Moûsai: Efficient Text-to-Music Diffusion Models

Anonymous ACL submission

Abstract

Recent years have seen the rapid development of large generative models for text; however, much less research has explored the connection between text and another "language" of communication – *music*. In our work, we bridge text and music via a text-to-music generation model that is highly efficient, expressive, and can handle long-term structure. Specifically, we develop *Moûsai*, a cascading two-stage latent diffusion model that can generate multiple minutes of high-quality stereo music at 48kHz from textual descriptions. Moreover, our model features high efficiency, which enables real-time inference on a single consumer GPU with a reasonable speed. Through experiments and property analyses, we show our model's competence over a variety of criteria compared with existing music generation models.[1]

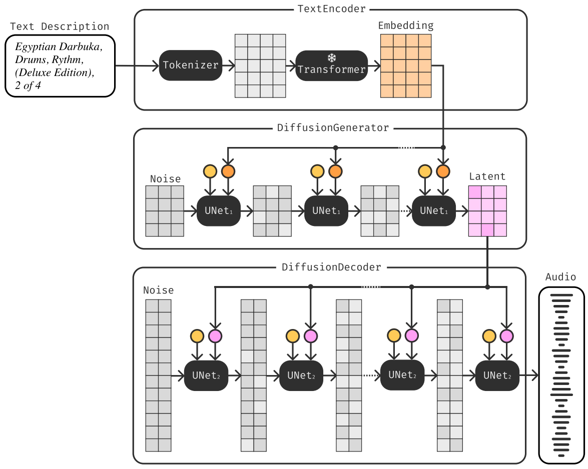


Figure 1: We propose a two-stage cascading diffusion method, where the first stage compresses the music using a novel diffusion autoencoder, and the second stage generates music from the reduced representation conditioned on the encoding of a textual description.

1 Introduction

In recent years, natural language processing (NLP) has made significant strides in understanding and generating human language, due to the advancements in deep learning and large-scale pre-trained models (Radford et al., 2018; Devlin et al., 2019; Brown et al., 2020). While the majority of NLP research has focused on textual data, there exists another rich and expressive "language" of communication – *music*. Music, much like text, can convey emotions (Germer, 2011), stories (Chung, 2006), and ideas (Bicknell, 2002), and has its own unique structure and syntax (Swain, 1995).

In this paper, we further bridge the gap between text and music by leveraging the power of NLP techniques to generate music conditioned on textual input. Through our work, we not only aim to expand the scope of NLP applications, but also contribute to the interdisciplinary research at the intersection of language, music, and machine learning techniques.

However, like text, music generation has long been a challenging task, as it requires multiple aspects at different levels of abstraction (van den Oord et al., 2016; Dieleman et al., 2018). Existing audio generation models explore the use of recursive neural networks (Mehri et al., 2017), adversarial generative networks (Kumar et al., 2019; Kim et al., 2021; Engel et al., 2019; Morrison et al., 2022), autoencoders (Deng et al., 2021), and transformers (Yu et al., 2022). With the recent advancement in diffusion-based generative models in computer vision (Ramesh et al., 2022; Saharia et al., 2022), researchers in speech have also started to explore the use of diffusion models in tasks such as speech synthesis (Kong et al., 2021; Lam et al., 2022; Leng et al., 2022), although only a few these models can apply well to the task of music generation.

Additionally, there are several long-standing challenges in the area of music generation: (1) music

[1] Our code and data are uploaded to the system, and will be released upon acceptance. Our anonymized music samples are available at `https://bit.ly/anonymous-mousai`.

generation at length, as most text-to-audio systems (Forsgren and Martiros, 2022; Kreuk et al., 2022) can only generate *a few seconds* of audio; (2) model efficiency, as many need to run on GPUs for hours to generate just one minute of audio (Dhariwal et al., 2020; Kreuk et al., 2022); (3) lack of diversity of the generated music, as many are limited by their training methods taking in a single modality (resulting in the ability to handle only single-genre music, but *not diverse* genres) (Caillon and Esling, 2021; Pasini and Schlüter, 2022); and (4) easy controllability by text prompts, as most are only controlled by latent states (Caillon and Esling, 2021; Pasini and Schlüter, 2022), the starting snippet of the music (Borsos et al., 2022), or text but are lyrics (Dhariwal et al., 2020) or descriptions of a daily sound like dog barking (Kreuk et al., 2022).

To address these challenges, we propose *Moûsai*,[2] a novel text-conditional two-stage cascading diffusion model. Specifically, the first stage trains a music encoder by diffusion magnitude-autoencoding (DMAE), which compress audio by the novel diffusion autoencoder; and the second stage learns to generate the reduced representation while conditioning on a textual description by text-conditioned latent diffusion (TCLD). The two-stage generation process is shown in Figure 1.

Apart from proposing the novel text-to-music diffusion model, we also introduce some special designs to boost model efficiency, making the model more accessible. First, our DMAE can achieve an audio signal compression rate of 64x. Moreover, we design a lightweight and specialized 1D U-Net architecture. Together, our model achieves a fast inference speed on a single consumer GPU in minutes, and a training time of approximately one week per stage on one A100 GPU, making it possible to train and run the overall system using resources available in most universities.

We train our model on a newly collected dataset, TEXT2MUSIC, with 50K text-music pairs, and show our model's advantage on 11 criteria, such as efficiency, text-music relevance, music quality, and long-context structure.

In summary, our contributions are as follows:

1. We are the first to propose the text-to-music diffusion model using a two-stage cascading latent diffusion modeling process.
2. We achieve high efficiency with a compression rate of 64x, and a specialized U-Net design, which achieves a training time of one week on an A100 consumer GPU, and real-time inference time.
3. Our model outperforms existing baselines by clear margins on 11 different evaluation criteria, demonstrating merits such as high efficiency, text-music relevance, music quality, and long-context structure.

[2] Moûsai is romanized ancient Greek for *Muses*, the sources of artistic inspiration (`https://en.wikipedia.org/wiki/Muses`), and also evokes a blend of *music* and *AI*.

2 Related Work

Connecting Text and Music The connection between text and music lies in the intersection of NLP and computational musicology. Previous work looks into aspects such as the similarity of music and linguistic structures (Papadimitriou and Jurafsky, 2020), music and dialog (Berlingerio and Bonin, 2018), and jointly modeling music and text for emotion detection (Mihalcea and Strapparava, 2012). Apart from several work that generates music from text (Dhariwal et al., 2020; Forsgren and Martiros, 2022), we are the first to explore diffusion models to interact text with music representations.

Generative Models Generative models aim to learn a lower-dimension representation space, and then reconstruct to the high-dimension space conditioning on the given information (Rombach et al., 2022; Yang et al., 2022; Kreuk et al., 2022; Ho et al., 2022). Some effective methods earlier include auto-encoding (Hinton and Salakhutdinov, 2006; Kingma and Welling, 2014), or quantized auto-encoding (van den Oord et al., 2017; Esser et al., 2021; Lee et al., 2022). Recent proposals focus on the quantized representation followed by masked or autoregressive learning on tokens (Villegas et al., 2022; Dhariwal et al., 2020; Kreuk et al., 2022), and diffusion models (Ramesh et al., 2022; Rombach et al., 2022; Saharia et al., 2022), which leads to impressive performance. To the best of our knowledge, we are the first to adapt the cascading diffusion approach for audio generation.

3 Moûsai: Efficient Long-Context Music Generation from Text

Our model Moûsai contains a two-stage training process. In Stage 1, we use diffusion magnitude-autoencoding (DMAE), which compresses the audio waveform 64x using a diffusion autoencoder.

In Stage 2, we use a latent text-to-audio diffusion model, to generate a novel latent space by diffusion while conditioning on text embeddings obtained from a frozen transformer language model.

3.1 Stage 1: Music Encoding by Diffusion Magnitude-Autoencoding (DMAE)

We design the first step of Moûsai to be learning a good music encoder to capture the latent representation space for music. Representation learning is crucial for generative models, as it can be drastically more efficient than handling the high-dimensional raw input data (Rombach et al., 2022; Yang et al., 2022; Kreuk et al., 2022; Ho et al., 2022; Villegas et al., 2022).

Overview To learn the representation space for music, we deploy a diffusion magnitude autoencoder (DMAE) shown in Figure 2. Specifically, we adopt our diffusion-based audio autoencoder, introduced in Section 3.1.3, to compress audio into a smaller latent space by 64x from the original waveform. To train the model, we first convert the waveform to a magnitude spectrogram, which is a better representation for audio models, and then we auto-encode it into a latent representation.

At the same time, we corrupt the original audio with a random amount of noise, and train our 1D U-Net (introduced in Section 3.1.4) to remove that noise. During the noise removal process, we condition the U-Net on the noise level and the compressed latent, which can have access to a reduced version of the non-noisy audio.

3.1.1 v-Objective Diffusion

We use the $\boldsymbol{v}$-objective diffusion process as proposed by Salimans and Ho (2022). Suppose we have a sample $\boldsymbol{x}_0$ from a distribution $p(\boldsymbol{x}_0)$, some noise schedule $\sigma_t \in [0,1]$, and some noisy data point $\boldsymbol{x}_{\sigma_t} = \alpha_{\sigma_t}\boldsymbol{x}_0 + \beta_{\sigma_t}\boldsymbol{\epsilon}$. The $\boldsymbol{v}$-objective diffusion tries to estimate a model $\hat{\boldsymbol{v}}_{\sigma_t} = f(\boldsymbol{x}_{\sigma_t}, \sigma_t)$ by minimizing the following objective:

$$\mathbb{E}_{t\sim[0,1],\sigma_t,\mathbf{x}_{\sigma_t}}\left[\|f_\theta(\mathbf{x}_{\sigma_t},\sigma_t) - \mathbf{v}_{\sigma_t}\|_2^2\right] \,, \quad (1)$$

where $\boldsymbol{v}_{\sigma_t} = \frac{\partial \boldsymbol{x}_{\sigma_t}}{\sigma_t} = \alpha_{\sigma_t}\boldsymbol{\epsilon} - \beta_{\sigma_t}\boldsymbol{x}_0$, for which we define $\phi_t := \frac{\pi}{2}\sigma_t$, and obtain its trigonometric values $\alpha_{\sigma_t} := \cos(\phi_t)$, and $\beta_{\sigma_t} := \sin(\phi_t)$.

3.1.2 DDIM Sampler for Denoising

The denoising step uses ODE samplers to turn noise into a new data point by estimating the rate of change. In this work, we adopt the DDIM sampler (Song et al., 2021), which we find to work well and have a reasonable tradeoff between the number of steps and audio quality. The DDIM sampler denoises the signal by repeated application of the following:

$$\hat{\boldsymbol{v}}_{\sigma_t} = f_\theta(\boldsymbol{x}_{\sigma_t}, \sigma_t) \quad (2)$$

$$\hat{\boldsymbol{x}}_0 = \alpha_{\sigma_t}\boldsymbol{x}_{\sigma_t} - \beta_{\sigma_t}\hat{\boldsymbol{v}}_{\sigma_t} \quad (3)$$

$$\hat{\boldsymbol{\epsilon}}_{\sigma_t} = \beta_{\sigma_t}\boldsymbol{x}_{\sigma_t} + \alpha_{\sigma_t}\hat{\boldsymbol{v}}_{\sigma_t} \quad (4)$$

$$\hat{\boldsymbol{x}}_{\sigma_{t-1}} = \alpha_{\sigma_{t-1}}\hat{\boldsymbol{x}}_0 + \beta_{\sigma_{t-1}}\hat{\boldsymbol{\epsilon}}_t, \quad (5)$$

which estimates both the initial data point and the noise at the step σ_t, for some T-step noise schedule $\sigma_T, \ldots, \sigma_0$ as a sequence evenly spaced between 1 and 0.

3.1.3 Diffusion Autoencoder for Audio Input

We propose a new diffusion autoencoder that first encodes a magnitude spectrogram into a compressed representation, and later injects the latent into intermediate channels of the decoding modules. The standard method to do diffusion, such as the image diffusion model (Rombach et al., 2022), is to compress the input into a lower-dimensional representation space and apply the diffusion process on the reduced latent space. We further compress and enhance the representation space by diffusion-based autoencoding (Preechakul et al., 2022), which is first introduced in computer vision, as a way to condition the diffusion process on a compressed latent vector of the input itself. Since diffusion serves as a more powerful generative decoder, and hence the input can be reduced to latent representations with higher compression ratios.

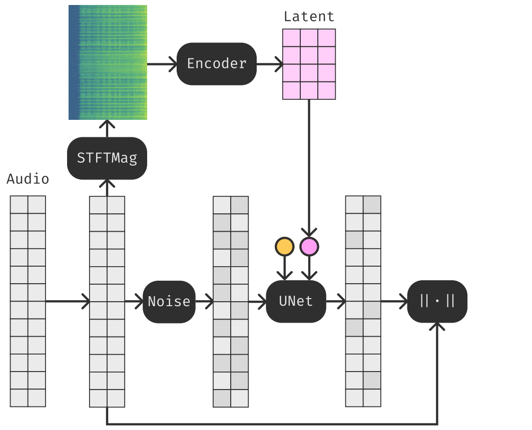


Figure 2: The training scheme of our diffusion magnitude autoencoder (DMAE). When denoising (bottom right), we condition the U-Net on the noise level (○) and compressed latent representation (○) from a reduced version of the non-noisy audio (the pink matrix).

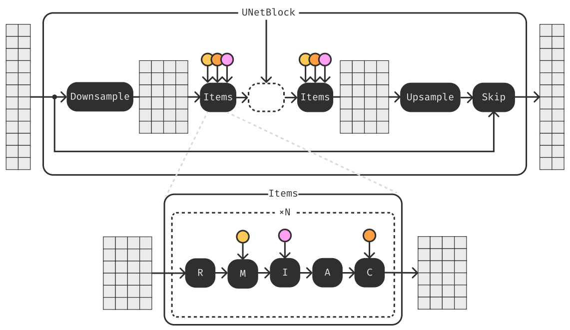


Figure 3: Our proposed 1D U-Net architecture. Each `UNetBlock` (top) consists of several U-Net items (bottom). In each U-Net item (bottom), we use a 1D convolutional ResNet (R), and a modulation unit (M) to provide the diffusion noise level as a feature vector conditioning (●). For Stage 1, we use an inject item (I) to inject external channels as conditioning (●), and for Stage 2, we use an attention item (A) to share time-wise information, and a cross-attention item (C) to condition on an external (text) embedding (●). Moreover, for the `UNetBlocks`, we can recursively nest them, which we indicate by the inner dashed region on the top.

3.1.4 Efficient and Enriched 1D U-Net

Another crucial module in our model is the efficient 1D U-Net that we design. We identify that the vanilla U-Net architecture (Ronneberger et al., 2015), originally introduced for medial image segmentation, has relatively limited efficiency and speed, as it uses an hourglass convolutional-only 2D architecture with skip connections.

Hence, we propose a novel U-Net with only 1D convolutional kernels, which is more efficient than the original 2D architecture in terms of speed, and can be successfully used both on waveforms or on spectrograms if each frequency is considered as a different channel.

Moreover, we infuse our 1D U-Net with multiple new components, as illustrated in Figure 3: a ResNet residual 1D convolutional unit, a modulation unit to alter the channels given features from the diffusion noise level, and an inject item to concatenate external channels to the ones at the current depth. Note that inject items are applied only at a specific depth in the decoder in the first stage to condition on the latent representation of the music.

In summary, our novel 1D U-Net features more modern convolutional blocks, a variety of attention blocks, conditioning blocks, and improved skip connections, maintaining an efficient skeleton of the hourglass architecture.

3.1.5 Overall Model Architecture

Our entire Stage 1, DMAE, works as follows. Let $\boldsymbol{w}$ be a waveform of shape $[c, t]$ for c channels and t timesteps, and $(\boldsymbol{m_w}, \boldsymbol{p_w}) = \text{stft}(\boldsymbol{w}; n = 1024, h = 256)$ be the magnitude and phase obtained from a short-time furier tranform of the waveform with a window size of 1024 and hop-length of 256. Then the resulting spectrograms will have shape $[c \cdot n, \frac{t}{h}]$. We discard phase and encode the magnitude into a latent $\boldsymbol{z} = \mathcal{E}_{\boldsymbol{\theta}_e}(\boldsymbol{m_w})$ using a 1D convolutional encoder. The original waveform is then reconstructed by decoding the latent using a diffusion model $\hat{\boldsymbol{w}} = \mathcal{D}_{\boldsymbol{\theta}_d}(\boldsymbol{z}, \boldsymbol{\epsilon}, s)$, where $\mathcal{D}_{\boldsymbol{\theta}_d}$ is the diffusion sampling process with starting noise $\boldsymbol{\epsilon}$ and s is the number of decoding (sampling) steps. The decoder is trained with $\boldsymbol{v}$-objective diffusion while conditioning on the latent $f_{\boldsymbol{\theta}_d}(\boldsymbol{w}_{\sigma_t}; \sigma_t, \boldsymbol{z})$, where $f_{\boldsymbol{\theta}_d}$ is the proposed 1D U-Net, called repeatedly during decoding.

Since only the magnitude is used and phase is discarded, this diffusion autoencoder is simultaneously a compressing autoencoder and vocoder. By using the magnitude spectrograms, higher compression ratios can be obtained than autoencoding directly the waveform. We found that waveforms are less compressible and efficient to work with. Similarly, discarding phase is beneficial to obtaining higher compression ratios for the same level of quality. The diffusion model can easily learn to generate a waveform with realistic phase even if conditioned only on the encoded magnitude.

In this way, the latent space for music can serve as the starting point for our text-to-music generator, which will be introduced next. To ensure this representation space fits the next stage, we apply a $\tanh$ function on the bottleneck, keeping the values in the range $[-1, 1]$. Note that we do not use a more disentangled bottleneck, such as the one in VAEs (Kingma and Welling, 2014), as its additional regularization reduces the amount of allowed compressibility.

3.2 Stage 2: Text-to-Music Generation by Text-Conditioned Latent Diffusion (TCLD)

Based on the learned music representation space, in this stage, we guide the music generation with text descriptions.

Overview As shown in Figure 4, we propose a text-conditioned latent diffusion (TCLD) process.

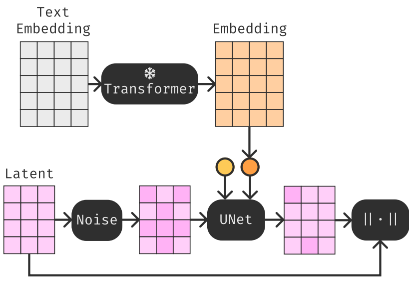


Figure 4: The training scheme of our text-conditioned latent diffusion (TCLD) generator. During the denoising process, we provide the U-Net a feature vector (●) and a text embedding (●).

Specifically, we first corrupt the latent space of music with a random amount of noise, then train a series of U-Nets to remove the noise, and condition the U-Nets' denoising process on a text prompt encoded by a transformer model. In this way, the generated music both conforms to the latent space of music and corresponds to the text prompt.

3.2.1 Text Conditioning

To obtain the text embeddings, prior work on text-conditioning suggests either learning a joint data-text representation (Li et al., 2022; Elizalde et al., 2022; Ramesh et al., 2022), or using embeddings from pre-trained language model as direct conditioning (Saharia et al., 2022; Ho et al., 2022) of the latent model. In our TCLD model, we follow the practice in Saharia et al. (2022) to use a pre-trained and frozen T5 language model (Raffel et al., 2020) to generate text embeddings from the given description. We use the classifier-free guidance (CFG) (Ho and Salimans, 2022) with a learned mask applied on batch elements with a probability of 0.1 to improve the strength of the text-embedding during inference.

3.2.2 Adapting the U-Net for Text Conditioning

To enable the U-Net to condition on the text embedding $\boldsymbol{e}$, we append two additional blocks to the U-Net: an attention item to share long-context structural information, and a cross-attention item to condition on the text embeddings, as in Figure 3. These attention blocks ensure information sharing over the entire latent space, which is crucial to learn long-range audio structure.

Given the compressed size of the latent space, we also increase the size of this inner U-Net to be larger than the first stage. And due to our efficiency design, it maintains a reasonable training and inference speed, even with large parameter counts.

3.2.3 Overall Model Architecture

We illustrate the detailed process in Figure 4. Consistent with the previous stage, we use $\boldsymbol{v}$-objective diffusion and the 1D U-Net architecture. When condition on the text embedding $\boldsymbol{e}$, we use the U-Net configuration $f_{\boldsymbol{\theta}_g}(\boldsymbol{z}_{\sigma_t}; \sigma_t, \boldsymbol{e})$ to generate the compressed latent $\boldsymbol{z} = \mathcal{E}_{\boldsymbol{\theta}_e}(\boldsymbol{m}_{\boldsymbol{w}})$. Then, the generator $\mathcal{G}_{\boldsymbol{\theta}_g}(\boldsymbol{e}, \boldsymbol{\epsilon}, s)$ applies DDIM sampling and calls the U-Net s times to generate an approximate latent $\hat{\boldsymbol{z}}$ from the text embedding $\boldsymbol{e}$ and starting noise $\boldsymbol{\epsilon}$. The final generation stack during inference to obtain a waveform is

$$\hat{\boldsymbol{w}} = \mathcal{D}_{\boldsymbol{\theta}_d}(\mathcal{G}_{\boldsymbol{\theta}_g}(\boldsymbol{e}, \boldsymbol{\epsilon}_g, s_g), \boldsymbol{\epsilon}_d, s_d) \,. \quad (6)$$

4 Experimental Setup

4.1 Collection of the Text2Music Dataset

To provide a fertile ground to train our text-to-music model on, we collect a new dataset, Text2Music, which consists of 50K text-music pairs totaling 2,500 hours. We ensure a high quality of stereo music sampled at 48kHz and cover a wide variety of music spanning multiple genres, artists, instruments, and provenience. Many existing open-source music datasets, such as Gillick et al. (2019); Hawthorne et al. (2019a), have limitations in terms of the specific musical instruments they encompass. While some datasets, like Engel et al. (2017); Boulanger-Lewandowski et al. (2012), cover a broader array of instruments, they fall short in representing a wide variety of genres. This inadequacy underscores the need for a more comprehensive dataset that encompasses a rich tapestry of musical genres and diverse instrumentation.

As for the procedure to collect the music, we follow Spotify's top recommendations to collect seven very large playlists, each containing on average 7K pieces of music. We iterate through every music sample in these playlists, for which we use the name of the music to search and download the music from YouTube, and we use the metadata to compose its corresponding text description, which contains the music title, author, album name, genre, and year of release.

We show the statistics about the diverse set of genres in our Text2Music dataset in Table 1.

Genre	# Pieces	Percentage (%) in Dataset
Pop	5,498	27.29
Electronic	3,875	19.38
Rock	3,584	17.79
Metal	1,796	8.92
Hip Hop	818	4.06
Others	4,492	22.56

Table 1: Our TEXT2MUSIC dataset covers a variety of music, e.g., pop, electronic, rock, metal, hip pop, etc.

4.2 Implementation Details

Our diffusion autoencoder has 185M parameters, and text-conditional generator has 857M parameters, with more architecture details in Appendix A.3. We train the music autoencoder on random crops of length 2^{18} ($\sim$5.5s at 48kHz), and the text-conditional diffusion generation model on fixed crops of length 2^{21} ($\sim$44s at 48kHz) encoded in the 32-channels, 64x compressed latent representation. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 10^{-4}, β_1 of 0.95, β_2 of 0.999, ϵ of 10^{-6}, and weight decay of 10^{-3}. And we use an exponential moving average (EMA) with $\beta = 0.995$ and power of 0.7.

5 Evaluation

5.1 Assessment Criteria Overview

Evaluating music is a highly challenging task. We survey a large number of papers, and find that previous work adopts a variety of objective and subjective metrics,[3] and the gist is that no single metric is perfect. After careful thinking, we design a comprehensive set of evaluation metrics covering three categories with a total of *11 metrics*, including both automatic and human evaluations. In the following, we will introduce the overall property analysis (Section 5.2), such as the sample rate, prompt type, and music type; efficiency (Section 5.3); text-music relevance (Section 5.4); music quality (Section 5.5); and long-term structure of the music (Section 5.6).

For fair comparison, we train all the baseline models from scratch on our TEXT2MUSIC dataset. Note that the recent models Noise2Music (Huang et al., 2023) does not release their source code, and MusicLM (Agostinelli et al., 2023) is not as efficient as our model in that it originally used 280K hours of training data, and, when training from scratch, it cannot converge on our 2.5K hours dataset.

5.2 Property Analysis

Comparing the overall properties of various models in Table 2, we see a set of impressive properties of the Moûsai model: (1) We are among the very few that can control music generation easily by *text descriptions* of the type of music we want, as most other models do not take text as input (van den Oord et al., 2016; Caillon and Esling, 2021; Borsos et al., 2022), or take only lyrics or descriptions of daily sounds (e.g., "a dog barking") (Kreuk et al., 2022; Dhariwal et al., 2020). The only other text-to-music model is the Riffusion model (Forsgren and Martiros, 2022), which only works with very short length of 5 seconds.

(2) Our model is also among the very few that enables *long-context* music generation for several minutes, among all others that can only generate seconds (van den Oord et al., 2016; Forsgren and Martiros, 2022; Kreuk et al., 2022; Pasini and Schlüter, 2022), except for Jukebox (Dhariwal et al., 2020) which generates songs given lyrics and takes very long to run inference.

(3) Moreover, we also highlight the *diversity* of music we generate, as our model design enables multi-genre music training, instead of single-genre ones in previous models (Caillon and Esling, 2021; Pasini and Schlüter, 2022), and we can find rhythm, loops, riffs, and occasionally even entire choruses in our generated music.

5.3 Efficiency of Our Model

Efficiency is another highlight of our model, where we only need an inference time similar to the audio length on a consumer GPU, which is several minutes, while many other text-to-audio models take many GPU hours (Dhariwal et al., 2020; Kreuk et al., 2022), as in Table 2. Our model is very friendly for research at university labs, as each model can be trained on a single A100 GPU in 1 week of training using a batch size of 32.

We also calculate the exact inference statistics for our Moûsai vs. Riffusion models in Table 4, and find that our model needs less than 1/5 the inference time, and almost half of the inference memory than Riffusion does. To make a fair comparison

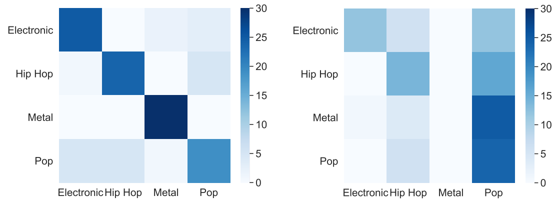
[3]The common metrics we surveyed include quality (Goel et al., 2022), fidelity (Goel et al., 2022; Hawthorne et al., 2019b; Hyun et al., 2022), musicality (Goel et al., 2022; Yu et al., 2022; Dhariwal et al., 2020), diversity (Goel et al., 2022; Dhariwal et al., 2020), and structure (Yu et al., 2022; Leng et al., 2022; Dhariwal et al., 2020).

Model	Sample Rate↑	Len.↑	Input (Text ✓)	Music (Diverse↑)	Example	Infer. Time↓	Data
WaveNet (2016)	16kHz@1	Secs	None	Piano or speech	Piano	**= Audio len.***	260
Jukebox (2020)	44.1kHz@1	**Mins***	**Lyrics**, author, etc.	Song with the lyrics	Song	Hours	70K
RAVE (2021)	**48kHz@2**	Secs*	Latent	Single-genre Music	Strings	**= Audio len.***	100
AudioLM (2022)	16kHz@1	Secs*	Beginning of the music	Piano or speech	Piano	Mins	40K
Musika (2022)	22.5kHz@2	Secs	Context vector	Single-genre Music	Piano	**= Audio len.***	1K
Riffusion (2022)	44.1kHz@1	5s	**Text** (genre, author, etc.)	**Music of any genre**	Jazzy clarinet	Mins	–
AudioGen (2022)	16kHz@1	Secs*	**Text** (a phrase/sentence)	Daily sounds	Dog barks	Hours	4K
Moûsai (Ours)	**48kHz@2**	**Mins***	**Text** (genre, author, etc.)	**Music of any genre**	African drums	**= Audio len.**	2.5K

Table 2: Comparison of our Moûsai model with previous music/audio generation models. We compare the followings aspects: (1) audio sample rate@the number of channels (Sample Rate↑, where the higher the better), (2) context length of the generated music (Len.↑, where the higher the more capable the model is to generate structural music; * indicates variable length, where we assume that autoregressive methods are variable by default, with an upper-bound imposed by attention), (3) input type (Input, where we feature using Text as the condition for the generation), (4) type of the generate music (Music, where the more Diverse↑ genre, the better), (5) an example of the generated music type (Example), (6) inference time (Infer. Time↓, where the shorter the better, and since the music length is seconds or minutes, the inference time equivalent to the audio length is the shortest, and we use * to show models that can run inference fast on CPU), and (7) total length of the music in the training data in hours (Data).

Model	Inf. Time (s) (↓)	Mem. (G) (↓)	RTF (↓)
Riffusion	218.0	8.85	5.07
Moûsai	**49.2**	**5.04**	1.14

Table 3: Efficiency evaluation of our Moûsai and Riffusion in terms of the inference time (Inf. Time), inference memory (Mem.) , and real time factor (RTF) to generate a single 43-second music clip.



(a) Confusion matrix for the music pieces generated by Moûsai. (y-axis: true genre; x-axis: inferred genre.)

(b) Confusion matrix for the music pieces generated by the Riffusion model.

Figure 5: For the text-music relevance check, we ask the annotators to infer the ground-truth genres of the generated music by (a) our model and (b) the Riffusion model. The darker diagonal means better results.

5.4 Evaluating the Text-Music Relevance

To assess how much the generated music corresponds to the given text prompt, we deploy both human and automatic evaluations.

Relevance & Distinctiveness by Human Evaluation We design a listener test where the annotators need to infer some coarse information of the text prompt behind a given piece of generated music. Since it is too challenging to infer the exact text prompt, we only ask annotators to infer the music genre indicated in the prompt.

To prepare the ground-truth prompts, we compose a list of 40 random text prompts spanning across the four most common music genres in our TEXT2MUSIC dataset: electronic, hip hop, metal, and pop. See Appendix C.1 for the entire list of prompts. Inspired by the two-alternative forced choice (2AFC) experiment design, we design a *four-alternative forced choice (4AFC)* paradigm, where the annotators need to categorize each music sample into exactly one of the four provided categories. See annotation details in Appendix C.1.

In Figure 5, we can see that our Moûsai model has the most mass on the diagonal (i.e., correctly identified), while the Riffusion model tends to generate generic samples that are mostly identified as pop for all ground-truth genres. This shows that the music generated by our model is both relevant to the test and distinct enough with the given genre against others.

Relevance by CLAP For automatic evaluation, we adopt the commonly used CLAP score (Wu et al., 2023) to quantify the alignment between the generated audio and the corresponding text. From Table 4, we can see that our model is two times better than Riffusion in terms of CLAP score, and also much faster in inference time.

Model	CLAP Score for Text-Music Relevance (↑)
Riffusion	0.06
Moûsai	**0.13**

Table 4: CLAP scores of our Moûsai and Riffusion.

5.5 Evaluating the Music Quality

We first introduce the four evaluation metrics for music quality, and then describe the results.

5.5.1 Metrics for Music Quality

To evaluate the quality of the generated music, we adopt four metrics: the automatic score by FAD, a music Turing test, and human evaluation on musicality and audio clarity.

For automatic evaluation, we deploy the widely adopted *Fréchet Audio Distance (FAD)* (Kilgour et al., 2019) to assess the fidelity of the generated music distribution in comparison to the real music distribution (i.e., how *similar* the generated music is to the authentic music). To facilitate the computation of FAD, we employ the commonly used PANN model (Kong et al., 2020) as a means to effectively encode the music.

Then, we also set up three human evaluations, all on a scale of 1 (worst) to 5 (best). First, we let human annotators to assess the *authenticity/fidelity* of the generated music via a music Turing test (Goel et al., 2022; Hawthorne et al., 2019b; Hyun et al., 2022). See more evaluation details in Appendix C.2.

The other two metrics we deploy are *musicality* and *audio clarity*. For musicality, we let human annotators rate the melodiousness and harmoniousness (Seitz, 2005) of the given music. And for audio clarity, or quality (Goel et al., 2022), we let them judge how close the quality is to a walkie-talkie (worst) or a high-quality studio sound system (best). The detailed setup of all our human evaluations are in Appendix C.2 and Appendix C.3.

5.5.2 Results

We show the evaluation results on all five metrics in Table 5. We can see that, on the automatic evaluation of FAD, our model has the best score, which is one magnitude smaller than previous models. Moreover, it also shows strong performance across the human evaluation metrics, outperforming the other two models on the music Turing test, harmoniousness, and sound clarity, as well as being comparable on the melodiousness metric.

Model	FAD (↓)	Fidelity	Melody	Harmony	Clarity
Riffusion	0.0018	2.8	2.66	2.48	2.37
Musika	0.0020	3.04	**3.21**	3.04	2.88
Moûsai	**0.00015**	**3.17**	3.15	**3.08**	**2.92**

Table 5: Music quality scores for the three models.

5.6 Long-Term Structure of the Music

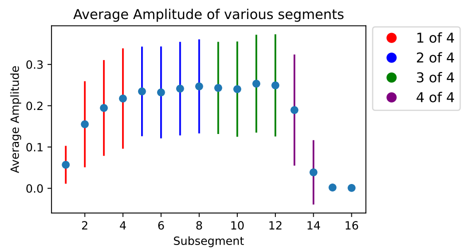


Figure 6: The average amplitude and variation of 1K random music samples spanning different segments.

In music composition, the arrangement of a piece typically follows a gradual introduction, a main body with the core content, and a gradual conclusion, also called the sonata form (Webster, 2001). Accordingly, we look into whether our generated music also shows such long-term structure. Using the same text prompt, we can generate different segments/intervals of it by attaching the expression "1/2/3/4 out of 4" at the end of the text prompt, such as "Italian Hip Hop 2022, 3 of 4." We visualize the results in Figure 6, where we see the first segment shows a gradual increase in both the average amplitude and variance, followed by continuously high average amplitude and variance throughout Segments 2 and 3, and finally concluding with a gradual decline in the last segment.

5.7 Effect of Hyperparameters

We also explore the effect of different hyperparameters, and find that increasing the number of attention blocks (e.g., from a total of 4–8 to a total of 32+) in the latent diffusion model can improve the general structure of the songs, thanks to the long-context view. Also, if the model is trained without attention blocks, the context provided by the U-Net is not large enough to learn any meaningful long-term structure. We describe other variations of hyperparameters and findings in Appendix E.

6 Conclusion

In this work, we presented Moûsai, a novel text-to-music generation model using latent diffusion. We show that, in contrast to earlier approaches, our model can generate minutes of music in real-time on a consumer GPU, with good music quality and text-audio binding. The work helps pave the way towards higher-quality, longer-context text-to-music generation for future applications.

Limitations and Future Work

Data Scale Enhancing the scale of both data and the model holds promising potential for yielding significant improvements in quality. Following (Dhariwal et al., 2020; Borsos et al., 2022), we suggest training with 50K-100K hours instead of 2.5K. Computer Vision studies like Saharia et al. (2022) show that utilizing larger pretrained language models for text embeddings plays an important role in achieving better quality outcomes. Drawing upon this, we hypothesize that the application of a larger pretrained language model to our second-stage model can similarly contribute to enhanced quality outcomes.

Models Some promising future modelling approaches that can be explored in future work include: (1) training diffusion models using perceptual losses on the waveforms instead of L2 — this might help decrease the initial size of the U-Net, as we would not have to process non-perceivable sounds, (2) improving the quality of the diffusion autoencoder by using mel-spectrograms instead of magnitude spectrograms as input, (3) other types of conditioning which are not text-based might be useful to navigate the audio latent space, which is often hard to describe in words — DreamBooth-like models (Ruiz et al., 2022), and (4) more sophisticated diffusion samplers to achieve higher quality for the same number of sampling steps, or similarly more advanced distillation techniques (Salimans and Ho, 2022).

Ethical Considerations

Our work aims to bridge the gap between text and music generation, enabling the creation of expressive and high-quality music from textual descriptions. While this research has the potential to benefit various applications, such as music therapy, entertainment, and education, we recognize that it may also raise concerns in terms of copyright, cultural appropriation, and the potential misuse of generated content.

Copyright and Intellectual Property: Our model may generate music that resembles existing copyrighted works, which could lead to potential legal disputes. First of all, for research-only use, it is exempted from copyright infringement. For other purposes, we suggest incorporating mechanisms to detect and avoid generating music that closely resembles copyrighted material.

Economic Impact on Musicians and Composers: The widespread adoption of text-to-music generation models may have economic implications for musicians and composers, potentially affecting their livelihoods. We believe that our model should be used as a tool to augment and inspire human creativity, rather than replace it. We encourage collaboration between AI researchers, musicians, and composers to explore new ways of integrating AI-generated music into the creative process, ensuring that the technology benefits all stakeholders.

In conclusion, we are committed to conducting our research responsibly and ethically. We encourage the research community to engage in open discussions about the ethical implications of text-to-music generation models and to develop guidelines and best practices for their responsible use. By addressing these concerns, we hope to contribute to the development of AI technologies that benefit society and promote creativity, while respecting the rights and values of all stakeholders.

References

Classical music: 50 greatest composers of all time. BBC Music Magazine.

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. 2023. MusicLM: Generating music from text.

Michele Berlingerio and Francesca Bonin. 2018. Towards a music-language mapping. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jeanette Bicknell. 2002. Can music convey semantic content? a kantian approach. *The Journal of Aesthetics and Art Criticism*, 60(3):253–261.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matthew Sharifi, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. 2022. AudioLM: A language modeling approach to audio generation. *CoRR*, abs/2209.03143.

Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. 2012. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Antoine Caillon and Philippe Esling. 2021. RAVE: A variational autoencoder for fast and high-quality neural audio synthesis. *CoRR*, abs/2111.05011.

Sheng-Kuan Chung. 2006. Digital storytelling in integrated arts education. *The International Journal of Arts Education*, 4(1):33–50.

Kangle Deng, Aayush Bansal, and Deva Ramanan. 2021. Unsupervised audiovisual synthesis via exemplar autoencoders. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A generative model for music. *CoRR*, abs/2005.00341.

Sander Dieleman, Aäron van den Oord, and Karen Simonyan. 2018. The challenge of realistic music generation: Modelling raw audio at scale. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8000–8010.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. 2022. CLAP: learning audio concepts from natural language supervision. *CoRR*, abs/2206.04769.

Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi. 2017. Neural audio synthesis of musical notes with wavenet autoencoders.

Jesse H. Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. 2019. Gansynth: Adversarial neural audio synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Patrick Esser, Robin Rombach, and Björn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 12873–12883. Computer Vision Foundation / IEEE.

Seth* Forsgren and Hayk* Martiros. 2022. Riffusion - Stable diffusion for real-time music generation.

Mark Germer. 2011. *Notes*, 67(4):760–765.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Computational Natural Language Learning (CoNLL)*.

Karan Goel, Albert Gu, Chris Donahue, and Christopher Ré. 2022. It's raw! audio generation with state-space models. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 7616–7633. PMLR.

Gal Greshler, Tamar Rott Shaham, and Tomer Michaeli. 2021. Catch-a-waveform: Learning to generate audio from a single short example. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 20916–20928.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. 2019a. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *International Conference on Learning Representations*.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse H. Engel, and Douglas Eck. 2019b. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey A. Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. 2022. Imagen video: High definition video generation with diffusion models. *CoRR*, abs/2210.02303.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598.

Qingqing Huang, Daniel S. Park, Tao Wang, Timo I. Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Havnø Frank, Jesse H. Engel, Quoc V. Le, William Chan, and Wei Han. 2023. Noise2music: Text-conditioned music generation with diffusion models. *CoRR*, abs/2302.03917.

Lee Hyun, Taehyun Kim, Hyolim Kang, Minjoo Ki, Hyeonchan Hwang, Kwanho Park, Sharang Han, and Seon Joo Kim. 2022. Commu: Dataset for combinatorial music generation. *CoRR*, abs/2211.09385.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. 2019. Fréchet audio distance: A metric for evaluating music enhancement algorithms.

Minsu Kim, Joanna Hong, and Yong Man Ro. 2021. Lip to speech synthesis with visual context attentional GAN. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 2758–2770.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. 2020. Panns: Large-scale pretrained audio neural networks for audio pattern recognition.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2021. Diffwave: A versatile diffusion model for audio synthesis. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. 2022. AudioGen: Textually guided audio generation. *CoRR*, abs/2209.15352.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C. Courville. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14881–14892.

Max W. Y. Lam, Jun Wang, Dan Su, and Dong Yu. 2022. BDDM: bilateral denoising diffusion models for fast and high-quality speech synthesis. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. Autoregressive image generation using residual quantization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 11513–11522. IEEE.

Yichong Leng, Zehua Chen, Junliang Guo, Haohe Liu, Jiawei Chen, Xu Tan, Danilo P. Mandic, Lei He, Xiang-Yang Li, Tao Qin, Sheng Zhao, and Tie-Yan Liu. 2022. Binauralgrad: A two-stage conditional diffusion probabilistic model for binaural audio synthesis. *CoRR*, abs/2205.14807.

Manling Li, Ruochen Xu, Shuohang Wang, Luowei Zhou, Xudong Lin, Chenguang Zhu, Michael Zeng, Heng Ji, and Shih-Fu Chang. 2022. Clip-event: Connecting text and images with event structures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 16399–16408. IEEE.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron C. Courville, and Yoshua Bengio. 2017. SampleRNN: An unconditional end-to-end neural audio generation model. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Rada Mihalcea and Carlo Strapparava. 2012. Lyrics, music, and emotions. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 590–599, Jeju Island, Korea. Association for Computational Linguistics.

Max Morrison, Rithesh Kumar, Kundan Kumar, Prem Seetharaman, Aaron C. Courville, and Yoshua Bengio. 2022. Chunked autoregressive GAN for conditional waveform synthesis. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Isabel Papadimitriou and Dan Jurafsky. 2020. Learning Music Helps You Read: Using transfer to study linguistic structure in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6829–6839, Online. Association for Computational Linguistics.

Marco Pasini and Jan Schlüter. 2022. Musika! fast infinite waveform music generation. *CoRR*, abs/2208.08706.

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. 2022. Diffusion autoencoders: Toward a meaningful and decodable representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10609–10619. IEEE.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2022. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *ArXiv*, abs/2208.12242.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *CoRR*, abs/2205.11487.

Tim Salimans and Jonathan Ho. 2022. Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jay A Seitz. 2005. Dalcroze, the body, movement and musicality. *Psychology of music*, 33(4):419–435.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Joseph P Swain. 1995. The concept of musical syntax. *The Musical Quarterly*, 79(2):281–308.

A. M. TURING. 1950. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. In *The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016*, page 125. ISCA.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6306–6315.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. 2022. Phenaki: Variable length video generation from open domain textual description. *CoRR*, abs/2210.02399.

James Webster. 2001. Sonata form. *The new Grove dictionary of music and musicians*, 23:687–698.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2023. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation.

Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. 2022. Diffsound: Discrete diffusion model for text-to-sound generation. *CoRR*, abs/2207.09983.

Botao Yu, Peiling Lu, Rui Wang, Wei Hu, Xu Tan, Wei Ye, Shikun Zhang, Tao Qin, and Tie-Yan Liu. 2022. Museformer: Transformer with fine- and coarse-grained attention for music generation. *CoRR*, abs/2210.10349.

A More Data Details

A.1 Data Collection Rationale

We have several desiderata when collecting the dataset. The data (1) must have text data paired with the music piece, and (2) must consistitute a *large* size, which means that our data crawling procedure needs to be scalable, without tedious manual efforts to curate. Note that it is crucial to get a large-sized dataset in order to unleash the performance of audio generation diffusion models.

A.2 Training setup for the text-music pairs

For the textual description, we use metadata such as the title, author, album, genre, and year of release. Given that a song could span longer than 44s, we append a string indicating which chunk is currently being trained on, together with the total chunks the song is made of (e.g., *1 of 4*). This allows to select the region of interest during inference. Hence, an example prompt is like *"Egyptian Darbuka, Drums, Rythm, (Deluxe Edition), 2 of 4."* To make the conditioning more robust, we shuffle the list of metadata and drop each element with a probability of 0.1. Furthermore, for 50% of the times we concatenate the list with spaces and the other 50% of the times we use commas to make the interface more robust during inference. Some example prompts in our dataset can be seen in Table 6.

Example Text Prompts in Our Dataset
Nr. 415 (Premium Edition), german hip hop, 2 of 7, 2012, XATAR, Konnekt
30 Años de Exitos, Mundanzas, 2 of 6, latin pop, Lupita D'Alessio, 2011
emo rap 2018 Runaway Lil Peep 4 of 5
Alone, Pt. II (Remixes) 2020 electro house Alone, Pt. II - Da Tweekaz Remix Alan Walker

Table 6: Example text prompts in our dataset.

A.3 Model Architecture and Parameters

Our diffusion autoencoder has 185M parameters, with 7 nested U-Net blocks of increasing channel count ([256, 512, 512, 512, 1024, 1024, 1024]), for which we downsample each time by 2, except for the first block ([1, 2, 2, 2, 2, 2, 2]). This makes the compression factor for our autoencoder to be 64x. Depending on the desired speed/quality tradeoff, more or less compression can be applied in this first stage. Following our single GPU constraint, we find that 64x compression factor is a good balance to make sure the second stage can work on a reduced representation. We discuss more about this tradeoff in Appendix E.5. The diffusion autoencoder only uses ResNet and modulation items with the repetitions [1, 2, 2, 2, 2, 2, 2]. We do not use attention, to allow decoding of variable and possibly very long latent representations. Channel injection only happens at depth 4, which matches the output of the magnitude encoder latent, after applying the $\tanh$ function.

Our text-conditional generator has 857M parameters (including the parameters of the frozen T5-base model) with 6 nested U-Net blocks of increasing channel counts ([128, 256, 512, 512, 1024, 1024]), and again downsampling each time by 2, except for the first block ([1, 2, 2, 2, 2, 2]). We use attention blocks at the depths [0, 0, 1, 1, 1, 1], skipping the first two blocks to allow for further downsampling before sharing information over the entire latent, instead use cross-attention blocks at all resolutions ([1, 1, 1, 1, 1, 1]). For both attention and cross-attention, we use 64 head features and 12 heads per layer. We repeat items with an increasing count towards the inner U-Net low-resolution and large-context blocks ([2, 2, 2, 4, 8, 8]), this allows good structural learning over minutes of audio.

B More Experiments

B.1 Hardware Requirements

We use limited computational resources as available in a university lab. (3) **Efficiency** is another highlight of our model, where we only needs an inference time equivalent to the audio length on a consumer GPU, which is several minutes, while many other text-to-audio models take many GPU hours (Dhariwal et al., 2020; Kreuk et al., 2022). Our model is very friendly for research at university labs, as each of our models can be trained on a single A100 GPU in 1 week of training using a batch size of 32; this is equivalent to around 1M steps for both the diffusion autoencoder and latent generator. For inference, as an example, a novel audio source of ~43s can be synthesized in less than 50s using a consumer GPU with a DDIM sampler and a high step count (100 generation steps and 100 decoding steps).

C More evaluation details

C.1 Annotation Details for the Genre Identification Test

Prompts We list all the text prompts composed for the four common music genres in Table 7.

Using these prompts, we generate music with both Moûsai and the Riffusion model (Forsgren and Martiros, 2022), with a total of 80 pieces of music, two for each prompt.

To validate this quantitatively, we conducted a listener test with three perceivers (annotators) with diverse demographic backgrounds (both female and male, all with at least a Master's degree of education). Each annotator listens to all 80 music samples we provide, and is instructed to categorize each sample into exactly one of the four provided genres.

Annotation We record how many times the perceiver correctly identifies the genre which the respective model was generating from. A large number (or score) means that the model often generated music that, according to the human perceiver, plausibly belonged to the correct category (when compared to the other three categories). To achieve a good score, the model needs to generate diverse and genre-specific music. We take the score as a quality score of the model when it comes to correctly performing text-conditional music generation.

In Figure 5, we display the confusion matrix of this genre identification test for both our model (left) and the Riffusion model (right). For our model, the annotators identify the right genres most of the time, whereas for the Riffusion model, the annotators often perceive the music as more generic, categorizing it as Pop.

C.2 Annotation Details for Turing Test

We let the annotators listen to a pair of music samples at a time, and judge which one is real and which is generated. To provide a more fine-grained score, we also ask them how much the generated music they identified sounds like real music, on a scale of 1 (almost not similar at all) to 5 (highly similar). We keep their annotation score if they identify the generated music correctly, and otherwise we rate the music as 5, which means that the music perfectly passes the Turing test.

As for the details, we create 90 music samples, including 15 generated samples paired with 15 real music samples for each of the three models (Riffusion, Musika, and Moûsai). We recruit two undergraduate annotators who have pursued playing music as a hobby for the past 10 years.

We conducted a rigorous evaluation employing an experiment with a similar spirit to the Turing test (TURING, 1950) for natural language, but commonly called as the fidelity test in audio evaluation (Hyun et al., 2022) or speaker test (Greshler et al., 2021; Hawthorne et al., 2019b) in audio evaluation. Our methodology involved presenting a group of expert annotators with a total of 60 distinct folders, 15 corresponding to each of Mousai, Mousai (classical-only), Riffusion, and Musika models. Each folder containing two music files, one being the original and the other generated using a given model prompted with its corresponding metadata.

The annotators were provided with the task of determining the fidelity and providing a rating on a scale of 1 to 5, reflecting the perceived degree of authenticity of the generated audio. In cases where the annotators incorrectly identified the generated audio, the respective model was awarded 5 points. Conversely, if the annotators correctly identified the generated audio, the model's rating was determined based on the score provided by the annotator. The annotators were compensated with 500 rupees (∼6.5 dollars) for this 3 hour task (which is well above daily minimum wage in India).

Following are the exact instructions provided to the annotators

1. You will be presented with batches of two audio samples in subfolders of this folder named from 1 to 60. Each subfolder contains two audios named a.wav and b.wav.
2. Listen to each sample carefully.
3. It's best to use headphones in a quiet environment if you can.
4. Some files may be loud, so it's recommended to keep the volume moderate.
5. One of the audio samples in each pair is a real recording, while the other is a generated (synthetic) audio.
6. Listen to each pair of audio samples carefully.
7. Pay attention to the quality, characteristics, and nuances of each audio sample.
8. This folder contains a spreadsheet file called 'Response_Task_2.xlsx'. Compare the sam-

ples to each other and provide a relative rating to the fake audio only out of 5, where 1 being the most fake and 5 being most real.

C.3 Annotation Details for Musicality

In order to ascertain the quality and artistic merit of the generated musical output, a rigorous human evaluation methodology was implemented. A total of 50 carefully curated folders, each containing three distinct audio files, were presented to human evaluators. These audio files were generated utilizing various models, all prompted by a specific prompt. We recruit two annotators, pursuing Bachelor of Technology degree from the Indian Institute of Technology, Kharagpur, India. Additionally, the two annotators have pursued playing music as a hobby for the past 10 years. The annotators were compensated with 500 rupees (∼6.5 dollars) for this 3 hour task (which is well above daily minimum wage in India).

Following are the exact instructions provided to the annotators

1. Listen to the music and rate it based on three aspects: Quality, Melody, and Harmony.
2. It's best to use headphones in a quiet environment if you can.
3. Some files may be loud, so it's recommended to keep the volume moderate.
4. This folder contains folders subfolders through 1-50. Each subfolders contains three audio files named A.wav, B.wav, and C.wav . You need to listen to each of them and rate them (relative to each other) based on quality, melody, and harmony.
5. For Quality, consider how clear the audio sounds. Does it resemble a walkie-talkie (bad quality) or a high-quality studio sound system (good quality)?
6. Melodiousness refers to the main pitch or note in the music. Pay attention to the rhythm and repetitiveness of the melody. A more rhythmic and repetitive melody is considered better, while the opposite is true for a less rhythmic melody.
7. Harmoniousness involves multiple notes played together to support the melody. Evaluate if these notes are in sync and enhance the effect of the melody. Higher scores should be given for good harmony and lower for poor harmony.
8. It is recommended view youtube videos: this or this short video explaining melody and harmony
9. This folder also contains a spreadsheet by the name "Response_Task_1.xlsx". Remember to provide ratings (out of 5) for each aspect of your evaluation in the file against appropriate folder number. Feel free to listen to each sample as many times before rating them.

D More Related Work

Audio generation is a challenging task. At the lowest level, we have digital waveforms that control air movement from speakers. Waveforms can be represented in different resolutions, or sample rates. Higher sample rates (e.g., 48kHz) allow for more temporal resolution and can represent higher frequencies, but at the same time it is computationally more demanding to generate. At higher levels of abstraction, we find qualitative properties such as texture (timbre) or pitch. Zooming out, we observe structure such as rhythm and melody that can span multiple seconds, or even structurally be composed into choruses that form minutes of interconnected patterns.

Audio can be represented with a single waveform (mono), two waveforms (stereo), or even more waveforms in the case of surround sound. Audio with two or more channels can give a sense of movement and spatialisation. From a modelling perspective, there are (1) unconditional models that generate novel samples from the training distribution without any additional information, or (2) conditional models that use a form of guidance, such as text, to control the generation. Models can be trained on a single modality (e.g., drums or piano) or on multiple modalities, which usually require more parameters for an increased modelling capacity and decrease in speed.

E Exploring Variations of the Model Architecture and Training Setup

E.1 High-Frequency Sounds

We observe that our model is good at handling low-frequency sounds. From the mel spectrograms Figure 7, and also the music samples we provide, we notice that our model performs well with drum-like sounds as frequently found in electronic, house, dubstep, techno, EDM, and metal music. This is likely a consequence of the lower amount of infor-

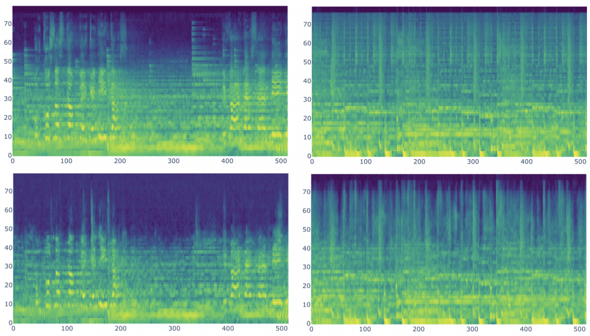


Figure 7: Mel spectrogram comparison between the true samples (top) and the auto-encoded samples (bottom); cf. text.

mation required to represent low-frequency sounds.

E.2 Improving the Structure

We find that increasing the number of attention blocks (e.g., from a total of 4 – 8 to a total of 32+) in the latent diffusion model can improve the general structure of the songs, thanks to the long-context view. If the model is trained without attention blocks, the context provided by the U-Net is not large enough to learn any meaningful long-term structure.

E.3 Text-Audio Binding

We find that the text-audio binding works well with CFG higher than 3.0. Since the model is trained with metadata such as title, album, artist, genre, year, and chunk, the best keywords to control the generation appear to be frequent descriptive names, such as the genre of the music, or descriptions commonly found in titles, such as *"remix"*, *"(Deluxe Edition)"*, and possibly many more. A similar behavior has been observed and exploited in text-to-image models to generate better looking results.

E.4 Trade-Off between Speed and Quality

We find that 10 sampling steps in both stages can be enough to generate reasonable audio. We can achieve improved quality and reduced noise for high-frequency sounds by trading off the speed, i.e., increasing the number of sampling steps in the diffusion decoder, e.g., 50 – 100 steps). Increasing the number of sampling steps in the latent diffusion model (again in the order of 50 – 100 steps) will similarly improve the quality, likely due to the more detailed generated latents, and at the same time result in an overall better structured music. To make sure the results are comparable when varying the number of sampling steps, we use the same starting noise in both stages. In both cases, this suggests that using more advanced samplers could be helpful to improve on the speed-quality trade-off.

E.5 Trade-Off between Compression Ratio and Quality

We find that decreasing the compression ratio of the first stage (e.g., to 32x) can improve the quality of low-frequency sounds, but in turn will slow down the model, as the second stage has to work on higher dimensional data. As proposed later in Section 6, we hypothesize that using perceptually weighted loss functions instead of L2 loss during diffusion could help this trade-off, giving a more balanced importance to high frequency sounds even at high compression ratios.

E.6 High-Frequency Audio Generation

We have encountered challenges in achieving satisfactory results when dealing with high-frequency audio signals, as detailed in Appendix E.1. To gain deeper insights into the underlying issues, we conducted an ablation experiment by exclusively training our model on classical music, a genre known for its prominent high-frequency characteristics. We train this model using 500 hours of music collected from albums of top classical composers[4] and other popular Spotify playlists. We notice a drop of 9.5% in the fidelity score of the generated music samples compared to those produced by our original model. Further, qualitative analysis reveals that melodic elements of these samples demonstrated commendable accuracy, the harmony notes appeared to be convoluted and disorganized. This finding highlights the significance of harmonization challenges when generating high-frequency audio and underscores the need for developing improved models in future research.

[4] (cla)

Genre = Electronic
– Drops, Kanine Remix, Darkzy, Drops Remixes, bass house, (Deluxe) (Remix) 3 of 4
– Electronic, Dance, EDM (Deluxe) (Remix) 3 of 4
– Electro House (Remix), 2023, 3 of 4
– Electro Swing Remix 2030 (Deluxe Edition) 3 of 4
– Future Bass, EDM (Remix) 3 of 4, Remix
– EDM (Deluxe) (Remix) 3 of 4
– EDM, Vocal, Relax, Remix, 2023, 8D Audio
– Hardstyle, Drop, 8D, Remix, High Quality, 2 of 4
– Dubstep Insane Drop Remix (Deluxe Edition), 2 of 4
– Drop, French 79, BPM Artist, Vol. 4, Electronica, 2016
Genre = Hip Hop
– Real Hip Hop, 2012, Lil B, Gods Father, escape room, 3 of 4
– C'est toujours pour ceux qui savent, French Hip Hop, 2018 (Deluxe), 3 of 4
– Dejando Claro, Latin Hip Hop 2022 (Deluxe Edition) 3 of 4
– Latin Hip Hop 2022 (Deluxe Edition) 3 of 4
– Alternative Hip Hop Oh-My, 2016, (Deluxe), 3 of 4
– Es Geht Mir Gut, German Hip Hop, 2016, (Deluxe), 3 of 4
– Italian Hip Hop 2022 (Deluxe Edition) 3 of 4
– RUN, Alternative Hip Hop, 2016, (Deluxe), 3 of 4
– Hip Hop, Rap Battle, 2018 (High Quality) (Deluxe Edition) 3 of 4
– Hip Hop Tech, Bandlez, Hot Pursuit, brostep, 3 of 4
Genre = Metal
– Death Metal, 2012, 3 of 4
– Heavy Death Metal (Deluxe Edition), 3 of 4
– Black Alternative Metal, The Pick of Death (Deluxe), 2006, 3 of 4
– Kill For Metal, Iron Fire, To The Grave, melodic metal, 3 of 4
– Melodic Metal, Iron Dust (Deluxe), 2006, 3 of 4
– Possessed Death Metal Stones (Deluxe), 2006, 3 of 4
– Black Metal Venom, 2006, 3 of 4
– The Heavy Death Metal War (Deluxe), 2006, 3 of 4
– Heavy metal (Deluxe Edition), 3 of 4
– Viking Heavy Death Metal (Deluxe), 2006, 3 of 4
Genre = Pop
– (Everything I Do), I Do It For You, Bryan Adams, The Best Of Me, canadian pop, 3 of 4
– Payphone, Maroon 5, Overexposed, Pop, 2021, 3 of 4
– 24K Magic, Bruno Mars, 24K Magic, dance pop, 3 of 4
– Who Is It, Michael Jackson, Dangerous, Pop (Deluxe), 3 of 4
– Forget Me, Lewis Capaldi, Forget Me, Pop Pop, 2022, 3 of 4
– Pop, Speak Now, Taylor Swift, 2014, (Deluxe), 3 of 4
– Pop Pop, Maroon 5, Overexposed, 2016, 3 of 4
– Pointless, Lewis Capaldi, Pointless, Pop, 2022, 3 of 4
– Saved, Khalid, American Teen, Pop, 2022, 3 of 4
– Deja vu, Fearless, Pop, 2020, (Deluxe), 3 of 4

Table 7: Text prompts composed for the four common music genres: electronic, hip hop, metal, and pop.