, possibly involving a "translation model" that maps intermediate layer outputs to final layer outputs [24, 25], could help to refine the DecoderLens for the music domain.

4.2 Results Experiment 2: Interchange Interventions

4.2.1 *Intervention effects across model components*

Figure 4A shows the average effect of intervening on different components (MLP, self-attention, cross-attention) across the Transformer layers of MusicGen-small, for both the "replace" technique (solid lines) and the "adjust" technique (dashed lines) (results for the medium and large model can be found in the Appendix). Starting with the "replace" technique, we observe a clear contrast between the MLP and the attention components: the MLP consistently shows positive intervention effects, whereas both self-attention and cross-attention predominantly show negative effects. With the "adjust" technique, intervening on the attention components results in positive scores, but they

*For simplicity, we only use one label for each concept in this analysis, i.e., the first label listed for each category in Table 1.

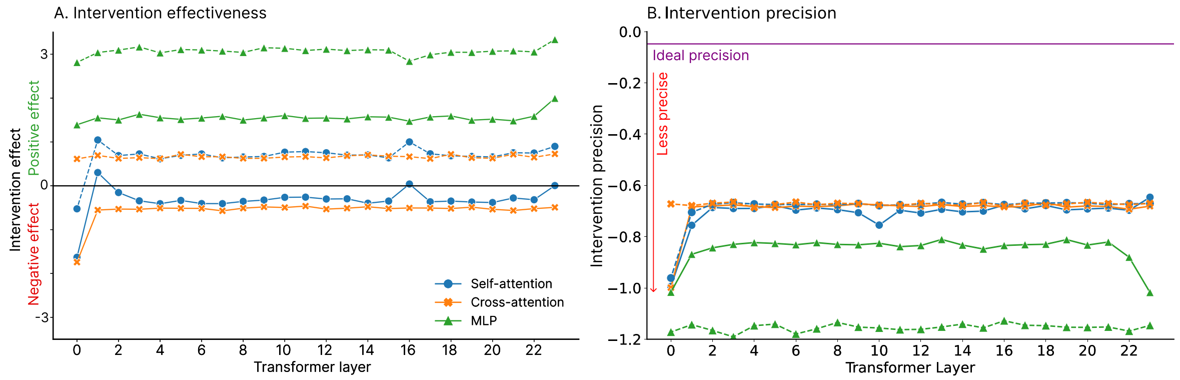

Figure 4: Results for Experiment 2 (**interchange interventions**) across model components and layers, for MusicGen-small only. Solid lines show the result for the "replace" technique, dashed lines show the results for the "adjust" technique. Figure A shows intervention effectiveness (as measured by our log odds ratio); Figure B shows intervention precision (as measured by the inversed KL-divergence). Higher scores are better for both metrics.

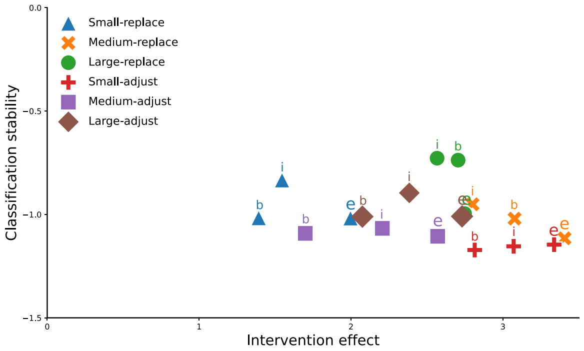

Figure 5: Intervention effect vs. intervention precision of the MLP across model sizes (small, medium, large) and intervention techniques (replace vs. adjust). The datapoints are labelled according to the layer where the intervention was performed (b = *beginning*, i = *intermediate*, e = *end*).

are still much lower than intervening on the MLP. This suggests that manipulating the sound of instruments and genres is achievable by intervening on the MLP output, but not to the same extent by intervening on the attention outputs.

Shifting focus to Figure 4B, we observe that interventions on all model components produce negative intervention precision scores This means that all interventions induce some type of alteration to the audio output. To interpret the magnitude of these changes, we compare them to the "ideal" intervention precision score, where only the original concept and desired concept probabilities flip while everything else remains unchanged. We find that the actual stability scores are much lower than this ideal scenario, suggesting that the interventions are rather invasive and change the audio in a way that goes beyond merely flipping the original concept to the desired concept. A potential future approach could be to perform the interventions on specific frames rather than on the entire audio [26].

4.2.2 *Dissecting intervention effects of the MLP*

Figure 4A suggests that interventions on the MLP yield the desired alteration (reducing the original concept and increasing the desired concept). We now analyze these effects in more detail. Figure 5 shows the relationship between intervention effectiveness and intervention precision for different model sizes (small, medium, large) and intervention techniques (replace vs. adjust) for the MLP only. For each combination of model size and intervention technique, we plot three scores: 1) the score for intervening on the **first layer**, 2) the average score for intervening on the **intermediate layers**, and 3) the score for intervening on the **final layer** (we average the intermediate layers since they showed very stable effects in Figure 4).

The effectiveness of intervention techniques seems to depend on model size. We see that the "adjust" technique performs best with the small model, while the "replace" technique shows better results with the medium and large models. We also notice that using the "replace" technique for the large model yields the highest intervention precision overall. This suggests that the large model might represent musicological concepts in a more modular manner compared to the smaller ones; thus, we can more easily modify only a single concept without affecting other concepts. One possible explanation could be that in the smaller models, due to limited space, individual neurons represent multiple features simultaneously, a phenomenon known as *superposition* [27].

Finally, for all model sizes, we observe that intervening at the final layer produces the best results, followed by intervening at intermediate layers. Intervening at the first layer tends to be least effective. This pattern may be attributed to the model's ability to "compensate" for interventions: when we intervene at early layers, the model still has plenty of opportunity to change the original or desired concept as it progresses through its forward pass.

4.2.3 *Effect on original vs. desired concept*

Figure 6 displays the intervention effect on the original and desired concept **separately**, as well as the **combined** score ($\text{score}_{\text{original}}$ + $\text{score}_{\text{desired}}$) for all three model sizes. We also separately show the effects for different intervention types: **inter-category** (instrument-

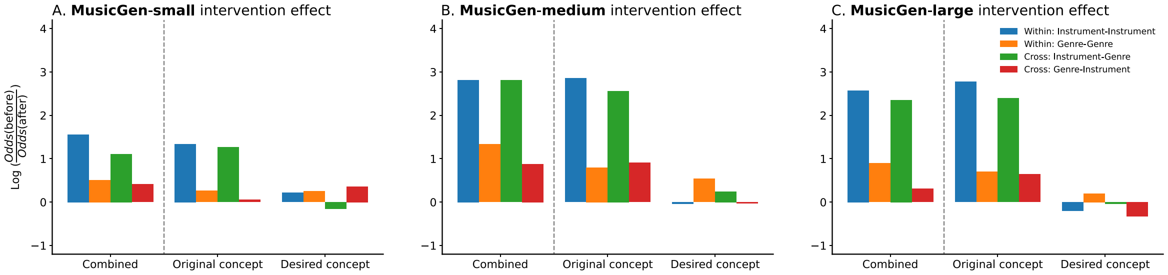

Figure 6: Combined and separated intervention effect of the **replace** technique on the original and desired concept for the MLP only, in the **small** (A), **medium** (B), and **large** (C) version of MusicGen. Bar colors indicate the intervention type (inter-category/cross-category).

instrument, genre-genre) and **cross-category** (instrument-genre, genre-instrument). Since intervention effects were similar across layers, each bar represents the average intervention effect across layers. We showcase the results for the “replace” technique here; the results for the “adjust” technique can be found in the Appendix.

When examining the scores for the original and desired concepts separately, a clear pattern emerges across all model sizes: the impact on the original concept is much bigger than on the desired concept. This suggests that interventions effectively reduce the original concept, but do not introduce the desired concept as effectively.

For the inter-category intervention effects (blue and orange bars), we observe that instrument–instrument interventions are much more effective than genre–genre interventions. This indicates that it is easier to manipulate the sound of individual instruments in the output than the sound of genres. This in turn suggests that instruments are represented in a more modular fashion than genres, which makes sense given the complex combination of features that are typically involved in a genre.

The pattern for cross-category interventions (green and red bars) is similar: instrument–genre interventions are more effective than genre–instrument interventions. Specifically, interventions inserting genre activations during a forward pass with an instrument prompt notably impact the instrument sound—but interventions inserting instrument activations during a forward pass with a genre prompt have a less pronounced effect on the genre. This supports the notion that genres are represented with multiple features, making them more resistant to manipulation compared to instruments’ more modular representation.

5. DISCUSSION

In this work, we explored the usability of text-oriented interpretability techniques for analyzing the representation of human-interpretable musicological concepts in MusicGen. We applied the DecoderLens for globally analyzing how the model conceptualizes musical instruments and genres across layers, and applied interchange interventions to dissect the role of individual layer components in generating specific instrument and genre sounds across several model sizes.

In our investigation, applying the DecoderLens to MusicGen revealed significant challenges in generating coherent audio from intermediate layers, a limitation underscored by the self-similarity matrix which showed that the last layer is vastly different from the rest of the model. Similarly, our attempts at interchange interventions, aimed at dissecting the influence of specific model components per layer on musical output, was fairly effective in removing existing musical concepts but was unsuccessful when it came to injecting new ones into the network, across all examined model sizes. These outcomes not only show the complexities inherent in interpreting generative music models but also underscore the need for music/audio specific intervention techniques.

In future work, we aim to adapt these interpretability techniques to be more suitable for audio. As for the DecoderLens, a single linear layer could be trained to map intermediate representations to final layer representations, possibly allowing for better decodability. Furthermore, we aim to explore different intervention techniques (e.g., intervening on specific frames instead of the entire sequence), which could contribute to less drastic alterations of the audio while still changing the desired concepts. These improvements may allow us to additionally explore other facets of music generation, such as tempo and rhythm.

5.1 Limitations

We used quantitative metrics based on a machine learning model to evaluate intervention effects, acknowledging that this approach introduces extra noise. However, it allowed us to investigate a larger parameter space compared to a user-listening study. For instance, we evaluated 100 prompts across multiple model components (100 * 3 components * 48 layers for the large model) for each permutation of concepts. Although human ratings from a listening study could provide many complementary insights, setting up such experiments is costly. Additionally, the authors themselves listened to several intervention results and noticed a lot of variation across samples, complicating the selection of a representative subset for a listening study. Therefore, we believe establishing robust quantitative results first is more practical. These results can inform the design of more focused and efficient listening studies, ensuring effective resource use and meaningful insights.

6. ACKNOWLEDGMENTS

We are grateful to Jamie Vlegels and Stefan Wijnja, whose research project inspired this work. This research was supported by the Dutch Research Council (NWO) as part of the project InDeep (NWA.1292.19.399).

7. REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] A. Agostinelli, T. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, "Musiclm: Generating music from text," 2023.

[3] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank *et al.*, "Noise2music: Text-conditioned music generation with diffusion models," *arXiv preprint arXiv:2302.03917*, 2023.

[4] F. Schneider, O. Kamal, Z. Jin, and B. Schölkopf, "Mousai: Text-to-music generation with long-context latent diffusion," *arXiv preprint arXiv:2301.11757*, 2023.

[5] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," 2023.

[6] M. Keller, G. Loiseau, and L. Bigo, "What musical knowledge does self-attention learn?" in *Workshop on NLP for Music and Spoken Audio (NLP4MuSA 2021)*, 2021.

[7] S. Han, H. Ihm, and W. Lim, "Systematic analysis of music representations from bert," *arXiv preprint arXiv:2306.04628*, 2023.

[8] N. Cosme-Clifford, J. Symons, K. Kapoor, and C. W. White, "Musicological interpretability in generative transformers," in *2023 4th International Symposium on the Internet of Sounds*. IEEE, 2023, pp. 1–9.

[9] V. Haunschmid, E. Manilow, and G. Widmer, "audioLIME: Listenable Explanations Using Source Separation," 13th International Workshop on Machine Learning and Music, 2020.

[10] S. Mishra, B. L. Sturm, and S. Dixon, "Local interpretable model-agnostic explanations for music content analysis." in *ISMIR*, vol. 53, 2017, pp. 537–543.

[11] F. Foscarin, K. Hoedt, V. Praher, A. Flexer, and G. Widmer, "Concept-based techniques for" musicologist-friendly" explanations in a deep music classifier," *arXiv preprint arXiv:2208.12485*, 2022.

[12] A. Langedijk, H. Mohebbi, G. Sarti, W. Zuidema, and J. Jumelet, "Decoderlens: Layerwise interpretation of encoder-decoder transformers," *arXiv preprint arXiv:2310.03686*, 2023.

[13] nostalgebraist, "Interpreting gpt: The logit lens." 2023. [Online]. Available: https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens

[14] M. Hanna, O. Liu, and A. Variengien, "How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model," *arXiv preprint arXiv:2305.00586*, 2023.

[15] T. Yamakoshi, J. McClelland, A. Goldberg, and R. Hawkins, "Causal interventions expose implicit situation models for commonsense language understanding," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 13 265–13 293. [Online]. Available: https://aclanthology.org/2023.findings-acl.839

[16] J. Vig, S. Gehrmann, Y. Belinkov, S. Qian, D. Nevo, Y. Singer, and S. Shieber, "Investigating gender bias in language models using causal mediation analysis," *Advances in neural information processing systems*, vol. 33, pp. 12 388–12 401, 2020.

[17] A. Chintam, R. Beloch, W. Zuidema, M. Hanna, and O. van der Wal, "Identifying and adapting transformer-components responsible for gender bias in an English language model," *arXiv preprint arXiv:2310.12611*, 2023.

[18] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," 2022.

[19] K. Koutini, J. Schlüter, H. Eghbal-Zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," *arXiv preprint arXiv:2110.05069*, 2021.

[20] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[21] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.

[22] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 746–751.

[23] T. Yamakoshi, J. L. McClelland, A. E. Goldberg, and R. D. Hawkins, "Causal interventions expose implicit situation models for commonsense language understanding," *arXiv preprint arXiv:2306.03882*, 2023.

[24] N. Belrose, Z. Furman, L. Smith, D. Halawi, I. Ostrovsky, L. McKinney, S. Biderman, and J. Steinhardt, "Eliciting latent predictions from transformers with the tuned lens," 2023.

[25] A. Y. Din, T. Karidi, L. Choshen, and M. Geva, "Jump to conclusions: Short-cutting transformers with linear transformations," *ArXiv*, vol. abs/2303.09435, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257557722

[26] J. Koo, G. Wichern, F. G. Germain, S. Khurana, and J. L. Roux, "Smitin: Self-monitored inference-time intervention for generative music transformers," *arXiv preprint arXiv:2404.02252*, 2024.

[27] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen *et al.*, "Toy models of superposition," *arXiv preprint arXiv:2209.10652*, 2022.