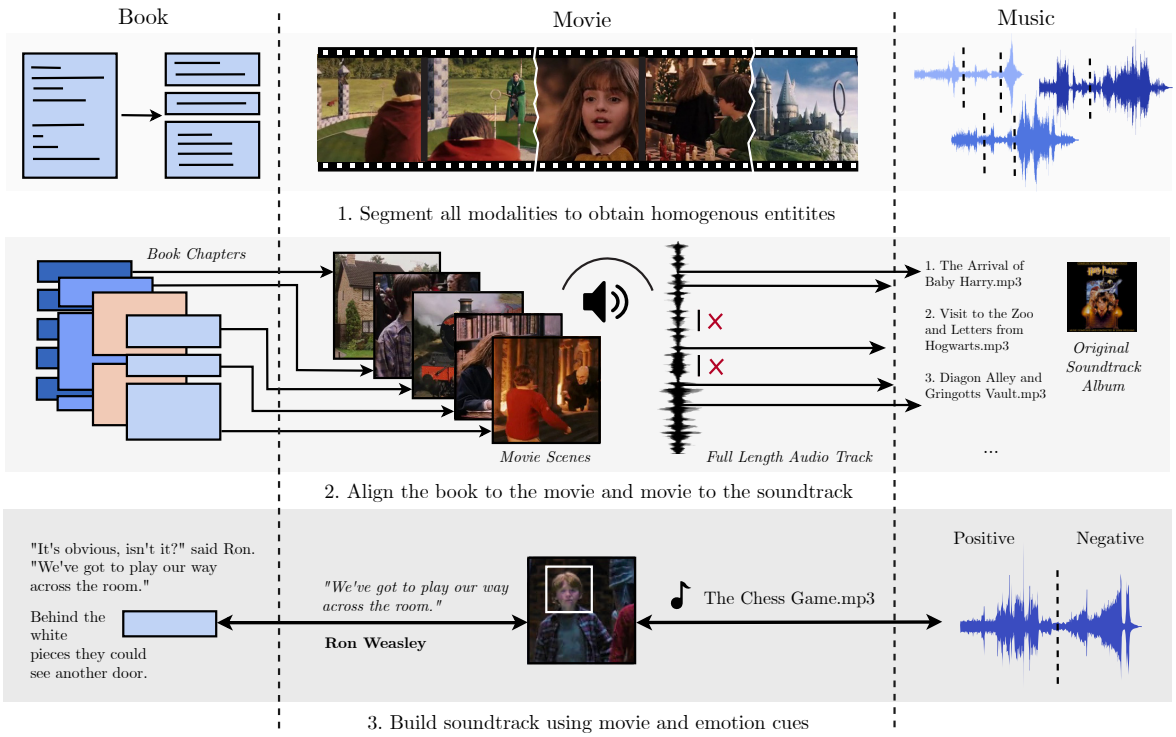

Figure 2. **Overview of approach.** We present a novel technique to build a musically rich book-length soundtrack. We accomplish this by first segmenting the book, its adaptation, and music into smaller homogenous chunks. These segments are then matched with the movie acting as an intermediary for text and music, thereby producing the final soundtrack.

emotions, *keystrength* [21] is a good attribute as it captures the tonal properties of music, and tonal changes likely suggest emotional shifts (*e.g.* major to minor mode suggesting a potential shift from positive to negative valence [22]). *Keystrength* as extracted via the MIRToolbox [23] gives us a probability for each possible key resulting in a 24 dimensional vector (12 major and 12 minor keys). We use a window size of 10 seconds with an 85% overlap, that helps ignore minor local variations in the track while retaining larger shifts. We then compute a self-similarity matrix [24] of the time-varying keystrength vector which is then used to calculate the novelty curve, with a kernel size of 64 [24]. The peaks in the novelty curve determine points of change. Finally, we use those peaks as our segment boundaries to produce L_j music segments $\{M_j^c\}_{c=1}^{L_j}$ (see Fig. 3).

Movie scene detection. As a third step, we segment the movie into narrative-coherent scenes $[V_1, \ldots, V_Q]$ using the approach described in [25]. First, the movie is divided into shots (consecutive video frames from the same camera). Then, a dynamic programming algorithm is used to find scene boundaries (a scene consists of multiple shots) so as to maximise intra-scene similarity.

Summarising, we denote $\mathcal{B}_i^p$ as the segment of book chapter $\mathcal{B}_i$; M_j^c as the music segment from track M_j; and V_q as the q^{th} scene from the movie $\mathcal{V}$.

3.2 Aligning the Book, Movie, and Music

We first align the book with the movie adaptation using a new two-stage coarse-to-fine alignment scheme. Then, we also align the movie audio to the soundtrack album.

Chapter-scene coarse alignment. We first assign a set of scenes from the movie to each book chapter. We use an approach similar to [26] where pairwise similarities are computed between a book chapter $\mathcal{B}_i$ and a video scene V_q via character histograms and matched dialogues. The chapter-scene relationship is then encoded as a graph (each node represents a chapter-scene pair), with edge weights representing similarity scores. Calculating the shortest path over this graph provides the alignment between all chapters and scenes. Additional details are provided in the Appendix.

Paragraph-scene refinement. The coarse alignment cannot be used directly for soundtracking as a chapter $\mathcal{B}_i$ contains distinct segments $\mathcal{B}_i^p$ that likely need different music. Thus, in addition to sparse dialogue matches, we compute similarities between sentences in the chapter segment $\mathcal{B}_i^p$ and frames of the video scene V_q using a pretrained vision-language model (CLIP [27]). To improve the quality of CLIP matching scores, we prune dialog and mundane sentences from the chapter segment using a TF·IDF based scoring system. This emphasises relatively rare characters and objects that are likely to give stronger matches than commonly occurring ones. We also retain sentences with a high *concreteness* index [28] that measures how likely a word can be seen or experienced (in contrast to *abstract* words). Finally, we take the top remaining sentences, encode them with CLIP, and calculate cosine similarity with all CLIP encoded video shot frames in the chapter's scenes (see Fig. 4). Scenes with a score higher than θ are assigned to the text segment using a mapping function,

$$A(\mathcal{B}_i^p) = \{V_q : \text{CLIP}(\mathcal{B}_i^p, V_q) > \theta\}\,. \tag{2}$$

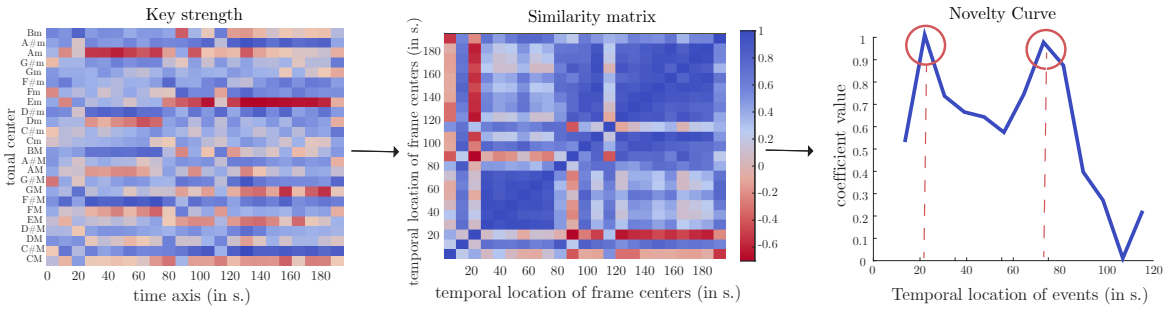

Figure 3. **Music segmentation pipeline.** We segment all tracks from the soundtrack to ensure a cohesive listening experience. We extract keystrength [21] that captures tonal properties of a soundtrack (left), compute the self-similarity matrix (center), and use that to calculate the novelty curve [24]. The peaks of this curve are used to segment the track.

Aligning the movie audio track with the soundtrack album. We identify the music in the movie by passing its audio track through Shazam [1] , a popular commercial audio-fingerprint based search application [29] using its free public API. This results in high-precision estimates for the music played every few seconds. While any audio fingerprinting approach would perhaps work, we found that Shazam was quite accurate in the presence of background noise and dialogue, which is pervasive in movies.

3.3 Weaving the Book Soundtrack

Post alignment with the movie, book chapter segments $\mathcal{B}_i^p$ belong to one of two sets: (i) those associated with at least one movie scene $\mathcal{S} = \{\mathcal{B}_i^p : |A(\mathcal{B}_i^p)| \geq 1\}$; and (ii) those without, $\bar{\mathcal{S}} = \{\mathcal{B}_i^p : A(\mathcal{B}_i^p) = \varnothing\}$. Recall, $A(\mathcal{B}_i^p)$ is the set of aligned video scenes to a book segment (*cf*. Eq. 2).

Extracting emotion labels. For all text segments $\mathcal{B}_i^p$, we classify each paragraph into positive, neutral, or negative using a BERT-based emotion classifier ϕ_{BERT}, trained on Reddit comments [30]. A majority vote across paragraphs is used to assign the emotion label to the chapter segment $E_{\text{book}}(\mathcal{B}_i^p) = \text{mode}(\phi_{\text{BERT}}(\mathcal{B}_i^p))$. For a music segment, similar to [5], we encode it's emotion as valence, $E_{\text{music}}(M_j^c)$, based on the mode of the song (major or minor). This is based on literature that indicates that tracks in minor tend to be associated with negative emotions and tracks in major with positive emotions [22]. We also tried approaches that predict emotion from audio [11, 31], but they didn't work as well for our application.

Importing music snippets from the video scene. For every text segment $\mathcal{B}_i^p \in \mathcal{S}$, we extract the movie timestamps corresponding to the matched dialogues or CLIP-based frame-sentence pairs. We use the audio-search to identify the overall track M_j being played at any of the above timestamps in the movie (in a small neighbourhood). A specific music segment M_j^c is chosen by matching emotion predictions *i.e.* $E_{\text{book}}(\mathcal{B}_i^p) = E_{\text{music}}(M_j^c)$ (we pick one randomly if there are several segments). While we can pick emotion-matching music segments without the book-movie alignment, it will likely result in spurious matches.

Emotion-based retrieval. For the chapter segments that are not aligned with any video scene, $\bar{\mathcal{S}}$, and those in $\mathcal{S}$ that did not find an emotion compatible soundtrack, we assign a random music segment among the set of emotionally compatible compositions. Note that while we can pick any music segment (even from different movies), we restrict to the soundtracks for this movie maintaining the composer's stylistic coherence.

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

4. RESULTS AND EVALUATION

Our approach is designed to be applicable to books that have mostly faithful movie adaptations in terms of few matching dialogues, a relatively similar plot, and at least some matching characters. We evaluate it on the first book and movie pair in the *Harry Potter* series, *Harry Potter and the Philosopher's Stone*.

4.1 Harry Potter: A Case Study

Our unsupervised text segmentation approach splits 17 chapters into 87 segments, with an average of 5.11 segments per chapter. Assuming a reading speed of 250 words per minute, each segment requires a ~4 minute track, for a total soundtrack duration of ~6 hours. Music segmentation

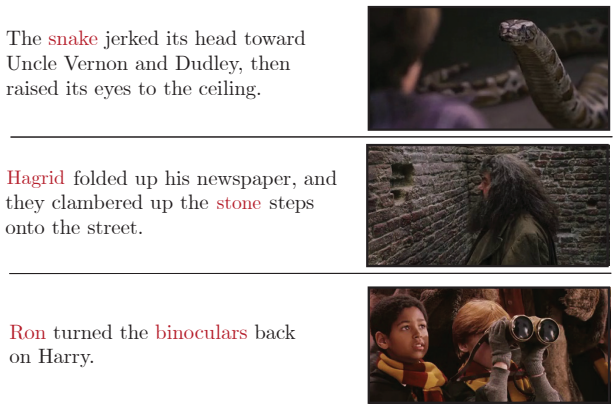

Figure 4. **CLIP-based paragraph-scene alignment.** Aligned examples of automatically selected visual sentences and their video frames in the scene using the vision-language model CLIP [27] in a zero-shot setting. Text highlights are words that we think that caused the match.

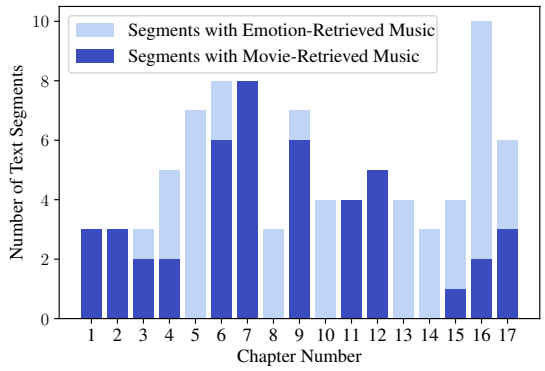

Figure 5. **Number of chapter segments with movie- vs. emotion-retrieved music.** Several chapters in the book are well represented in the movie and are predominantly soundtracked using movie cues. For others, we use an emotion-based retrieval method to build a soundtrack.

of 19 tracks from the soundtrack album results in 47 audio chunks with an average duration of ~52 seconds. For the movie, we obtain 120 scenes using the scene detection algorithm. Computing the book-movie alignment results in a strong chapter-scene alignment, with 82% accuracy, calculated as the percentage of movie shots correctly assigned to a chapter [26].

For the fine-grained alignment, 34 chapter segments have dialogue matches, while the CLIP based refinement results in a 130% improvement in the number of chapter segments associated with the movie; only 9 segments remain unmatched. After performing the final book-music alignment, 45 chapter segments have music imported from a matching video scene through the book-movie alignment. The remainder 42 segments are soundtracked using music segments fetched via emotion matching. See Fig. 5 for a chapter-wise distribution.

4.2 Experiment Design

As the primary goal of our method is to improve the reading experience, we obtain user feedback by setting up semi-structured interviews with 10 individuals, who each read two chapters of the *Harry Potter* book. We randomly picked one chapter for each person while the final book chapter was read by all. We chose the final chapter as it represents all aspects of our method: it includes music retrieved via movie cues as well as pure emotion based retrieval, apart from being a key chapter in terms of the plot. This allows us to fairly evaluate the effectiveness of our soundtrack at a book level as reading these chapters takes around an hour for each participant.

All participants are asked to read these chapters consecutively and answer a series of questions about several aspects of the reading experience. No user data is collected through this process and user consent is collected prior to recording the interviews. The study was also authorised by the author's institutional ethics committee.

Reading application. A bottleneck with playlist based approaches for soundtracking books is that they require user-input to transition at appropriate instants. We resolve this by creating an application that plays music in the background of the displayed text. Our application loops the music segment infinitely and cross-fades to the next as the reader moves on to the next text segment. This ensures complete immersiveness and lets participants with *varying* reading speeds truly enjoy the soundtrack. For readers to have a first-hand experience, we package this application for a few chapters at the project page.

Participant information. A total of 10 individuals (18-22 age range, 6 male, 4 female) volunteered for the study. Each participant had previously read the *Harry Potter* book without music and happened to be familiar with the movie as well. We also provided each participant with a small monetary reward as compensation for their time.

4.3 Findings

The soundtrack improved immersiveness. All participants reported that the soundtrack improved the immersiveness of the reading experience. This is remarkable as all participants noted that they typically do not listen to music while reading. Some participants were more receptive to the soundtrack than others and spoke glowingly about the movie-like immersion afforded by the music. These participants could recognise that the music played was similar to the score at the relevant movie scene. Others, who were positive but less movie-inclined, focused more on how the music *"set the environment"* (P4). Phrases such as *"set the mood"*, *"intensified the emotion"*, *"fit the vibe"*, and *"enhanced the experience"* were common among such participants.

> *"The music provided insight into the tone of the chapter and [...] beyond imagination, provided a soundtrack to what was read"* (P1)

Surprisingly, some participants reported that the first section of the random chapter that they read first threw them off initially, suggesting that there is an adjustment period to this experience. These participants reported that they were very comfortable with later sections (*"after the first 10-12 lines"* - P2) once they had gotten into a reading flow.

The soundtrack helped visualise the book. Some participants could recognise that the music played was similar to the score at the relevant movie scene. These participants were especially appreciative of the soundtrack and stated that it helped *"visualise the book, with movie like visuals"* (P1). Such participants typically appreciated the music's cohesiveness and were likely referring to the pleasing soundscape laid out by our soundtrack. One participant explicitly pointed out that the soundtrack helped visualise character interactions and that without, it would have just been *"two characters talking"*.

Many were also receptive to the recurring motifs and themes that appeared throughout, such as the central *Harry Potter* theme and stated that it helped imagine the fictional

world. Only one participant, P3, expressed complete disagreement with the music played in a text segment, for ironically the same reason, stating that the signature *Harry Potter* motif distracted them. Barring this, the same person spoke warmly about the remaining soundtrack, suggesting that the overtness of the motif, which permeates culture today, may be subject to individualistic preferences.

Music helped focus.

> *"I get distracted when reading so it helped me focus on certain parts"* (P5)

When describing the immersiveness of the reading experience, three participants specifically pointed out that the music helped them read continuously, without distractions that are typically present when reading in the absence of a soundtrack. It should be noted that few participants who described an aversion for listening to music during typical recreational reading specifically pointed out that they avoid pop music, suggesting that instrumental music, in general, may be better suited for reading purposes.

Moderate repetition is not a concern. Most participants noticed repetitive music in some text segments that they read but only brought it up when explicitly asked. One participant suggested playing a different, similarly composed track instead of repeating the music, but in agreement with the rest of the pool, felt that repetition did not affect the experience. These participants also said that the repetition wasn't glaringly obvious and that it did not distract or take away from the immersion. It should be noted that repetition refers to the same track being played on a cross-fade loop while a person reads a text segment, due to variable reading speeds. Despite this, it is noteworthy that our musical segments are homogeneous enough to be played repeatedly and that the text segments are narratively cohesive to warrant such repetition. When specifically asked about the diversity of the music played, all participants expressed positive opinions and noted that the tracks kept changing as and when required.

Narrative transitions mirrored in music. Many participants engaged in a conversation with the authors post-interview about how the soundtrack was built. All such participants were startled by the fact that the entire process was completely automatic. Their surprise was primarily due to the fact that the music was automatically matched with relevant areas in text and that it transitioned at appropriate narrative points.

> *"It actually made the experience better as the transition put you in the mood for the expected emotion - from melancholy to sad."* (P9)

When asked specifically about these narrative transitions during the interview, all participants agreed that it was emblematic of emotional or narrative shifts. Since the last chapter was common across all participants, there was strong consensus about the narrative shifts here especially, with participants noting that the music increased in tension as the final hero-villain clash developed and that it eased into more mellow, tender music once it was resolved.

> *"When the tension built in the plot, the music transitioned to match it."* (P1)

Some participants explicitly appreciated the foreshadowing made possible by the music, as suspenseful music at the start of a segment would precede a similar narrative plotline. Others simply elaborated on their earlier comments about the immersiveness and re-emphasised the high congruency between text segments and the matched music.

Low-arousal music preferred. We asked participants to describe the emotional suitability of the music, specifically, and in line with prior answers, and received a favourable response. The few critiques that were present revolved around the energy/arousal of the music played at certain instances. Some participants were dissatisfied with segments that were high in arousal, even though the valence matched, as it broke the immersion. They described how this music distracted them from reading and drew too much attention to the track itself. This is likely an important factor to consider for future work on soundtracking books.

Movie-based music cues stronger than emotion-based cues. At the end of the interview, participants were asked to describe their favourite and least favoured pieces of music from the two chapters, if any. The best segments were split between music retrieved via movie cues and those retrieved via emotion matching, with the former being more prominent. On the other hand, the least favoured segments, often mentioned only after the author's insistence, were typically emotion-based matches that were described in terms of their neutrality. This finding suggests that our approach can effectively retrieve narrative-specific music and that such a system is perhaps well-suited for book soundtracking, as opposed to pure emotion-based methods.

5. CONCLUSION

We presented a novel system to automatically weave a book-length soundtrack, using the music present in the relevant movie adaptation. Perceptual validation of the constructed soundtrack for the first book and movie pair of the *Harry Potter* series provided very positive feedback that validated several proposed design decisions and techniques for text, movie, and music processing. Participants in our perceptual study were particularly receptive to music that was fetched via the movie and uniformly stated that it improved the immersiveness of the reading experience, and even transported them to the fictional world.

Future work. While our method has been successful in generating a soundtrack for a book with a movie adaptation, it needs to be modified to make it work on books without adaptations. Future work can potentially use our approach to investigate common trends across books to determine new cross-narrative rules. Our approach can also be extended for human-in-the-loop collaborative soundtrack construction applications, though such a use case is beyond the scope of this work.

6. ACKNOWLEDGMENTS

We thank Siddarth Baasri for his valuable input in analysing motifs present in the *Harry Potter* soundtrack and Saravanan Senthil for illustrating Fig. 1.

7. REFERENCES

[1] M. G. Boltz, "Musical soundtracks as a schematic influence on the cognitive processing of filmed events," *Music Perception*, vol. 18, pp. 427–454, 2001. 1

[2] M. C. Pennington and R. P. Waxler, *Why reading books still matters: The Power of Literature in Digital Times*. Routledge, 2017. 1

[3] M. Chełkowska-Zacharewicz and M. Paliga, "Music emotions and associations in film music listening: The example of leitmotifs from The Lord of the Rings movies," *Annals of Psychology (Polish)*, 2019. 1

[4] J. Salas, "Generating music from literature using topic extraction and sentiment analysis," *IEEE Potentials*, vol. 37, pp. 15–18, 2018. 2

[5] H. Davis and S. M. Mohammad, "Generating music from literature," in *Workshop on Computational Linguistics for Literature @ European Association of Computational Linguistics (CLfL@EACL)*, 2014. 2, 4

[6] S. Harmon, "Narrative-inspired generation of ambient music," in *International Conference on Computational Creativity (ICCC)*, 2017. 2

[7] M. Thorogood and P. Pasquier, "Computationally created soundscapes with audio metaphor," in *International Conference on Computational Creativity (ICCC)*, 2013. 2

[8] A.-M. Oncescu, A. S. Koepke, J. F. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries," in *Interspeech*, 2021. 2

[9] M. Won, S. Oramas, O. Nieto, F. Gouyon, and X. Serra, "Multimodal Metric Learning for Tag-Based Music Retrieval," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021. 2

[10] R. Cai, C. Zhang, C. Wang, L. Zhang, and W.-Y. Ma, "MusicSense: Contextual Music Recommendation using Emotional Allocation Modeling," in *ACM International Conference on Multimedia (ACM MM)*, 2007. 2

[11] M. Won, J. Salamon, N. J. Bryan, G. J. Mysore, and X. Serra, "Emotion embedding spaces for matching music to stories," in *International Society for Music Information Retrieval (ISMIR)*, 2021. 2, 4

[12] S. Reddy and J. Mascia, "Lifetrak: Music in Tune with your Life," in *International Workshop on Human-Centered Media (HCM)*, 2006. 2

[13] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lüke, and R. Schwaiger, "InCarMusic: Context-Aware Music Recommendations in a Car," in *International Conference on Electronic Commerce and Web Technologies (EC-Web)*, 2011. 2

[14] A. Stupar and S. Michel, "Picasso: Automated soundtrack suggestion for multi-modal data," in *International Conference on Information and Knowledge Management (CIKM)*, 2011. 2

[15] A. Zehe, L. Konle, L. K. Dümpelmann, E. Gius, A. Hotho, F. Jannidis, L. Kaufmann, M. Krug, F. Puppe, N. Reiter, A. Schreiber, and N. Wiedmer, "Detecting Scenes in Fiction: A new Segmentation Task," in *European Association of Computational Linguistics (EACL)*, 2021. 2

[16] S. Sarfraz, N. Murray, V. Sharma, A. Diba, L. Van Gool, and R. Stiefelhagen, "Temporally-weighted hierarchical clustering for unsupervised action segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2

[17] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "MPNet: Masked and Permuted pre-training for Language Understanding," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[18] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2019. 2

[19] M. A. Hearst, "Text tiling: Segmenting text into multi-paragraph subtopic passages," *Comput. Linguistics*, vol. 23, pp. 33–64, 1997. 2

[20] M. Riedl and C. Biemann, "Text Segmentation with Topic Models ," *Journal for Language Technology and Computational Linguistics (JLCL)*, vol. 27, no. 47-69, pp. 13–24, 2012. 2

[21] E. Gómez, "Tonal description of polyphonic audio for music content processing," *INFORMS J. Comput.*, vol. 18, pp. 294–304, 2006. 3, 4

[22] P. N. Juslin and J. A. Sloboda, *Handbook of Music and Emotion: Theory, Research, Applications*. Oxford University Press, 2011. 3, 4

[23] O. Lartillot, P. Toiviainen, and T. Eerola, "A matlab toolbox for music information retrieval," in *Data Analysis, Machine Learning and Applications*, 2008. 3

[24] J. Foote and M. L. Cooper, "Media segmentation using self-similarity decomposition," in *IS&T/SPIE Electronic Imaging*, 2003. 3, 4

[25] M. Tapaswi, M. Bäuml, and R. Stiefelhagen, "StoryGraphs: Visualizing Character Interactions as a Timeline," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3

[26] M. Tapaswi, M. Bäuml, and R. Stiefelhagen, "Book2Movie: Aligning Video Scenes with Book Chapters," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3, 5

[27] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*, 2021. 3, 4

[28] M. Brysbaert, A. B. Warriner, and V. Kuperman, "Concreteness ratings for 40 thousand generally known english word lemmas," *Behavior Research Methods*, vol. 46, pp. 904–911, 2014. 3

[29] A. Wang, "An industrial strength audio search algorithm," in *International Society for Music Information Retrieval (ISMIR)*, 2003. 4

[30] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, "GoEmotions: A Dataset of Fine-Grained Emotions," in *Association of Computational Linguistics (ACL)*, 2020. 4

[31] T. Eerola, O. Lartillot, and P. Toiviainen, "Prediction of multidimensional emotional ratings in music from audio using multivariate regression models," in *International Society for Music Information Retrieval (ISMIR)*, 2009. 4

MUSIKA!
FAST INFINITE WAVEFORM MUSIC GENERATION

Marco Pasini **Jan Schlüter**
Institute of Computational Perception, Johannes Kepler University Linz, Austria
marco.pasini.98@gmail.com jan.schlueter@jku.at

ABSTRACT

Fast and user-controllable music generation could enable novel ways of composing or performing music. However, state-of-the-art music generation systems require large amounts of data and computational resources for training, and are slow at inference. This makes them impractical for real-time interactive use. In this work, we introduce Musika, a music generation system that can be trained on hundreds of hours of music using a single consumer GPU, and that allows for much faster than real-time generation of music of arbitrary length on a consumer CPU. We achieve this by first learning a compact invertible representation of spectrogram magnitudes and phases with adversarial autoencoders, then training a Generative Adversarial Network (GAN) on this representation for a particular music domain. A latent coordinate system enables generating arbitrarily long sequences of excerpts in parallel, while a global context vector allows the music to remain stylistically coherent through time. We perform quantitative evaluations to assess the quality of the generated samples and showcase options for user control in piano and techno music generation. We release the source code and pretrained autoencoder weights at github.com/marcoppasini/musika, such that a GAN can be trained on a new music domain with a single GPU in a matter of hours.

1. INTRODUCTION

Generating raw audio remains a difficult task to perform, considering the high temporal dimensionality of waveforms. Recently, a number of techniques based on deep learning architectures have been proposed: however, they often present limitations such as low generated music quality, lack of general coherence between distant time frames and slow generation speed. Regarding unconditional audio generation, autoregressive models are able to generate high quality audio with long-range dependencies; however, the sampling process is extremely slow and inefficient, which hinders possible real-world applications. On the other hand, non-autoregressive models can reach real-time generation, while they struggle to synthesize samples with satisfactory sound quality and are only able to generate samples of a fixed duration.

Considering the current shortcomings of non-autoregressive audio generation systems, in this work we propose Musika, a GAN-based system that allows fast unconditional and conditional generation of audio of arbitrary length. We achieve this by combining the following contributions:

- The use of a raw audio autoencoder which allows to encode samples into lower-dimensional invertible representations that are easier to model. We engineer the autoencoder with the specific goal of maximizing inference speed and minimizing training time, by relying on the generation of magnitude and phase spectrograms with low temporal resolution and an efficient adversarial training process
- The use of a latent coordinate system for the task of infinite-length audio generation
- The addition of a global style conditioning which allows the infinite-length generated samples to be stylistically coherent through time
- The possibility to perform both unconditional and conditional generation, with a variety of different conditioning signals, such as note density and tempo information

By avoiding auto-regression, generation can be fully parallelized and works much faster than real-time even on CPU.

2. RELATED WORK

A popular family of generative models for audio consists in autoregressive models, such as WaveNet [1], SampleRNN [2] and Jukebox [3]. WaveNet was the first model to show that autoregressive generation of raw audio waveforms is possible, and uses dilated convolutions to acquire a large receptive field over the input sample. SampleRNN is a model consisting of a hierarchical stack of recurrent units that are able to model the waveform at different resolutions, and thus capture a larger context and reduce the computational cost required to model the next sample. Jukebox uses a hierarchical VQVAE [4, 5] to encode raw samples into a sequence of discrete codes at different levels. It then uses autoregressive transformers [6] to both generate new top-level samples and upsample them to the lower

© M. Pasini, and J. Schlüter. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M. Pasini, and J. Schlüter, "Musika! Fast Infinite Waveform Music Generation", in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.,* Bengaluru, India, 2022.

levels, accepting different features, such as lyrics, as conditioning. While autoregressive systems can achieve satisfying audio quality and long-range coherence, they suffer from extremely slow generation, as audio samples are produced sequentially. For example, Jukebox requires more than eight hours to generate one minute of audio on a V100 GPU. As an exception, RAVE [7] achieves real-time synthesis by encoding raw audio of a specific domain with a variational autoencoder [8] into a compact latent space and using a lightweight autoregressive model to generate codes: however, the short receptive field of the autoregressive model does not allow to model dependencies over distant time windows in the generated audio.

Non-autoregressive models avoid the slow sequential generation, but are mostly employed for *conditional* audio synthesis. For example, several works focus on the task of inverting a low-dimensional audio representation (often a mel-spectrogram) back to the original waveform, which constitutes a building block of modern text-to-speech (TTS) systems [9–11]. In contrast, literature on long-form non-autoregressive unconditional audio generation is scarce. Systems such as WaveGAN and SpecGAN [12], GANSynth [13], DrumGAN [14] and MP3Net [15] attempt to generate audio of a fixed length using various architectures of Generative Adversarial Networks [16] (GANs). WaveGAN and SpecGAN represent the first works in which a GAN is successfully applied to audio, in the waveform and spectrogram representations, respectively. GANSynth generates instantaneous frequency (IF) and magnitude of spectrograms with high frequency resolution of short monophonic instrument notes [17], showing that generating IFs and magnitudes instead of waveforms is advantageous for highly harmonic sounds. DrumGAN synthesizes drum sounds using the real and imaginary components of a complex STFT spectrogram and demonstrates the effectiveness of this audio representation, first introduced in [18]. Finally, MP3Net achieves minute-long coherent piano music generation using Modified Discrete Cosine Transform (MDCT) spectrograms as audio representations and Progressive GAN [19] as the model: however, the generated samples suffer from low perceived audio quality. To the best of our knowledge, UNAGAN [20] is the only non-autoregressive GAN-based system capable of generating audio of arbitrary length. The model takes a sequence of noise vectors as input and uses a hierarchical structure to achieve short-term coherence in the generated mel-spectrograms, which are then inverted to waveform using a pretrained MelGAN vocoder [9]. However, the model only generates single-channel audio and the independent sampling of noise vectors cause the generated samples to lack coherence through time.

Contrary to autoregressive models, the majority of GAN-based unconditional audio generation models are only able to synthesize audio samples of a fixed length. However, in the field of computer vision, several recent works propose models capable of generating images of arbitrary size, by synthesizing single image patches in parallel and assembling them into the final image. This process results in a fast and efficient generation of images on modern hardware. The two most notable contributions to this line of work are InfinityGAN [21] and ALIS [22]. InfinityGAN generates in parallel single patches that are coherent with each other by disentangling global appearance, local structures and textures, which are then fed into a generator, together with coordinate information, to synthesize the final patch. ALIS proposes the use of latent vectors as anchor points for the coordinate system of the model: the resulting generator is equivariant and can thus produce coherent patches from interpolations of different latent codes. However, both of the methods rely on prior knowledge regarding the particular image domain that is being generated: the experiments are conducted on a dataset of images of landscapes, where the image features of a single patch are spatially invariant on the horizontal dimension and thus permit infinite length generation along the horizontal axis.

The process of generating sequences of encoded representations has been explored in different previous works [4, 5, 23] for both image and audio data. However, these works focus on encoding samples to a discrete set of codes using vector-quantized variational autoencoders, and propose to model sequences of codes using autoregressive models. On the other hand, [24] proposes to autoencode molecules with a basic autoencoder, to then generate sequences of continuous-valued latent vectors with a GAN. This approach manages to circumvent the problematic behaviour of GANs when applied on discrete data [25], in this case molecules in the SMILES format. Similarly to this work, we propose to generate latent representations of audio with the aim of making the generation and training process faster, and achieving coherent generated samples over long time windows.

3. METHOD

Let $\mathbf{x} = \{x_1, ..., x_T\}$ be the waveform of an audio sample. We aim to encode a waveform $\mathbf{x}$ into a sequence of latent vectors $\mathbf{c} = \{c_1, ..., c_{T/r_{time}}\}$ with time compression ratio r_{time}, sampled at a lower sampling rate than the original waveform. We use an autoencoder model to perform this task, such that a reconstruction of the original waveform can be obtained from the encoded latent vectors.

We then aim to model the distribution $p(\mathbf{c})$ with a Generative Adversarial Network (GAN). We employ a latent coordinate system that is used as conditioning for the generator G to generate sequences of latent vectors of arbitrary length. We additionally condition the generator with a variety of conditioning signals, such that the generation process can be guided by human input. Finally, the generated sequence of latent vectors is inverted to a waveform with the previously trained decoder.

3.1 Audio Autoencoder

Considering the inherent high dimensionality of waveforms, generating long sequences of raw audio samples is prohibitively expensive. A frequently used audio representation in the field of speech processing and music informa-