A GENERATIVE FRAMEWORK FOR COMPOSITION-AWARE LOOP RECOMMENDATION IN MUSIC PRODUCTION: DRUM2BASS USE CASE

Xiaowan YI (易小琬)[1] and **Mathieu BARTHET**[2]

[1,2]*Centre for Digital Music*, **Queen Mary University of London**, London, UK

ABSTRACT

Loops can be defined as short audio items intended to be repeated seamlessly. They have become popular in computer-based music production, particularly in genres such as electronic music and hip hop. A common practice in loop-based music production is to select several loops from a library and play them concurrently. As the producer's loop library expands, a notable challenge emerges in the loop retrieval process: choice overload becomes increasingly prevalent, as navigating and listening to thousands of audio items can be overwhelming, potentially impeding producers' creativity. To facilitate the loop retrieval process, we propose a novel generative framework for loop recommendation that takes into account the composition elements. The framework first generates a best-guess loop conditioned on the current composition elements. The generated best-guess loop then serves both as a serendipitous item for recommendation and an anchor for retrieving existing loops to recommend. We demonstrate this framework with a use case in generating and recommending bass loops based on a seed drum loop (Drum2Bass). We evaluate the retrieval performance of Drum2Bass using computational metrics and provide audio examples of generated and recommended bass loops for listening.

1. INTRODUCTION

In computer-assisted music production, loops generally refer to bar-aligned and structured audio items of short duration [1]. They are widely used in many music genres and in particular, electronic [2] and hip hop music [3]. There are emerging databases to assist composition by providing a wide range of production-ready loops such as Ampify Music [1], Splice [2] and Loopmasters [3]. With such richness of content, musicians of all levels can now access vast amount of composition materials. However, it may impede creativity as it can be daunting to navigate and listen to thousands of audio pieces just to retrieve several loops matching a producer's current composition.

To this end, we propose a generative framework for composition-aware loop recommendation that aims to facilitate the loop retrieval process during music production. We introduce the concept of *composition-aware recommendation* to describe algorithms that recommend audio content by analysing the present elements within a musical composition. The proposed framework leverages generative models for loop generation, with these generated loops feeding into a novel paradigm for loop recommendation. Given a seed loop in a producer's current composition, the framework firstly generates a loop in raw audio waveform to complement the seed loop. This generated best-guess loop, which does not exist in the producer's current loop library, is recommended to the producer as a serendipitous item. Following this, a similarity-based search is carried out across the loop library using the generated loop as an anchor. Loops within the library that are most similar to the generated best-guess loop are subsequently recommended to the producer. The loop generation capability can contribute to building an explainable recommendation framework as it enables producers to listen to and interpret the intermediate step behind the framework's recommendation process. To test this framework, we investigate the particular use case of generating and recommending bass loops given a drum loop, referred to as Drum2Bass. We evaluate Drum2Bass using both retrieval-relevancy-based and diversity-based metrics.

The remainder of this paper is structured as follows. Section 2 discusses related works and Section 3 addresses the problem context for composition-aware loop recommendation. The proposed framework is described in Section 4. Section 5 reports on the Drum2Bass use case, including the dataset, models, training process, and evaluation. Conclusions and future works are presented in Section 6.

2. RELATED WORK

2.1 Composition-aware Content Retrieval for Music Production

Composition-aware content retrieval for music production is a relatively under-explored area where previous works focused on three types of music content (one-shot drum samples [4], stems [5], and loops [6, 7]) and mainly adopted classification or joint embedding method. Classification methods may use a binary classifier to determine if retrieved music content (like a percussion stem [5]) is a good match for the given composition context (e.g. a seed vocal stem [5]). The music content recommendations are ordered according to the classification scores. Following a

[1] https://ampifymusic.com/
[2] https://www.splice.com/
[3] https://www.loopmasters.com/

joint embedding method, music content and corresponding composition context are mapped into a learned latent space where being closer according to some metrics indicates a better match. The music content recommendations are ordered by ranking the distance computed between the latent representations. SampleMatch [4] tackled one-shot drum sample retrieval problem using a joint embedding method. Neural Loop Combiner [6] worked on finding good combinations of loops extracted from hip hop music. The authors experimented with both methods and found that the classification method outperformed the joint embedding method in a subjective listening test. Works by Huang et al. [5] modelled stem retrieval for mashup creation via a classification method.

2.2 Variational Autoencoder in Music Content Generation

Variational Autoencoder (VAE) [8] and its variations, which typically comprise an encoder and a decoder, have been adopted in recent works related to music content generation [9]. The VAE family has exhibited strong representation learning capability and the latent representations produced by the encoder are utilised across a range of creative applications. RAVE [9] adopted VAE for modelling raw audio waveforms and obtained compact latent representations by post-training analysis. It demonstrated real-time audio synthesis and timbre transfer capabilities due to the compactness of the latent representation. SoundStream [10] extended VAE with residual vector quantization for producing discrete latent representations from raw audio waveforms. It was adopted in MusicLM [11], a recent text-conditioned music generation model, where the learned discrete latent representations of audio content were used as modelling targets (referred to as acoustic tokens). MusicVAE [12] proposed the use of a hierarchical decoder in the VAE architecture for modelling long-term sequence of music notes. MusicVAE was used to generate interpolations between loop notes in the symbolic domain.

3. COMPOSITION-AWARE LOOP RECOMMENDATION

3.1 Problem Definition

In the context of composition-aware loop recommendation, we introduce the following definitions:

- Loop: A loop is an audio item that can be seamlessly repeated. Loops can feature one or multiple instruments (e.g. bass loop, beat, etc.). While instrumentation varies across loop providers, an example of popular set for electronic music is : drum, bass, melodic, harmonic, vocal. In order to focus the problem, we constrain each loop to have one and only one instrument $Inst_j$.

- Loop pack: Loops are often grouped in packs, which are also referred to as sample packs. Loops within one pack typically have the same key, tempo and genre. Loop packs often share a similar aesthetic within a pack, designed by the provider to sound good when played together right out of the box.

- Loop library L: A loop library L is represented as a set of N different loops $\{l_1, l_2, ..., l_N\}$. These loops may or may not be grouped in loop packs.

- Loop-based music composition: We propose to narrow down our scope to the process of selecting and retrieving loops from a producer's loop library, where retrieved loops get played concurrently to other loops in a music section.

- Composition-aware loop recommendation: Given a loop l_m of instrumentation $Inst_j$ in the current composition, a composition-aware loop recommender aims to produce a ranked list of loops from a different instrument $Inst_{k \neq j}$ specified by the producer. Loops in the recommendation list are ranked based on their predicted degree of music compatibility with the seed loop. This is different than predicting musical similarity to a seed item, as often used in music recommendation for listening.

3.2 Challenges

We identify three main challenges for composition-aware loop recommendation. Firstly, there is a lack of user-item (producer-loop) interaction data in music production software; this data would be crucial to support collaborative filtering prevalent in traditional recommendation systems. To the best of our knowledge, there are no public datasets providing metadata on loop combinations for a varied set of producers, compositions, and genres. Consequently, we propose to adopt a content-based recommendation approach, which circumvent the requirement for user-item interaction data. Audio content-based recommendation usually relies on extracted or learned audio items representations; these are used to infer similarities between items and rank them [13]. Unlike traditional methods that search for similar items, we seek compatible items that complement the seed item. This poses a second research challenge as compatibility-based audio content retrieval is an understudied problem. Both the classification and joint embedding methods require negative data, in our case incompatible loop pairs (loops that do not sound good when being played together). With a lack of datasets providing both compatible and incompatible loops, rule-based strategies (e.g. juxtaposing incompatible keys, tempos and beat phrases in the work by Huang et al. [5] or random associations) are often required for assembling incompatible data from datasets with compatible data. It is shown in the work by Chen et al. [6] that incompatible data mining strategy itself is a system design variable. In their study, the classification method and the joint embedding method exhibited different performances across various data mining strategies, and no optimal strategy was identified. A question remains whether it is possible to develop a loop recommendation framework that can differentiate loops based on compatibility without relying on pre-defined incompatible loop pairs. Lastly, a loop recommendation framework supports a creative process where properties such as user control over recommendation, serendipity and diversity are sometimes more desirable than metrical accuracy and precision [14]. This poses challenges for the design of the recommendation framework as well as the evaluation methods.

4. PROPOSED FRAMEWORK

We decompose composition-aware loop recommendation into two subproblems: generative modelling and loop pair compatibility modelling. To this end, we propose a generative framework that comprises two training phases with each phase addressing one subproblem.

4.1 Two Training Phases

4.1.1 *Phase 1: generative modelling*

The first phase is a generative modelling phase where latent-variable generative models are trained to reconstruct loops in raw audio waveforms. There is one generative model trained for each instrument loop category (e.g. one model for drum loops and another model for bass loops), as shown in Fig 1. The artificial neural network architecture we use is the Variational Autoencoder (VAE). The goal of the first phase is to learn compact representations, referred to as embeddings, for each instrument loop and to ensure that such embeddings are capable of generation, i.e. having enough information for the instrument-specific decoder to generate raw audio waveforms. It is worth noting that the data requirement for the first phase is just a loop dataset that has instrument labels for each loop.

4.1.2 *Phase 2: mapping compatible loop pair embeddings*

Once the first phase of training is done, both instrument loop VAE models are frozen. The second phase is where we try to directly model the mapping relation between compatible loop pair's embeddings, as shown in Fig 1. To this end, we train an extra module to map the different instrument loop embeddings to a shared latent space where proximity indicates compatibility. Compatibility-based recommendation is then reduced to searching for loops that are close to each other in the shared latent space. One example of such module is to train a Multilayer Perceptron (MLP) to, for instance, predict the compatible bass loop embedding given a drum loop embedding. Because the training objective is to directly predict the target compatible loop embedding which is neither a classification score nor contrastive loss, it bypasses the need for incompatible loop pairs mining. Hence, the data requirement for the second phase is a loop dataset that has compatible loop pairs (loops that sound good when played concurrently). Due to the lack of publicly available compatible loop pair datasets, we propose two strategies for creating such datasets. One way is to extract pairs from loop library that has loop-pack structure. Loops of different instruments can potentially be considered compatible if they belong to the same loop pack and share the same

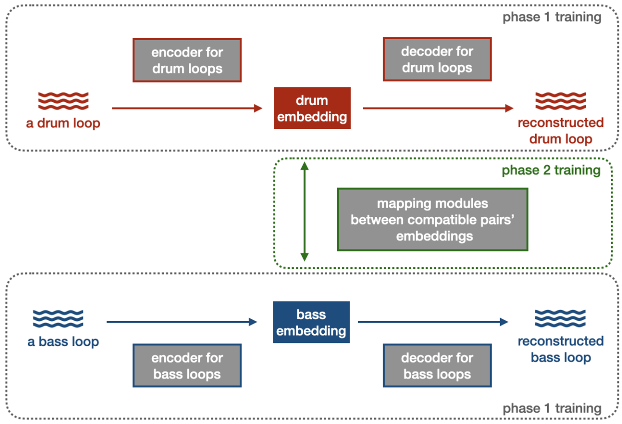

Figure 1. The proposed framework's two training phases. Phase 1: two VAEs are separately trained for each instrument. Phase 2: a mapping module is trained to predict compatible loop's embedding.

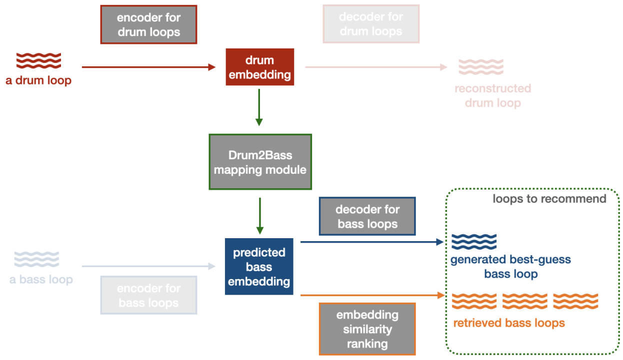

Figure 2. The framework deployment with an example of recommending compatible bass loops from a drum loop through the Drum2Bass mapping module.

key, tempo and genre. This strategy is used in our experiment. Another way is to extract pairs from multitrack music datasets. We can obtain loops by running songs through a loop extraction algorithm [15]. Pairs of loops that originate from the same song section are assumed to be compatible. Neural Loop Combiner [6] adopted this data extraction strategy.

4.2 Model Deployment

During deployment, all loop embeddings, which are encoded by the corresponding instrument-specific encoders, can be pre-calculated and stored. Given a seed loop in the composition and a target instrument, the framework retrieves the loop's stored embedding which is then passed into the corresponding latent mapping module that outputs the predicted compatible loop's embedding. The predicted compatible loop embedding can be fed into the decoder for generating a best-guess loop. It can also be used as a recommendation query for retrieving similar loops from the loop library. An example pipeline is shown in Fig 2. It is worth noting that the proposed framework affords several other applications that are specific to music creation. The trained instrument-specific VAE models in phase 1 enable timbre transfer and latent interpolation of loops. Timbre transfer can be done by encoding and decoding a loop from instrument $Inst_k$ using the VAE model trained with loops of $Inst_{j\neq k}$, which is similar to the RAVE application demonstration. Latent interpolation can be done by interpolating and decoding the embeddings of two loops of the same instrument. This method creatively combines the essence of two loops to generate new ones, similar to how MusicVAE works.

5. EXPERIMENTS: DRUM2BASS USE CASE

In this work, we address the specific case of recommending a bass loop based on a seed drum loop, referred to as Drum2Bass.

5.1 Dataset

We use a proprietary techno and house music loop library of 44.1 kHz sample rate from Ampify Music. It has 117 loop packs comprised of 846 drum loops and 613 bass loops. Each pack has at least one drum loop and one bass loop, and some packs contain more drum loops than bass loops. We extract 846 compatible loop pairs by matching each drum loop with one bass loop from the same pack. We then split the dataset into training/validation/test sets using the ratios 0.85/0.05/0.1. The split is done at the loop pack level so that loops from the same pack are not found in different sets.

5.2 Models

For the first phase, we choose RAVE [9] as our VAE model implementation. We separately train one RAVE model on drum loops (*DrumRAVE*) and another one on bass loops (*BassRAVE*). The training objective for each model comprises a reconstruction loss term and a prior regularisation loss term:

$$L_{RAVE} = E_{\hat{x}\sim p(x|z)}[S(x,\hat{x})] + \beta * D_{KL}[q_\phi(z|x)||p(z)], \quad (1)$$

where x is a loop from the dataset in the audio waveform, z is the latent variable with a prior distribution $p(z)$. $q_\phi(z|x)$ is the posterior distribution parameterised by the RAVE encoder and $p(x|z)$ is the conditional likelihood distribution parameterised by the decoder. $S(x,\hat{x})$ denotes the multi-scale spectral distance between the original and reconstructed waveform. D_{KL} denotes the Kullback-Leibler divergence. After training, both models' encoders and decoders are frozen before proceeding to the second phase.

For the latent mapping module in the second phase, we start with an MLP of four fully connected layers that maps drum embeddings to compatible bass embeddings. The training objective for the MLP is the L2 distance in the latent space, calculated between the predicted and the groundtruth compatible bass loop embeddings:

$$L_{latent} = ||\hat{y} - BassRAVE_{enc}(y)||_2, \quad (2)$$

with

$$\hat{y} = MLP(DrumRAVE_{enc}(x)), \quad (3)$$

where y is the groundtruth compatible bass loop (in the audio domain) for seed drum loop x, and $\hat{y}$ is the predicted compatible bass loop embedding. $BassRAVE_{enc}$ and $DrumRAVE_{enc}$ denote the RAVE encoders respectively. We also experiment with the L2 distance computed in the decoded space, which is calculated after running embeddings through the frozen BassRAVE decoder:

$$L_{dec} = ||BassRAVE_{dec}(\hat{y}) - y_{dec}||_2, \quad (4)$$

where

$$y_{dec} = BassRAVE_{dec}(BassRAVE_{enc}(y)). \quad (5)$$

We denote the MLP variant with loss computed in the latent space (i.e. without being decoded) by *MLP-Latent* and the MLP variant with loss computed in the decoded space by *MLP-Dec*. Our hypothesis is that *MLP-Latent* will yield better compatible loop retrieval performance as the embedding similarity ranking deploys distance measured in the latent space instead of the decoded space. *MLP-Dec* may produce better loop generation quality as it optimizes towards raw audio waveform distance.

5.3 Training

For training the RAVE models in phase 1, audio segments of a fixed length of 65536 sample points are randomly selected from each loop with zero-padding. We train DrumRAVE and BassRAVE each for $3,000,000$ training steps with batch size 8 and start with a learning rate 0.0001 using ADAM optimizer. For training the MLP in phase 2, we extract a 4-second window from each loop pair to compute embeddings with tiling where applicable. We choose to keep loops at their original tempo so no time-stretching is applied. As the slowest tempo from the loop library is 60 BPM, 4-second window ensures there is at least 4 beats of content taken into account. The input and output size of the MLP, which is the RAVE embedding size, is $(128, 87)$ for each 4-second loop where 128 is the feature resolution and 87 is the temporal resolution. We train *MLP-Latent* and *MLP-Dec* each for $50,000$ training steps with batch size 32 and start with a learning rate 0.0001 using ADAM optimizer.

5.4 Results and Discussion

The evaluation of the Drum2Bass framework is divided into two parts: one focuses on loop generation performance, and the other assesses loop recommendation performance with computational evaluation detailed in Section 5.4.1. For loop generation performance, we invite readers to listen to the audio examples here[4]. Overall, the generated bass loops adhere to the seed drum loop's tempo without being explicitly conditioned on tempo information. The generated loops exhibit diversity across different seed drum loops, and there are interesting rhythmic patterns generated such as syncopation. The generation quality still needs improvement, though we have not performed extensive hyper-parameter tuning in this experiment.

[4] https://bit.ly/generative-loop-recommender-audio-examples

Variant	R_{mn}	R_{md}	ILD@5	ILD@10
MLP-Latent	**0.394**	**0.306**	0.623	0.640
MLP-Dec	0.524	0.473	0.626	0.638

Table 1. Computational Evaluation Results

5.4.1 Computational Evaluation

Here we report the evaluation results regarding the loop recommendation performance, with computational metrics shown in Table 1. Given a seed drum loop i, we create a recommendation list of bass loops by ranking them according to the cosine similarities between their embeddings and the generated best-guess bass embedding. We then assess the quality of the recommendation list using both rank-based metrics and a diversity-based metric. Specifically, we propose to use Mean Normalized Rank R_{mn} and Median Normalized Rank R_{md} as rank-based metrics:

$$R_{mn} = \frac{1}{N_{drum}} \sum_{i=0}^{N_{drum}} \frac{rank_i}{N_{bass}}, \tag{6}$$

$$R_{md} = median\{\frac{rank_i}{N_{bass}} \,|i = 1, 2, ..., N_{drum}\}, \tag{7}$$

where N_{drum} and N_{bass} denote the number of drum loops and bass loops, respectively. We use $rank_i$ to denote the rank of the groundtruth compatible bass loop in the ranked recommendation list for the seed drum loop i. Dividing $rank_i$ by N_{bass} normalizes it into $(0, 1]$. R_{mn} represents the mean of all normalized ranks, while R_{md} represents the median, effectively ignoring potential outliers. The closer the normalized rank is to 0, the better the framework performs in ranking the most compatible bass loop higher in the recommendation list among other bass loops, thereby alleviating choice overload for music producers. These rank-based metrics were also adopted in related works, including SampleMatch [4] and Neural Loop Combiner [6], for the same reason. In response to the challenge mentioned in Section 3.2 regarding the assessment of loop recommendation frameworks within the context of creative applications, we introduce a diversity-based evaluation. We use an inter-list diversity metric (ILD) which considers the uniqueness of different drum loops' bass loop recommendation lists:

$$ILD@L = \frac{1}{\binom{N_{drum}}{2}} \sum_{i \neq j}^{N_{drum}} 1 - \frac{q_{ij}(L)}{L}, \tag{8}$$

where $q_{ij}(L)$ is the number of common bass loops in the top L places of any two drum loops' recommendation lists. $\binom{N_{drum}}{2}$ is the normalization term denoting the total number of combinations when choosing two drum loops from N_{drum} loops. A higher value of $ILD@L$ generally indicates greater diversity among the top L bass loop recommendations for any two drum loops.

As can be seen in Table 1, the best rank-based evaluation result is achieved by the variant *MLP-Latent*. It outperforms the random sampling baseline (which has a normalized rank of 0.5) and *MLP-Dec* in both R_{mn} and R_{md} which is in line with our initial hypothesis. We conjecture that this is because the loss measured in the decoded space may not directly agree with the loss measured in the latent space. The final recommendation item ranking is solely based on the distance measured in the latent space which favors *MLP-Latent*. The results on $ILD@5$ and $ILD@10$ indicate that *MLP-Latent* and *MLP-Dec* exhibit similar diversity-based recommendation performance. On average, more than half of the bass loops in one seed drum loop's recommendation list differ from those in any another seed loop's recommendation list. Notably, $ILD@5$ values of 0.623 (*MLP-Latent*) and 0.626 (*MLP-Dec*) imply that, on average, more than three out of the top five recommended loops are different between any two recommendation lists.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we discussed the challenges of composition-aware loop recommendation and proposed a novel two-phase generative framework to address these challenges. We demonstrated the framework through the Drum2Bass use case, which generates and recommends bass loops based on a seed drum loop. Our experiments with Drum2Bass highlighted the framework's capabilities in loop generation, loop compatibility modeling without relying on negative (incompatible) data mining, and loop recommendation with inter-list diversity. We investigated two phase-2 loss variants in Drum2Bass, and the computational evaluation results showed that the loss computed in the latent space achieved better retrieval performance than the loss computed in the decoded space. Both variants had high inter-list recommendation diversity. However, these two computational metrics might be partial in evaluating a recommendation framework for music production. We will extend the evaluation with a user study in the future to better reflect the system's efficacy during actual music production. As there is still some headroom for improvement in loop generation quality and retrieval performance, we will refine Drum2Bass by experimenting with other state-of-the-art techniques including adding latent code quantization for phase 1 training and adopting models with greater expressiveness (e.g. a decoder-only transformer architecture) for phase 2 training. Additionally, we want to extend our experiment in the future to incorporate public dataset such as the Freesound Loop Dataset [16] and establish benchmarks for the composition-aware loop recommendation challenge.

Acknowledgments

This work is funded by UK Research and Innovation [grant number EP/S022694/1] and Focusrite. Our special thanks are extended to Focusrite Ampify Music team for their technical and dataset support.

7. REFERENCES

[1] G. Stillar, "Loops as genre resources," *Folia Linguistica*, vol. 39, no. 1-2, pp. 197–212, 2005.

[Online]. Available: https://doi.org/10.1515/flin.2005.39.1-2.197

[2] M. J. Butler, *Unlocking the groove: Rhythm, meter, and musical design in electronic dance music*. Indiana University Press, 2006.

[3] D. Charnas, *Dilla time: The life and afterlife of J Dilla, the hip-hop producer who reinvented rhythm*. MCD, 2022.

[4] S. Lattner, "Samplematch: Drum sample retrieval by musical context," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, pp. 781–788. [Online]. Available: https://archives.ismir.net/ismir2022/paper/000094.pdf

[5] J. Huang, J. Wang, J. B. L. Smith, X. Song, and Y. Wang, "Modeling the compatibility of stem tracks to generate music mashups," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 187–195. [Online]. Available: https://doi.org/10.1609/aaai.v35i1.16092

[6] B. Chen, J. B. L. Smith, and Y. Yang, "Neural loop combiner: Neural network models for assessing the compatibility of loops," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, pp. 424–431. [Online]. Available: http://archives.ismir.net/ismir2020/paper/000225.pdf

[7] F. Bruford, O. Lartillot, S. McDonald, and M. B. Sandler, "Multidimensional similarity modelling of complex drum loops using the groovetoolbox," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, 2020, pp. 263–270. [Online]. Available: http://archives.ismir.net/ismir2020/paper/000211.pdf

[8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6114

[9] A. Caillon and P. Esling, "Rave: A variational autoencoder for fast and high-quality neural audio synthesis," *CoRR*, vol. abs/2111.05011, 2021. [Online]. Available: https://arxiv.org/abs/2111.05011

[10] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021. [Online]. Available: https://doi.org/10.1109/TASLP.2021.3129994

[11] A. Agostinelli, T. I. Denk, Z. Borsos, J. H. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. H. Frank, "Musiclm: Generating music from text," *CoRR*, vol. abs/2301.11325, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2301.11325

[12] A. Roberts, J. H. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 4361–4370. [Online]. Available: http://proceedings.mlr.press/v80/roberts18a.html

[13] M. Schedl, H. Zamani, C.-W. Chen, Y. Deldjoo, and M. Elahi, "Current challenges and visions in music recommender systems research," *International Journal of Multimedia Information Retrieval*, vol. 7, no. 2, pp. 95–116, 2018. [Online]. Available: https://doi.org/10.1007/s13735-018-0154-2

[14] M. Schedl, P. Knees, and F. Gouyon, "New paths in music recommender systems research," in *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017*. ACM, 2017, pp. 392–393. [Online]. Available: https://doi.org/10.1145/3109859.3109934

[15] J. B. Smith and M. Goto, "Nonnegative tensor factorization for source separation of loops in audio," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. IEEE, 2018, pp. 171–175. [Online]. Available: https://doi.org/10.1109/ICASSP.2018.8461876

[16] A. Ramires, F. Font, D. Bogdanov, J. B. L. Smith, Y. Yang, J. Ching, B. Chen, Y. Wu, W. Hsu, and X. Serra, "The freesound loop dataset and annotation tool," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, 2020, pp. 287–294. [Online]. Available: http://archives.ismir.net/ismir2020/paper/000241.pdf