CMU F23 10-701 FINAL PROJECT REPORT: A SURVEY OF RECENT ML-BASED MUSIC GENERATION METHODS

ANDREW ZIGERELLI, QIXIAO ZHU, RIKKI HUNG, AND SHIH-LUN WU
{anz37, qixiaoz, rikkih, shihlunw}@andrew.cmu.edu

Abstract. Automatic music generation has received renewed research interests in the era of deep learning. It is now a burgeoning field with a variety of task formulations and model constructions. In this survey paper, we put our emphasis on the most recently proposed methods (i.e., those proposed in 2023). We strive to first dissect the vital machine learning-based components shared across methods, and then explain how each method integrated such components. We also cover the commonly adopted evaluation metrics, and propose several new metrics focusing on musical controllability, which the existing metrics largely neglected. We build inference and evaluation pipelines for two popular methods: AudioLDM and MusicGen, and conduct experiments that quantify the benefit of choice users enjoy with multiple generations under the same input, and reveal the current models' lack of musical controllability. Our inference and evaluation pipelines can be found at: `colab.research.google.com/drive/1PY3wjQQIoQOSDrJTMNfNqRqt1y-8BjJA?usp=sharing`.

1. Introduction

Music is a form of art that enables humankind to express emotions and tell stories beyond the barrier of languages. Fundamental to the enchantment of music are its meticulous structures [23] both on the frequency axis, e.g., harmony and instrumentation, and the time axis, e.g., development of musical themes/ideas, which partly explains why music composition takes both great creativity and a tremendous amount of time. As an attempt to understand the human creative process of music, the first system to algorithmically compose music was realized in the 1950s [40], using explicitly programmed control branches, random numbers, and markov chains [64]. Subsequently, before the era of machine learning (ML) and deep learning, researchers approached music generation primarily using formal grammars [43, 65, 79], writing music theory and the composers' knowledge into transition rules, similar to how linguists analyze languages [18].

As deep learning became the mainstream of artificial intelligence (AI) research in the 2010s thanks to the ImageNet challenge and surrounding work [58, 77], music generation research also shifted to neural network-based approaches [13], which are much less rigid than formal grammars and produce more compelling outputs. Early exemplary works include MelodyRNN [90] and WaveNet [86], which, respectively, generates music in the symbolic (i.e., as a stream of notes) and audio domains. Recently, with the advent of more powerful generative models, e.g., Transformers [89] and diffusion models [41], and representation learning techniques for audio [24, 88, 101] and text [47, 71, 95], text-conditioned audio-domain music generation models gained high popularity in the year of 2023 [9, 15, 22, 62]. Comparing the field's progress between now (i.e., 2023) and 7 years ago, we have advanced from generating simple melodies [90] or performances for a specific instrument [86], to creating music with broad styles and instrumentations using straightforward free-form text conditions [9, 22, 62]. The latest literature reviews on music generation are either becoming outdated [13, 50] due to fast advancement of the field, or, focused on symbolic-domain approaches [51], which left out the majority of the most recent research works. Therefore, an updated survey is warranted.

In this paper, we begin by setting the our survey scope and explaining major aspects of music generation models (Sec. 2). As the field has not yet standardized on task formulation and model design, we detail the key technical components (Sec. 3), introduce how recent models brought the components together (Sec. 4), and cover the mainstream evaluation metrics along with new metrics focused on the neglected aspect of controllability for musical attributes (Sec. 5). We then conduct experiments (Sec. 6) examining hyperparameter sensitivity, compute efficiency, output quality, and controllability of two representative works (AudioLDM [62] and MusicGen [22]), and conclude our survey with a brief remark on future directions of the field (Sec. 7).

Date: December 9, 2023.
CMU F23 10-701 Final Project Report: A Survey of Recent ML-based Music Generation Methods

Table 1. Summary of representative music generation systems discussed in our survey. The output music formats are: (i) **sound waveform** (16∼44 kHz sampling rate) for **audio**-domain systems, (ii) **music notes** (with time, duration, and instrument information) for **symbolic–token seq** systems, and (iii) **piano roll** [30] for **symbolic–image** systems. (Important abbreviations: *wave*: sound waveform, *spec*: spectrogram, *Trans./Transfmr.*: Transformer, *tok.*: tokens, LM: language modeling.)

Domain–Modality	System	Year	Core module	Core training	Peripheral modules (function)	Peripheral training	Conditions allowed
Audio–Token seq	*Jukebox* [26]	'20	Transformer	LM	CNN (audio tok.↔wave) [73]	VQVAE	• artist/genre • lyrics
	MusicLM [9]	'23			• CNN (audio tok.↔wave) [101] • Transfmr. (semantic tok.↔wave) [20] • Transfmr. (text-audio aligned emb.) [47]	• RVQ+GAN (audio tok.↔wave) • MLM+InfoNCE (semantic tok.↔wave) • InfoNCE (text-audio aligned emb.)	• free text • melody
	MusicGen [22]	'23			• CNN+Transfmr. (audio tok.↔wave) [24] • Transfmr. (text emb.) [71]	• RVQ+GAN+LM (audio tok.↔wave) • T5 pretraining (text emb.)	• free text • melody
Audio–Image/Wave	*Noise2Music* [48]	'23	CNN	diffusion	Transfmr. (text emb.) [71]	T5 pretraining	free text
	AudioLDM [62]	'23			• CNN (spec↔wave) [56] • CNN (latents↔spec) [55] • Transfmr. (text-audio aligned emb.) [95]	• GAN (spec↔wave) • VAE (latents↔spec) • InfoNCE (text-audio aligned emb.)	• free text • melody • left/right context
	Music-ControlNet [92]	'23			CNN (spec↔wave) [57]	diffusion	• mood/genre • melody • dynamics • (down)beats
Symbolic–Token seq	*Music-Transfmr.* [46]	'18	Transformer	LM	MIDI-like tokenization [68]	n.a.	left context
	Compound-Word Trans. [44]	'21			Compound-Word tokenization	n.a.	• left context • melody
	Anticipatory-Mus. Trans. [84]	'23			MIDI-like tokenization [68]	n.a	any music fragments (w/ time, pitch & instrument info)
Symbolic–Image	*MuseGAN* [29]	'17	CNN	GAN	n.a.	n.a.	complete instrument tracks
	Mittal+ [67]	'21	Transformer	diffusion	RNN (latents↔notes) [74]	VAE	left/right context

2. Survey Methodology

2.1. Survey Scope. We restrict the scope to only cover methods that can generate music under *weak* conditions, i.e., conditions that are much less informative and/or much easier to create than the output music. For example, simple text descriptions of music and melody lines both qualify as weak conditions. According to the principle above, we exclude: (i) methods that are conditioned on full lyrics [99] or video [27, 32], (ii) music style transfer works that require reference music as an input [14, 21, 94], and (iii) score-to-audio synthesis works that complete expressive performance aspects for written sheet music [35, 91, 96]. Moreover, we will not cover common datasets used to train music generation systems, since they have been introduced in detail in a previous survey paper [50], and latest works mostly trained their models on proprietary datasets [9, 22, 92].

2.2. Categorization. In our view, ML-based music generation models differ from each other fundamentally in three aspects: (i) the **output domain**, (ii) the underlying **ML formulation/modality**, and (iii) the **input conditions** through which users influence the generated music. In the rest of this paper, we will remind readers about which categories, according to the three aspects above, a component (or system) belongs to when it is introduced. Table 1 provides a summary of all representative works (Sec. 4) covered in our survey, organized by the before-mentioned aspects and detailing additional peripheral techniques (Sec. 3).

2.2.1. *Output Domain.* Humans represent, create, and understand music in various ways. Composers typically write music into musical scores, which can be digitally represented using Musical Instrument Digital Interface (MIDI) [8]. MIDI decomposes a score into instrument, note on/off, time shift, and other integer-indexed *events*. Meanwhile, in scenarios like jamming sessions, performers improvise and record music on the spot, where sound is produced directly. Following the two ways musicians convey musical content, there are two mainstream output domains for music generation models: **audio** and **symbolic**. The two domains each come with their own merits. **Audio** models produce waveforms [9, 22] or spectrograms [35, 62], which display the instantaneous energy levels of various frequency bands on a 2D image. Audio-domain outputs include acoustic details, e.g., micro-timing and vibrato, which are crucial to musical expressiveness. **Symbolic-**domain models typically generate music in a format that can be easily converted into musical scores—for example, a sequence of MIDI events [46, 84], beat-synchronized events [49], or a piano roll [29, 67], which is an image where the axes indicate time and pitch, and notes are drawn as lines. Symbolic form is more amenable to iterative editing, which is central to human-AI co-creation setups [45].

2.2.2. *ML Formulation/Modality.* Somewhat orthogonal to the output domain, i.e. how music is presented to users, is the underlying ML problem formulation and modalities of internal representations. This aspect can also be classified as two large families: **token sequence** modeling (i.e., in *discrete* space) or **image/waveform** modeling (i.e., in *continuous* space). Readers might think that all audio-domain methods employ image/waveform modeling, while all symbolic-domain methods use token sequence modeling, but in fact, there exist methods that fall in all four possible combinations of output domain and ML modality.

Token sequence (or **token seq** in short) modeling approaches were derived from Natural Language Processing (NLP)—a piece of music is represented by a sequence of events (or tokens) $X = (x_1, x_2, \ldots, x_N)$ with its likelihood maximized by $p(X|\boldsymbol{c})$ where $\boldsymbol{c}$ are user-controlled conditions. **Image/waveform** modeling approaches, meanwhile, mainly descended from generative models for Computer Vision (CV), e.g., Generative Adversarial Networks (GAN) [34] and Denoising Diffusion Probabilistic Models (DDPM, or simply diffusion models) [41]. In these methods, music is represented as an image (i.e., spectrogram or pianoroll) $X \in \mathbb{R}^{W \times H \times D}$ [29, 62, 67], where W, H, D are respectively the height, width, and number of channels, or waveform $X \in \mathbb{R}^{Tf_s \times D}$ [48], where T is the duration in seconds, and f_s is the sampling rate. GAN or diffusion models then generate music via $X = f(\boldsymbol{c}, \boldsymbol{\epsilon})$ where f is a functional mapping learned by the model via estimating $p(X|\boldsymbol{c})$, and $\boldsymbol{\epsilon}$ is some random noise injected to ensure sample diversity.

2.2.3. *Input Conditions.* From an application point of view, the conditions $\boldsymbol{c}$, which the model $p(X|\boldsymbol{c})$ affords, are arguably the most crucial aspect as they serve as the means for users to steer their creative processes. Input conditions can roughly be categorized as **global** or **local** (i.e., time-varying) by nature. **Global** conditions include, for example, musical genre [26], tempo [49], instrumentation [29], and text description [9, 22, 62], which is popular recently and may contain multiple attributes. Meanwhile, examples of **local** conditions are chord progression [97], melody [22, 84], local tempo changes [49], dynamics [92], and rhythm [92]. In this paper, we will focus on how different types of conditions can be injected into the model.

2.3. Distinctions from Previous Surveys. Previous literature reviews [13, 50, 51], which are much longer in length, either put similar weights on ML and non-ML parts (e.g., output music formats [13] and datasets [50, 51]), or covered a wider range of tasks [13, 50] (i.e., the ones mentioned in Sec. 2.1) and the full chronological timeline of methods [50]. To be complementary to the surveys above, we put a substantial emphasis on recently proposed methods (i.e., those proposed between the last comprehensive survey [50] and ours) and relevant ML techniques that are impactful across application fields (e.g., CV and NLP). Furthermore, we implement a notebook containing inference and evaluation pipelines for representative recent models [22, 62], which is meant to familiarize general users/creators with ML-based music generation models swiftly.

3. Key Technical Components

3.1. Domain-specific Internal Representations. The representations of music inside ML models have their root in traditional sound industry practices of audio codecs (e.g., *AAC* and *MP3*). During transmission, recordings are compressed using audio codecs (i.e., digital signal representations with a lower bit rate) for bandwidth purposes, which come with a codebook where the receiving end can reconstruct the original audio. Traditionally, such codebooks were engineered with domain-knowledge, but with deep learning, researchers have trained high-performing neural audio codecs using lossy reconstruction [24, 88, 101].

Vector Quantization. (For **audio–token sequence** systems.) Recognizing the inherently discrete nature of sequence modeling tasks and aiming to adapt strong sequence models [89] to continuous data, Vector-quatized variational autoencoders (VQVAE) [88] presented a reparameterization trick to convert continuous latents to discrete codewords, combining ideas of autoencoding [55], K-means centroids [63] and straight-through gradient estimation [10]. VQVAE demonstrated high-fidelity reconstruction, on par with continuous counterparts [55] on image and speech related tasks. VQVAE-2 [73] trained multiple VQVAEs in parallel to focus on different resolutions. Soundstream [101] specialized the technique to the domain of audios and introduced a new technique, residual vector quantization (RVQ), inspired by residual connections, which cascadingly and iteratively encodes an audio sequence using a series of codebooks targeted at different granularities. A discriminator [60] and adversarial loss [34] was also added to enhance the realisticness of reconstructed audio. Encodec [24] improved upon Soundstream by adding language modeling (LM) [89] as an auxiliary loss on the encoded codewords, which further reduced the bandwidth without harming fidelity.

Symbolic Music Tokenization. (For **symbolic–token sequence** systems.) For symbolic inputs, tokenization techniques were developed to deterministically serialize musical notes to enable language modeling. [68] proposed a scheme that mimics MIDI events—NOTE-ON & NOTE-OFF tokens marks the onset and release of a note (with a specific pitch), TIME-SHIFT tokens track the progress of time (in multiples of 10 msec), and VELOCITY tokens specify the loudness for expressive dynamics. [49] introduced a beat-synchronized scheme that used BEAT, BAR, and NOTE-DURATION tokens to replace TIME-SHIFT's, which promoted a clearer sense of rhythm in the generations. [44] proposed the *compound-word* scheme that groups either note-related, or time-related tokens into one compound token, reducing the required sequence length by more than 50%.

3.2. Core ML Modules. Following the success of language models in NLP [70, 82, 89], or GAN [34] and diffusion [41] in vision, researchers also leveraged these backbones and training objectives to model music.

Backbone Architectures. For all recent systems using the **token sequence** modeling formulation, Transformers [89] are the default backbone thanks to its self-attention mechanism that facilitates reference to the full previous musical context. For systems with the **image/waveform** formulation, residual convolutional neural networks (CNN) [37] have been the standard. A commonly used variation is the UNet [76], which inherently promotes hierarchical feature encoding and reconstruction.

Training Objectives. Let the desired music output be X, **token sequence** models explicitly approximate and maximize the data likelihood by minimizing $-\log p(X \mid \boldsymbol{c}) = -\sum_{i=1}^{N} \log p(x_i | X_{1:i-1}, \boldsymbol{c})$, i.e., the negative log-likelihood. On the other hand, **image/waveform** models implicitly approach $p(X\,|\boldsymbol{c})$ either using adversarial losses [34] or the denoising diffusion process [41]. Adversarial losses train a discriminator to recognize true or generated data, and a generator to 'fool' the discriminator by drawing the true and generated data distributions close. The denoising diffusion process first injects random noise into the clean data to construct latent features, and requires the model to recover the injected noise, which provably maximizes the lower bound of $p(X\,|\boldsymbol{c})$ (more specifically, the *evidence lower bound* [55], or ELBO) under the Markov chain assumption over latent features with different noise levels.

3.3. Conditioning Signals and Mechanisms. For more straightforward types of **global** conditions like artist [26], genre [26, 92], or mood [92], we can learn embeddings and infuse such information either as prefix to the language model [26] or via cross-attention [92]. For **local** conditions of the same complexity/richness as the desired output, autoregressive models [26, 44, 46] can inherently receive the left context as the 'primer', and generate the continuation, while non-autoregressive models (e.g., diffusion [62]) can naturally accept both left and right contexts, and perform an 'infilling' task. Below, we explain in more detail how (i) **global** conditioning via free-form text (which may contain more information than mood, genre, and artist), and (ii) **local** conditions that are weaker (e.g., melody or chords) than the desired output music can be implemented.

Audio-Text Contrastive Learning. The practical value of contrastive learning, i.e., learning a latent space that aligns conceptually similar data via the InfoNCE loss [87], was first realized in the visual domain [16, 36]. Later, this technique was extended to multimodal inputs, e.g. text & image [69] and text & audio/music [47, 95]. For the latter case, this is done by separately encoding text and audio inputs with unimodal deep encoders, and then applying InfoNCE loss to draw paired encoded representations close while pushing unpaired ones further away. Having an aligned text-audio embedding space gives a unique advantage: we can train text-conditioned music generation models (i.e., text-to-music models) using audio embeddings without paired text descriptions, then input text conditions/embeddings at inference.

Interleaving Conditions with Generation. Interleaving is a simple technique that provides strong hints on the 'locality' of local conditions. For example, in Compound Word Transformer [44], the input sequence for melody-conditioned generation is formatted as [`(melody of bar #1), (full music of bar #1), (melody of bar #2), (full music of bar #2), ...`], which always keeps the immediate conditions aside when generating the full music. In Anticipatory Music Transformer [84], the interleaving scheme was further relaxed—conditioning information of any instrument, pitch, and onset time (i.e., that can be represented as tokens), is interleaved into the sequence δ seconds ahead (in wall-clock time) of when it should appear.

4. Integrated Systems

4.1. Audio–Token Sequence Models. An early work under this category is Jukebox [26], which leveraged VQVAE [73, 87] to discretize audio waveform into tokens, and then applied Transformers [89] to autoregressively model the audio tokens conditioned on artists, music genres (as prefix embeddings), or lyrics (via cross attention). The generated tokens are then decoded as waveform by the VQ-VAE.

MusicLM [9] and MusicGen [22] are both representative follow-up works that used Transformers to model audio tokens, but took different approaches to (i) incorporate the text conditions, and (ii) factorize the generation of audio tokens. MusicLM [9] extended AudioLM [12], an audio-to-audio sequence model with text conditioning, to the text-to-audio/music setting. Previously, AudioLM achieved music of high fidelity and long term coherence via representing music in its modeling process as both acoustic tokens and semantic tokens. Acoustic tokens, learned by SoundStream [101], compress music (as an audio waveform) to a compact and discrete representation though residual vector quantization (RVQ), while semantic tokens, learned by the w2v-BERT model [20], map audio to higher-level semantic features via contrastive learning [87] and masked token prediction [25]. In MusicLM, text descriptions are not required at training. At inference, given a text input, which would be transformed into text-audio aligned MuLan [47] embeddings, one transformer would generate semantic tokens, and then, conditioned on the semantic tokens, a second Transformer would generate the acoustic tokens, which are finally convered to audible waveform through the SoundStream decoder.

MusicGen [22] was trained on a dataset with paired audio and text. It used the EnCodec RVQ [24] to compress audio waveform to acoustic tokens. The text input was converted to embeddings by the T5 large language model [71], which are to be cross-attended by the acoustic token generation model. Then, aiming for efficiency and simplicity, a single Transformer model was used to generate all the acoustic tokens following a delay pattern [53], which allows the model to iteratively refine the details by generating increasing finer-grained acoustic tokens (for the same instant in the audio) in several time steps. MusicGen also accepts melody conditions in the chromagram [83] format.

4.2. Audio–Image/Waveform Models. This line of research was primarily motivated by the success of diffusion models [41, 81] in image generation [72, 75, 78]. Noise2Music [48] proposed to generate either a Mel spectrogram, or a low-resolution (3.2 kHz) audio waveform conditioned on text input, using a UNet [76] diffusion backbone. From the intermediate output, a cascade of UNet diffusion models were used to convert/upsample it iteratively to the final 24-kHz waveform. To address the scarcity of paired audio-text data, Noise2Music leveraged a large language model, LaMDA [85], to rewrite the more abundant artist and song titles into free-form text descriptions. The text condition then enters the diffusion model as text embeddings [71] via cross-attention, which is similar to [22].

Another representative work is AudioLDM [62], which was based on latent diffusion [75]. Architecture-wise, AudioLDM first used a variational autoencoder (VAE) [55] to compress time slices in a Mel spectrogram into a sequence real-valued vectors, or latent features. A U-Net diffusion backbone generates the VAE latent features instead, which are then decoded back to a Mel spectrogram. The last step of converting a Mel spectrogram to audio waveform was done by the HiFi-GAN vocoder [56], which promotes the realisticness of the final waveform output via multiple adversarial losses [34]. The 3 model components, i.e., vocoder, VAE, and latent diffusion, are trained separately. Similar to MusicLM [9], AudioLDM utilized text-audio aligned embeddings [95] to eliminate the need of paired text during training.

Music ControlNet [92] adapted local pixel-wise controls for diffusion-based image generation [102, 103] to time-varying controls for audio/music. The proposed framework opened the door for any control signals that can be extracted from audio using signal processing [66] or music information retrieval (MIR) [17] techniques. Simultaneous control over melody, rhythm [11], and dynamics was realized in Music ControlNet.

4.3. Symbolic–Token Sequence Models. Symbolic-domain outputs have the advantage of being more editable, but would rely on a downstream neural synthesizer or vocoder [35, 96] (which of out of our survey scope) to sound more expressive. Music Transformer [46] employed the MIDI-like tokenization in [68] and used a single Transformer for autoregressive language modeling. It proposed to use relative positional attention [80], as opposed to the original absolute positional attention [89], as an individual musical note relies heavily on context, which can be the relative time, pitch, and dynamics to other notes. Music Transformer does not afford users to use text descriptions as the condition, but instead allows them to input a short 'primer' musical excerpt for the model to generate a continuation. Compound Word Transformer [44] utilized a memory-efficient Transformer [52] on top of its condensed tokens, which made fitting into memory the full-song token sequences possible, and enhanced the long-range musical structure in its generations.

Anticipatory Music Transformer [84] improved upon [46] and [44] both in the flexibility of conditions and the conditioning mechanism. In addition to primer or melody conditions, [84] allowed conditions to be concerned with any time span in the generation, and to be in various forms, including chords, melody, and drum patterns as these can all be represented by a combination of tokens in the (slightly extended) tokenization scheme of [68]. On conditioning mechanism, [84] proposed to insert the condition tokens when the wall-clock time is precisely an amount, δ seconds, earlier than when the conditions should appear in the music. This method informs the model of future musical context and preserves the locality, which collectively guides the model to smoothly transition/adapt to the upcoming conditions.

4.4. Symbolic–Image Models. MidiNet [97] and MuseGAN [29], both based on GANs [34], are seminal works under this family, with the latter expanding the capability to generating multi-instrument piano rolls. [67] proposed to apply diffusion [41] to model the latents of MusicVAE [74] instead, which can then be bar-wise decoded into note sequences. This family of approaches has recently declined in popularity, as the output duration is restricted by the image width, and the expressiveness is worse than audio-domain models.

5. Evaluation Metrics

5.1. Objective Metrics for Audio Domain. Fréchet Audio Distance (**FAD**) [54], derived from Fréchet Inception Distance (FID) [39] for image generation tasks, examines the overall **realisticness** (including quality and diversity) by computing the Fréchet distance between two estimated Gaussians of encoded audio features [33] for a pair of generated and real audio. Meanwhile, **CLAP**, inspired by CLIPScore [38, 69] for joint image-text tasks, evaluates the generated audio's **adherence to input text** by computing the consine simularity between embeddings from audio and text encoders. As opposed to FAD's set-wise computation, CLAP score is computed on individual generations.

5.2. Objective Metrics for Symbolic Domain. Aside from the most commonly used perplexity [5, 46, 84] for language models, researchers have proposed metrics that measure the distributional discrepancy between musical attributes of generated symbolic music and human compositions [93, 98]. Some notable attributes are: pitch classes histogram, onset counts/intervals, chord n-grams, and repetitive structures. However, these metrics were not shown to have high correlation with human judgement as did FAD ($r = 0.52$ [54]) or CLAP score ($r = 0.51$ [38] for CLIPScore on image captioning).

5.3. Towards Better Focus on Musical Controllability. The CLAP score provides effective evaluation of the overall relevance between the prompt and the generated audio. However, more musically-focused users are likely interested in finer-grained musical controllability of the models. Hence, in addition to the chroma cosine similarity introduced in [22] to evaluate **melody** control, we propose three new metrics evaluating the **tonality** (i.e., key), **dynamics**, and **tempo** controls respectively. These metrics make use of established signal processing or music information retrieval (MIR) tools:

- **Tonality:** Krumhansl-Schmuckler key matching [59], which classifies music into one of 24 majors/minors.
- **Dynamics:** Average root-mean-square (RMS) energy, computed across short windows of audio signal.
- **Tempo:** Audio feature-based beat tracker [100], with which the beats per minute (BPM) can be estimated.

For tonality, we compute the accuracy (i.e., whether the estimated key matches the input key). For dynamics, we take the Spearman's correlation between input dynamics values/levels, and output RMS energy values. For tempo, we measure the mean absolute error (MAE) between the input BPM and that estimated from the generation. Note that, in a similar spirit to holistic evaluation for LMs [61], these metrics can be computed regardless of the mechanisms used to realize the controls. For text-to-music models [9, 16, 62], such controls may be imposed via: "`Generate music in A minor` (key), `forte` (dynamics), `and 120 bpm` (tempo)".

5.4. Subjective Evaluation. As the artistic beauty of music depends on the perception of human listeners, most music generation works [22, 48, 84] conducted listening studies to complement objective evaluation. Listeners are typically required to either (i) compare generations pairwise, in which case the results are presented as win-loss counts [48, 84], or (ii) independently rate each generation on a 5- or 100-point scale, from which the mean opinion score (MOS) can be computed [22]. Common questions asked include overall

TABLE 2. Results for Experiment I—benchmarking AudioLDM [62] and MusicGen [22] (see Sec. 6.3 for setup). 500 text prompts sampled from MusicCaps [9] are used as the input condition. (*ddim*: # of denoising steps; *RTF*: real-time factor, i.e., how many seconds of audio is generated in 1 second of wall-clock time.)

Hyperparameters			Metrics			
			FAD↓	**CLAP↑**	**Time per batch (stdev)↓**	**RTF↑**
AudioLDM (650M params)	ddim=10	guidance=2.5	10.12	.25	**6.26** (0.82)	**15.98**
	ddim=25	guidance=2.5	6.14	.31	11.87 (2.09)	8.42
	ddim=50	guidance=2.5	4.73	.34	13.61 (1.35)	7.34
	ddim=100	guidance=2.5	4.43	.34	43.33 (2.38)	2.31
	ddim=100	guidance=1	6.08	.31	42.28 (1.24)	2.37
	ddim=100	guidance=2.5	4.43	.34	43.33 (2.38)	2.31
	ddim=100	guidance=5	**4.03**	**.35**	43.26 (1.28)	2.31
	ddim=100	guidance=10	4.12	.34	42.89 (1.31)	2.33
MusicGen (1.5B params)		top-k=25	4.60	.27	68.32 (2.10)	.73
		top-k=250	4.39	.29	71.74 (1.44)	.70
		top-k=1250	4.82	.29	70.43 (1.34)	.71

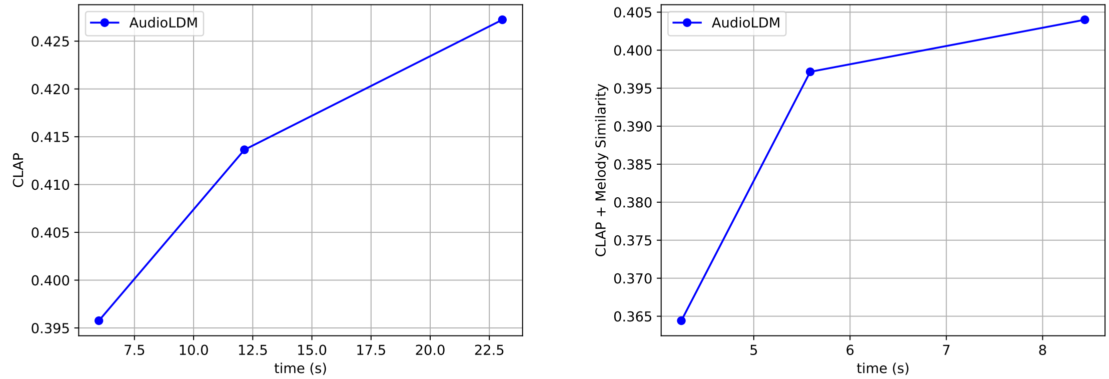

FIGURE 1. Results for Experiment II—quantifying the benefit of choice (see Sec. 6.4 for details). Using AudioLDM, we generate a batch of {2, 5, 10} outputs under the same text condition (**left**), or text+melody conditions (**right**). With larger batch sizes, we can observe sizeable improvements on the maximum CLAP (left) or CLAP+melody control (right) scores among samples in a batch, with a tradeoff on inference time.

TABLE 3. Results for Experiment III—examining musical controllability of text-to-music models (see Sec. 6.5 for setup). The musical controllability metrics (see Sec. 5.3 for details) are the last column in each column group. In general, both models perform poorly on adhering to musical controls in the text input.

		Control	**Dynamics** (in text)			**Tonality** (in text)			**Tempo** (in text)			**Melody**		
			FAD↓	**CLAP↑**	**Corr↑**	**FAD↓**	**CLAP↑**	**Acc↑**	**FAD↓**	**CLAP↑**	**MAE↓**	**FAD↓**	**CLAP↑**	**CosSim↑**
AudioLDM (650M params)	ddim= 25	guidance= 2.5	11.81	0.21	-0.13	10.39	0.19	0.05	10.62	0.21	178.39	17.92	0.12	0.96
	ddim= 25	guidance= 10	11.75	0.27	-0.22	9.72	0.23	0.04	9.36	0.24	177.49	10.96	0.27	0.85
	ddim= 50	guidance= 2.5	11.08	0.25	-0.12	9.39	0.23	0.05	9.91	0.25	177.13	15.73	0.15	0.93
	ddim= 50	guidance= 10	11.11	0.27	-0.13	9.20	0.25	**0.06**	8.99	0.25	177.23	10.69	0.30	0.79
MusicGen (1.5B params)		top-k= 50	**8.57**	**0.32**	-0.10	9.56	0.30	0.05	8.27	0.30	174.26	**8.67**	0.33	**0.98**
		top-k= 1250	9.25	**0.32**	**0.00**	**8.99**	**0.33**	0.05	**8.14**	**0.33**	**173.34**	8.87	**0.36**	0.95

quality (preference) and relevance to input controls, which mirror FAD and CLAP scores, and more musically abstract ones like coherence and richness [44, 93], which are difficult to computationally measure.

6. Implementation and Experiments

6.1. Inference Pipeline. We build an inference API for two representative text-to-music models, AudioLDM [62] and MusicGen [22], with publicly released repositories [1, 2]. For convenience, we built a wrapper class `MusicModel` to expose a unified interface that abstracts away individual calling procedures for the two models. Our API outputs a batch of generated waveforms (with the batch size specifiable), and supports two major conditioning setups: (i) **text input** only, and (ii) **text+melody inputs** (with melody being an audio file). The two models come with different tunable hyperparameters as their ML formulation differ. For AudioLDM, these are (i) the number of DDIM [81] denoising steps and (ii) the classifier-free guidance scale [42]. The former controls how many intermediate steps to take to refine pure random noise into music. We expect an increase in the generation quality with higher number of denoising steps. The guidance scale dictates the strength of the text condition. Larger values promote the generation's adherence to text input at the cost of diversity [42]. For MusicGen, we experiment with different *top-k* [31] values during autoregressive sampling, i.e., truncating the distribution to only consider the top-k most probable tokens. Similar to guidance scale, a higher top-k induces better diversity, while potentially harming relevance to text or melody controls.

6.2. Evaluation Pipeline. We leverage open-source packages to implement 6 metrics in total, including FAD, CLAP, and all four musical controllability metrics introduced in Sec. 5.3. The packages used are:

- **FAD:** `frechet-audio-distance` [4] with the VGGish [33] backbone model.
- **CLAP score:** `laion-clap` [6] with the checkpoint trained on music data.
- **Dynamics control:** `feature.rms()` function in `librosa` [66].
- **Tonality control:** `Tonal_Fragment` class in `musical-key-finder` [7].
- **Tempo control:** `RhythmExtractor2013` class in `essentia` [3].
- **Melody control:** `ChromaCosineSimilarityMetric` class in `audiocraft` [1].

6.3. Experiment I—Benchmarking AudioLDM and MusicGen. We explore the hyperparameter space for both AudioLDM and MusicGen. Besides FAD and CLAP scores, we also record the inference time to understand the quality-time tradeoff. Due to resource constraints, we sample 500 text prompts from the MusicCaps dataset [9] used in evaluating [9, 22, 95], and generate one 10-second audio for each prompt. The results are shown in Table 2. The overall best performer is AudioLDM with 100 denoising steps and a guidance scale of 5. The trend suggests that higher guidance helps with AudioLDM. Meanwhile, MusicGen is much slower, likely due to its larger size and autoregressive nature, without a clear advantage on quality.

6.4. Experiment II—Quantifying the Benefit of Choice. We are interested in how much users can benefit from generating a batch of outputs with the same input condition(s), and then selecting the best one. This has not been discussed before, but from a practical point of view, this strategy may greatly improve user experience, with little overhead thanks to parallel computation. As an exploration, we ask AudioLDM to generate batches of {2, 5, 10} under the same (i) text condition, or (ii) text+melody conditions, and obtain the maximum (i) CLAP or (ii) "CLAP+melody cosine similarity" scores among the batch as a proxy for choosing the best output. The relationship between batch size and such maximum scores is depicted in Fig. 1, which shows that larger batch sizes indeed improve the metrics (by 5% to 10% relatively), with a sub-linear increase in computation time.

6.5. Experiment III—Examining Musical Controllability. In this experiment, we attempt to condition the models with dynamics, tonality, tempo, and melody inputs. Except for the melody condition, which is natively supported by our inference API, we write the musical controls into the text prompt in this format: "`[mood] [genre] music in [musical control]`". We create 6 dynamics (`pianissimo` to `fortissimo`), 24 keys (`C, C#, ..., B major/minor`), 20 tempos (`60∼440 bpm`), and 20 melodies to be cross-producted with 20 (mood, genre) pairs, which leads to 120∼480 test examples for each control. The results are displayed in Table 3. Overall, the models follow the melody control well ($\geq$0.8 chroma cosine similarity). However, the near-zero dynamics correlation, near-random tonality accuracy ($\leq$6%), and large tempo MAE (>170) all suggest that enforcing those controls as part of the text prompt does not work.

7. Conclusion and Future Directions

In this paper, we surveyed recent deep learning-based methods for music generation under weak conditions. We discussed the crucial technical components, and described how those components are assembled into complete systems. Furthermore, we implemented the inference and evaluation pipelines for two representative models, AudioLDM [62] and MusicGen [22], which we then used to conduct experiments that benchmarked the two models, quantified the benefits of multiple generations under the same condition, and revealed the models' lack of musical controllability. Future research may leverage, for example, instruction tuning [19], to enhance the musical controllability in a data-efficient way. Another important direction is to continue improving the long-range musical structure and development [23], which can then be paired with intuitive control mechanisms [28] to vastly streamline our music composition workflow.

References

[1] Audiocraft, official repository containing MusicGen. `https://github.com/facebookresearch/audiocraft/tree/main`, 2023.

[2] AudioLDM official repository. `https://github.com/haoheliu/AudioLDM`, 2023.

[3] Essentia audio analysis package. `https://github.com/MTG/essentia`, 2023.

[4] Fréchet audio distance package. `https://pypi.org/project/frechet-audio-distance/`, 2023.

[5] HuggingFace blog: perplexity for language models. `https://huggingface.co/docs/transformers/perplexity`, 2023.

[6] Laion ai clap package. `https://github.com/LAION-AI/CLAP?tab=readme-ov-file#pretrained-models`, 2023.

[7] Musical-key-finder package. `https://github.com/jackmcarthur/musical-key-finder`, 2023.

[8] Official midi specifications. `https://www.midi.org/specifications`, 2023.

[9] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, et al. MusicLM: Generating music from text. *arXiv:2301.11325*, 2023.

[10] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[11] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. Madmom: A new python audio and music signal processing library. In *ACM International Conference on Multimedia*, 2016.

[12] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, et al. AudioLM: a language modeling approach to audio generation. *IEEE/ACM Trans. on Audio, Speech, and Language Processing (T-ASLP)*, 2023.

[13] J.-P. Briot, G. Hadjeres, and F.-D. Pachet. Deep learning techniques for music generation–a survey. *arXiv preprint arXiv:1709.01620*, 2017.

[14] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer. MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer. In *ISMIR*, 2018.

[15] K. Chen, Y. Wu, H. Liu, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov. MusicLDM: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies. *arXiv:2308.01546*, 2023.

[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[17] K. Choi, G. Fazekas, K. Cho, and M. Sandler. A tutorial on deep learning for music information retrieval. *arXiv preprint arXiv:1709.04396*, 2017.

[18] N. Chomsky. On certain formal properties of grammars. *Information and control*, 1959.

[19] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[20] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu. W2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021.

[21] O. Cífka, U. Şimşekli, and G. Richard. Groove2groove: One-shot music style transfer with supervision from synthetic data. *IEEE/ACM Trans. on Audio, Speech, and Language Processing (T-ASLP)*, 2020.

[22] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez. Simple and controllable music generation. *arXiv:2306.05284*, 2023.

[23] S. Dai, H. Yu, and R. B. Dannenberg. What is missing in deep music generation? a study of repetition and structure in popular music. In *ISMIR*, 2022.

[24] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[26] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever. Jukebox: A generative model for music. *arXiv:2005.00341*, 2020.

[27] S. Di, Z. Jiang, S. Liu, Z. Wang, L. Zhu, Z. He, H. Liu, and S. Yan. Video background music generation with controllable music transformer. In *ACM MM Conference*, 2021.

[28] C. Donahue, I. Simon, and S. Dieleman. Piano genie. In *International Conference on Intelligent User Interfaces (IUI)*, 2019.

[29] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang. MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. *arXiv preprint arXiv:1709.06298*, 2017.

[30] H.-W. Dong, W.-Y. Hsiao, and Y.-H. Yang. Pypianoroll: Open source python package for handling multitrack pianoroll. *ISMIR Late-breaking demos*, 2018.

[31] A. Fan, M. Lewis, and Y. Dauphin. Hierarchical neural story generation. In *ACL*, 2018.

[32] C. Gan, D. Huang, P. Chen, J. B. Tenenbaum, and A. Torralba. Foley music: Learning to generate music from videos. In *ECCV*, 2020.

[33] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.

[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *NeurIPS*, 2014.

[35] C. Hawthorne, I. Simon, A. Roberts, N. Zeghidour, J. Gardner, E. Manilow, and J. Engel. Multi-instrument music synthesis with spectrogram diffusion. In *ISMIR*, 2022.

[36] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[37] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[38] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[39] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NeuIPS*, 2017.

[40] L. A. Hiller and L. M. Isaacson. *Experimental Music: Composition with an electronic computer.* McGraw-Hill Book Co, 1959.

[41] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[42] J. Ho and T. Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

[43] S. R. Holtzman. Using generative grammars for music composition. *Computer Music Journal*, 1981.

[44] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *AAAI*, 2021.

[45] C.-Z. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinculescu, and C. J. Cai. AI song contest: Human-AI co-creation in songwriting. In *ISMIR*, 2020.

[46] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck. Music transformer: Generating music with long-term structure. In *ICLR*, 2019.

[47] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. Ellis. MuLan: A joint embedding of music audio and natural language. In *ISMIR*, 2022.

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

[49] Y.-S. Huang and Y.-H. Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *ACM MM Conference*, 2020.

[50] S. Ji, J. Luo, and X. Yang. A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. *arXiv preprint arXiv:2011.06801*, 2020.

[51] S. Ji, X. Yang, and J. Luo. A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges. *ACM Computing Surveys*, 2023.

[52] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *ICML*, 2020.

[53] E. Kharitonov, A. Lee, A. Polyak, Y. Adi, J. Copet, K. Lakhotia, T.-A. Nguyen, M. Riviere, A. Mohamed, E. Dupoux, et al. Text-free prosody-aware generative spoken language modeling. In *ACL*, 2022.

[54] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi. Fr\'echet audio distance: A metric for evaluating music enhancement algorithms. *arXiv:1812.08466*, 2018.

[55] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2013.

[56] J. Kong, J. Kim, and J. Bae. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 2020.

[57] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *ICLR*, 2021.

[58] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[59] C. L. Krumhansl and M. A. Schmuckler. The petroushka chord: A perceptual investigation. *Music Perception*, 1986.

[60] K. Kumar, R. Kumar, T. De Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. De Brebisson, Y. Bengio, and A. C. Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In *NeurIPS*, 2019.

[61] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

[62] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley. AudioLDM: Text-to-audio generation with latent diffusion models. In *ICML*, 2023.

[63] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. on Information Theory*, 1982.

[64] A. A. Markov. Extension of the law of large numbers to dependent quantities. *Izv. Fiz.-Matem. Obsch. Kazan University*, 1906.

[65] J. McCormack et al. Grammar based music composition. *Complex systems*, 1996.

[66] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto. Librosa: Audio and music signal analysis in python. In *Python in Science Conference*, 2015.

[67] G. Mittal, J. Engel, C. Hawthorne, and I. Simon. Symbolic music generation with diffusion models. *arXiv preprint arXiv:2103.16091*, 2021.

[68] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan. This time with feeling: Learning expressive musical performance. *arXiv preprint arXiv:1808.03715*, 2018.

[69] 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 *ICML*, 2021.

[70] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *Open AI Blog*, 2018.

[71] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 2020.

[72] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv:2204.06125*, 2022.

[73] A. Razavi, A. Van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *NeurIPS*, 2019.

[74] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck. A hierarchical latent vector model for learning long-term structure in music. In *ICML*, 2018.

[75] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[76] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, 2015.

[77] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[78] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

[79] W. Schulze. *A formal language theory approach to music generation.* MS Thesis, University of Stellenbosch, 2009.

[80] P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. In *NAACL*, 2018.

[81] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *ICLR*, 2020.

[82] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NeurIPS*, 2014.

[83] F. Takuya. Realtime chord recognition of musical sound: Asystem using common lisp music. In *International Computer Music Conference*, 1999.

[84] J. Thickstun, D. Hall, C. Donahue, and P. Liang. Anticipatory music transformer. *arXiv preprint arXiv:2306.08620*, 2023.

[85] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, et al. LaMDA: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[86] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A generative model for raw audio. *arXiv:1609.03499*, 2016.

[87] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[88] A. van den Oord, O. Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017.

[89] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[90] E. Waite. Generating long-term structure in songs and stories. *Google Blog*, 2016.

[91] B. Wang and Y.-H. Yang. PerformanceNet: Score-to-audio music generation with multi-band convolutional residual network. In *AAAI*, 2019.

[92] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan. Music ControlNet: Multiple time-varying controls for music generation. *arXiv preprint arXiv:2311.07069*, 2023.

[93] S.-L. Wu and Y.-H. Yang. The Jazz Transformer on the front line: Exploring the shortcomings of ai-composed music through quantitative measures. In *ISMIR*, 2020.

[94] S.-L. Wu and Y.-H. Yang. MuseMorphose: Full-song and fine-grained piano music style transfer with one Transformer VAE. *IEEE/ACM Trans. on Audio, Speech, and Language Processing (T-ASLP)*, 2023.

[95] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP*, 2023.

[96] Y. Wu, E. Manilow, Y. Deng, R. Swavely, K. Kastner, T. Cooijmans, A. Courville, C.-Z. A. Huang, and J. Engel. MIDI-DDSP: Detailed control of musical performance via hierarchical modeling. In *ICLR*, 2022.

[97] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang. MidiNet: A convolutional generative adversarial network for symbolic-domain music generation. In *ISMIR*, 2017.

[98] L.-C. Yang and A. Lerch. On the evaluation of generative models in music. *Neural Computing and Applications*, 2020.

[99] Y. Yu, F. Harscoët, S. Canales, G. Reddy M, S. Tang, and J. Jiang. Lyrics-conditioned neural melody generation. In *Int. Conf. on MultiMedia Modeling (MMM)*, 2020.

[100] J. R. Zapata, M. E. P. Davies, and E. Gómez. Multi-feature beat tracking. *IEEE/ACM Trans. on Audio, Speech, and Language Processing (T-ASLP)*, 2014.

[101] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi. SoundStream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (T-ASLP)*, 2021.

[102] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[103] S. Zhao, D. Chen, Y.-C. Chen, J. Bao, S. Hao, L. Yuan, and K.-Y. K. Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *arXiv:2305.16322*, 2023.

Carnegie Mellon University. 5000 Forbes Avenue Pittsburgh, PA 15213
Email address: {anz37, qixiaoz, rikkih, shihlunw}@andrew.cmu.edu