Mustango: Toward Controllable Text-to-Music Generation

Jan Melechovsky[1*], **Zixun Guo**[2*], **Deepanway Ghosal**[1],
Navonil Majumder[1], **Dorien Herremans**[1†], **Soujanya Poria**[1†]
[1] Singapore University of Technology and Design, Singapore
[2] Queen Mary University of London, UK

https://github.com/AMAAI-Lab/mustango
https://huggingface.co/spaces/declare-lab/mustango

Abstract

The quality of the text-to-music models has reached new heights due to recent advancements in diffusion models. The controllability of various musical aspects, however, has barely been explored. In this paper, we propose `Mustango`: a music-domain-knowledge-inspired text-to-music system based on diffusion. `Mustango` aims to control the generated music, not only with general text captions, but with more rich captions that can include specific instructions related to chords, beats, tempo, and key. At the core of `Mustango` is MuNet, a Music-Domain-Knowledge-Informed UNet guidance module that steers the generated music to include the music-specific conditions, which we predict from the text prompt, as well as the general text embedding, during the reverse diffusion process. To overcome the limited availability of open datasets of music with text captions, we propose a novel data augmentation method that includes altering the harmonic, rhythmic, and dynamic aspects of music audio and using state-of-the-art Music Information Retrieval methods to extract the music features which will then be appended to the existing descriptions in text format. We release the resulting `MusicBench` dataset which contains over 52K instances and includes music-theory-based descriptions in the caption text. Through extensive experiments, we show that the quality of the music generated by `Mustango` is state-of-the-art, and the controllability through music-specific text prompts greatly outperforms other models such as `MusicGen` and `AudioLDM2`.

1 Introduction

Recently, diffusion models (Ho et al., 2020) have shown prowess in image (OpenAI, 2023a), audio (Liu et al., 2023a,b; Ghosal et al., 2023; Borsos et al., 2023) and music (Huang et al., 2023; Schneider et al., 2023) generation tasks. Generating music directly from a diffusion model poses some unique challenges. First, music adheres to specific rules related to, for instance, tempo, key, and chord progressions. Evaluating whether or not the generated music follows these conditions remains challenging. For instance, `MusicLM` (Agostinelli et al., 2023), a text-to-music model, ensures that the generated music matches the text prompts in terms of instrumentation and music vibe. However, the musicality of the generated music (e.g., musically meaningful harmonies and steady tempo) remains only partially addressed. Secondly, the availability of paired music and textual description datasets is limited (Agostinelli et al., 2023; Huang et al., 2023). Although the textual descriptions in the existing datasets include details like instrumentation or vibe, more representational descriptions that capture the structural, melodic, and harmonic aspects of music are missing. We thus argue that including this information during generation may improve the current text-to-music models in terms of musicality and controllability (e.g., following metrical structure and chord progressions). More information on related work can be found in Appendix E. Beyond existing text-to-music systems' capability (e.g., setting correct instrumentation), our proposed `Mustango` model enables musicians, producers, and sound designers to create music clips with specific text-specified conditions like following a chord progression, setting tempo, and key selection.

In this paper, we release the `MusicBench` dataset which is derived from the `MusicCaps` (Agostinelli et al., 2023) dataset and propose `Mustango` to address these challenges. To create the `MusicBench` dataset, we use two augmentation methods: *description enrichment* and *music diversification*. The aim of *description enrichment* is to augment the existing text descriptions with beats and down-

[*] Co-first authors. Both authors contributed equally.
[†] Both authors contributed equally and led this project.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 8293–8316
June 16-21, 2024 ©2024 Association for Computational Linguistics

beats location (inferred from tempo information in the text prompt), underlying chord progression, key, and tempo as control information. During inference, these additional descriptive texts could steer the music generation towards user-specified music quality. We use state-of-the-art music information retrieval (MIR) methods (Mauch and Dixon, 2010; Heydari et al., 2021; Bogdanov et al., 2013) to extract such control information from our training data. Furthermore, to diversify the music samples in the training set, we augment this dataset with variants of the existing music, altered along three aspects—tempo, pitch, and volume—that essentially determine the rhythmic, harmonic, and interpretive aspects of music. The text descriptions are also altered accordingly. The resulting MusicBench dataset is 11 times the size of the original MusicCaps (Agostinelli et al., 2023) dataset. Our proposed controllable text-to-music model Mustango incorporates a novel MuNet (music-domain-knowledge-informed UNet) that can instill the input chords, beats, key, and tempo, along with the textual description, in the generated music during the reverse-diffusion process. The results in §4 indicate Mustango creates more musically meaningful output and shows improved controllability (e.g., changing chords) over the existing text-to-music models. Our MusicBench dataset, Mustango implementation, and comparative music samples are available through https://github.com/AMAAI-Lab/mustango.

The overall contributions of this paper are:
(i) We propose Mustango, a text-to-music diffusion model with our novel MuNet module to explicitly guide the music generation towards input tempo, key, chords, and general textual description.
(ii) We release the MusicBench dataset with ~53K pairs of music audio and description with information on musical attributes like chords, key, and beats. This is achieved by altering the music samples of MusicCaps along the harmony, tempo, and volume dimensions and enriching the captions with the aforementioned musically-relevant attributes.
(iii) We empirically verify that our Mustango model is able to generate high quality music faithful to the input text descriptions, chords, and beats.

2 Dataset Creation

In this section, the methods of music feature extraction and data augmentation are introduced. Then, the application of these methods and the details of our dataset are discussed.

2.1 Feature Extraction and Description Enrichment

We extract four common music features: beats and downbeats, chords, keys, and tempo, and use them to enhance the text prompts and guide music generation. We use BeatNet (Heydari et al., 2021) to extract the beat and downbeat features, $b \in \mathbb{R}^{L_{beats} \times 2}$, where the first dimension represents the type of beat according to the meter (e.g., 1, 2, 3) and the second represents the timing of each corresponding beat in seconds. The second feature *tempo*, measured in beats per minute (BPM), is estimated by averaging the reciprocal of the time interval between beats. Chordino (Mauch and Dixon, 2010) is used to extract the chord features, $c \in \mathbb{R}^{L_{chords} \times 3}$, where the first dimension represents the roots of the chord sequence, the second represents the chord type (e.g., major, minor, maj7, etc.), and the third represents whether the chords are inverted. Finally, Essentia's (Bogdanov et al., 2013) KeyExtractor algorithm[1] is used to extract the key. The extracted features are used to enrich the textual descriptions and guide the reverse diffusion process. We notice a similar data enrichment approach in concurrent research (Gardner et al., 2023).

These features are then expressed in text format following several text templates (e.g., 'The song is in the key of A minor. The tempo of this song is Adagio. The beat counts to 4. The chord progression is Am, Cmaj7, G.'). We refer to these as control sentences and they will be appended to the original text prompt to form the enhanced prompts. A full list of the different control sentence templates can be found in Appendix I (Table 5).

2.2 Augmentation and Music Diversification

Our dataset augmentation for both music audio and text prompts increases the total amount of training data 11-fold to improve both audio quality and controllability of our model. Standard text-to-audio augmentations may not suit the nature of music audio. For example, the augmentation method used for Tango (Ghosal et al., 2023), whereby two audio samples normalized to similar audio levels are superimposed and their prompts concatenated, would not work for music due to overlapping rhythms, dissonance in harmony, and overall musical concept mismatch.

[1] essentia.upf.edu/reference/std_KeyExtractor.html

Therefore, we alter individual music samples along one of the three dimensions—pitch, speed, and volume—which determine the melodic, rhythmic, and dynamic aspects of music. We use `PyRubberband`[2] to shift the pitch of the music audio within a range of ±3 semitones following a uniform distribution. We decided to use this range in order to keep the timbre of instruments relatively untouched, as larger pitch shifts could result in unnatural timbre. We change the speed of the music audio by ±(5 to 25)%, drawn from a uniform distribution as well. Finally, we alter the volume of the audio by introducing a gradual volume change (both crescendo and decrescendo) with the minimum volume drawn from a uniform distribution from 0.1 to 0.5 times the original track's amplitude, while the maximum volume is kept untouched.

The text descriptions are enhanced and modified in tandem with the alterations to the music audio. However, to enhance the robustness of the model, we randomly discard one to four sentences from the prompt that describe the aforementioned music features. More details are illustrated in the Appendix. Finally, we used ChatGPT (OpenAI, 2023b) to rephrase the text prompts to add variety to the text prompts.

2.3 MusicBench

In this study, we make use of the MusicCaps (Agostinelli et al., 2023) dataset, which comprises a collection of 5,521 audio clips featuring music. Each clip is 10 seconds long and is sourced from the train and evaluation splits of the AudioSet (Gemmeke et al., 2017) dataset. These audio clips are accompanied by on average four-sentence-long English caption that describe the music. However, due to the inaccessibility of some audio files, our dataset comes from 5,479 samples.

We split our dataset as shown in Fig. 1. First, we split the data into TrainA and TestA sets. Subsequently, four control sentences corresponding to the music features are spliced with the original prompts to obtain the TrainB and TestB sets from TrainA and TestA, respectively. Then, by instructing ChatGPT to rephrase the TrainB text prompts, we get the final TrainC set.

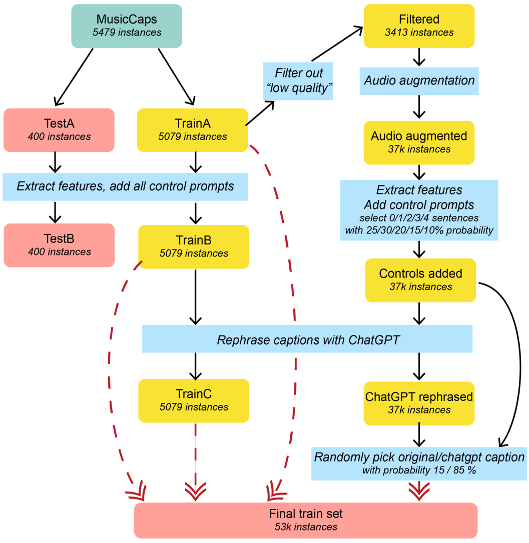


Figure 1: Composition of `MusicBench` dataset.

In addition, before performing audio augmentation, we filter out 'low quality' samples by removing samples that mention the terms 'quality' (as it is typically related to poor quality) or 'low fidelity' in the captions of TrainA set, to get 3,413 instances. The higher quality samples are altered (see §2.2) to form a set of 37k augmentation samples, comprising 6 pitch-shifted, 4 tempo-altered and 1 volume-altered sample per original sample. In the case of pitch-shifted samples, instead of randomly sampling from a uniform distribution, we used all 6 unique semitone shifts (from -3 to +3, excluding 0). Thereafter we randomly select control prompts to concatenate with the original captions. We pick $0/1/2/3/4$ prompts with a probability of $25/30/20/15/10\%$, respectively. We do this to increase the robustness of the model, as the model should be able to take inputs both with and without control sentences specifying the four music features. Then, to further increase text input robustness, we rephrase all of the captions using ChatGPT (see Appendix H). We find this step a necessary addition in our augmentation pipeline as the audio augmentation produces 11 similar samples that share a big portion of their caption with the original MusicCaps caption. By paraphrasing, we create more unique instances. In our final training dataset, we use both of the rephrased and non-rephrased prompts with a probability of $85/15\%$, respectively. Finally, we take this augmented set and concatenate it with sets TrainA, TrainB, and TrainC to get our final training set consisting of 52,768 samples, hereafter referred to as `MusicBench`. We note that TestA and TestB sets consist of 200 'low quality' (as explained above), and 200 'high quality' samples. This means that the test set distribution is slightly different from that of train set. Our intention was to create a difficult evaluation set to test

[2] `github.com/bmcfee/pyrubberband`

the controllability of Mustango in tougher conditions.

3 Mustango

Mustango consists of two components: 1) Latent Diffusion Model; 2) MuNet.

3.1 Latent Diffusion Model (LDM)

Inspired by Tango (Ghosal et al., 2023) and AudioLDM (Liu et al., 2023a), we leverage the latent diffusion model (LDM) to reduce computational complexity meanwhile maintaining the expressiveness of the diffusion model. More specifically, we aim to construct the latent audio prior z_0 extracted using an extra variational autoencoder (VAE) with condition $\mathcal{C}$, which in our case refers to a joint music and text condition. Similar to Tango, we leverage the pre-trained VAE from AudioLDM to obtain the latent code of the audio.

Through the forward-diffusion process (Markovian Hierarchical VAE), the latent audio prior z_0 turns into a standard gaussian noise $z_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, as shown in Eq. (1) where a pre-scheduled gaussian noise ($0 < \beta_1 < \beta_2 < \cdots < \beta_N < 1$) is gradually added at each forward step:

$$q(z_n|z_{n-1}) = \mathcal{N}(\sqrt{1-\beta_n} z_{n-1}, \beta_n \mathbf{I}). \tag{1}$$

For the reverse process, which reconstructs z_0 from Gaussian noise $z_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we propose MuNet (see §3.2), which is able to steer the generated music towards the given condition $\mathcal{C}$. Intuitively, backward diffusion aims to iteratively reconstruct the latent audio prior z_{n-1} from the previous step z_n until z_0, using a denoiser $\hat{\epsilon}_\theta^{(n)}(z_n, \mathcal{C})$. This denoiser is driven by classifier-free guidance, similar to Tango. The reverse diffusion process is outlined in Appendix A.

This reconstruction is trained using a noise-estimation loss where $\hat{\epsilon}_\theta^{(n)}$ is the estimated noise and γ_n is the weight of reverse step n:

$$\mathcal{L}_{DM} = \sum_{n=1}^{N} \gamma_n \mathbb{E}_{\epsilon_n \sim \mathcal{N}(\mathbf{0},\mathbf{I}), z_0} ||\epsilon_n - \hat{\epsilon}_\theta^{(n)}(z_n, \mathcal{C})||_2^2.$$

3.2 MuNet

The reverse-diffusion process, briefly described in §3.1, is conditioned on both musical attributes (beat b and chord c) and text τ ($\mathcal{C} := \{\tau, b, c\}$). This is realized through the Music-Domain-Knowledge-Informed UNet (MuNet) denoiser as follows:

$$\begin{aligned}
&U^{(1)} = z_n \\
&A_\tau = \mathrm{MHA}_{\theta_\tau}(Q = U^{(l)}, K/V = \text{FLAN-T5}(\tau)) \\
&A_b = \mathrm{MHA}_{\theta_b}(Q = A_\tau, K/V = \boldsymbol{Enc^b(b)}) \\
&A_c = \mathrm{MHA}_{\theta_c}(Q = A_b, K/V = \boldsymbol{Enc^c(c)}) \\
&U^{(l+1)} = \mathrm{UNet}_\theta^{(l)}(A_c) \\
&\epsilon_\theta^{(n)}(z_n, \mathcal{C}) := U^{(L+1)}
\end{aligned} \tag{2}$$

where, MHA is the multi-headed attention block (Vaswani et al., 2017) for the cross attentions, where Q, K, and V are query, key, and value, respectively, and FLAN-T5 is the text encoder model (Chung et al., 2022), adopted from Tango. We prioritize applying cross-attention to the beat first, as we consider a consistent rhythm to be the fundamental basis for the generated music. Subsequently, we can focus on conditioning based on chords.

MuNet consists of a UNet (Ronneberger et al., 2015)—consisting of in total L downsampling, middle, and upsampling blocks—and multiple conditioning cross-attention blocks. We use two encoders, $\boldsymbol{Enc^b}$ and $\boldsymbol{Enc^c}$, to encode the beat and chord features which leverage both the state-of-the-art Fundamental Music Embedding (FME) as well as an onset-and-beat-based positional encoding (Guo et al., 2023) which we name Music Positional Encoding (MPE). These ensure the musical features are properly captured and several fundamental music properties (e.g., intervals between pitches are translational invariant) are preserved.

We introduce the two encoders $\boldsymbol{Enc^b}$ and $\boldsymbol{Enc^c}$ that extract the beat and chord embeddings from the raw input. The beat encoder $\boldsymbol{Enc^b}$, defined in Eq. (3), encodes the beat types $b[:, 0]$ (§2.1) using One-Hot Encoding ($\boldsymbol{OH_b}$) and the beat timings $b[:, 1]$ with Music Positional Embedding. By concatenating these beat types and timing encodings and passing them through a trainable linear layer ($\boldsymbol{W_b}$), we obtain the final beat features:

$$\boldsymbol{Enc^b}(b) := \boldsymbol{W_b}(OH_b(b[:, 0]) \oplus MPE(b[:, 1])) \tag{3}$$

$$\begin{aligned}
\boldsymbol{Enc^c}(c) := \boldsymbol{W_c}(&\mathrm{FME}(c[:, 0]) \oplus OH_t(c[:, 1]) \oplus \\
&OH_i(c[:, 2]) \oplus \mathrm{MPE}(c[:, 3]))
\end{aligned} \tag{4}$$

In the chord encoder in Eq. (4), we obtain the chord embeddings by first concatenating i) FME-embedded (Guo et al., 2023) chord roots $c[:, 0]$ (see §2.1); ii) One-Hot encoded chord type $(c[:, 1])$;

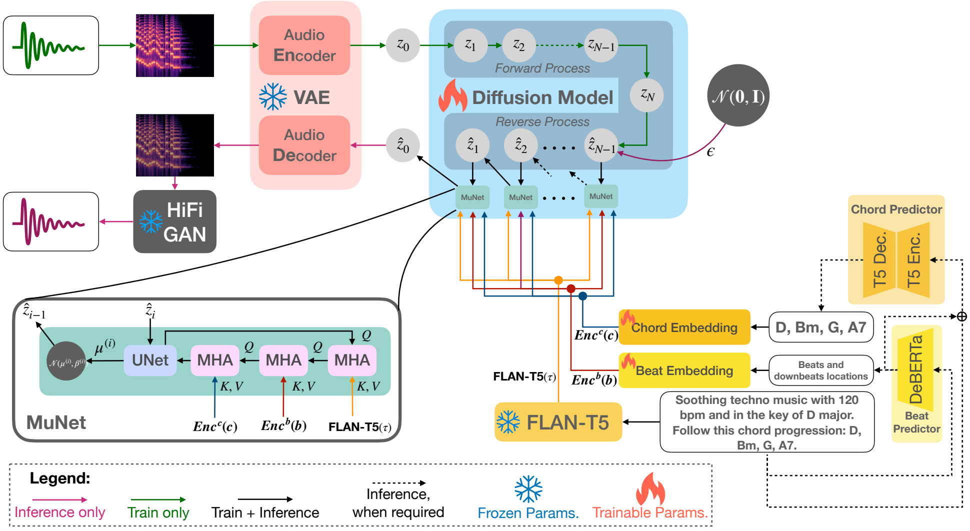


Figure 2: Depiction of our proposed Mustango model. Beats and chords are inferred from the caption when they are not provided as input.

iii) One-Hot encoded chord inversions ($c[:, 1]$); and iv) MPE-embedded (Guo et al., 2023) timing of the chords ($c[:, 3]$). Subsequently, this concatenated representation is passed through a trainable linear layer ($\boldsymbol{W_c}$). Notably, we incorporate a music-domain-knowledge informed music embedding through the use of the Fundamental Music Embedding from Guo et al. (2023), which effectively captures the translational invariant property of pitches and intervals, resulting in a more musically meaningful representation of the chord.

After obtaining the encoded beat and chord embeddings, we use two additional cross-attention layers to integrate these music conditions during the denoising process, whereas Tango used one cross-attention layer to incorporate only text conditions. This enables MuNet to leverage both music and text features during the denoising process, resulting in more controllable and meaningful music generation.

3.3 Inference

During the training phase, we use teacher forcing and hence utilize the ground truth beats and chord features to condition the music generation process. However, during inference, we employ two transformer-based text-to-music-feature generators that have been trained independently to predict the beat and chord features as follows:

Beats: We use the DeBERTa Large model (He et al., 2022) as the beats predictor. The model takes the text caption as input and predicts: i) the beat count (meter) of corresponding music, and ii) the sequence of interval duration between the beats. We predict them from the token representations of the final layer of the model. The beat count takes an integer value between 1 and 4 for the instances in our training data. Hence, we predict the beat using a four-class classification setup from the first token of the output layer. The interval durations are predicted as a float value from the second token onwards. As an example, if the beat count is predicted as 2 and the interval durations are predicted as $t_1, t_2, t_3, \ldots$, then the predicted beats are as follows: 1 at t_1, 2 at $t_1 + t_2$, 1 at $t_1 + t_2 + t_3$, etc. We keep the predicted beats time up to 10 seconds and ignore predicted timestamps beyond that.

Chords: We use the sequence to sequence FLAN-T5 Large model (Chung et al., 2022) as the chords predictor. The model takes the concatenation of the text caption and the verbalized beats as input. The verbalized beats are prepared for the example we illustrated earlier as follows: *Timestamps: t_1, $t_1 + t_2$, $t_1 + t_2 + t_3 \ldots$, Max Beat:* 2. The model is trained to generate the verbalized chords sequence with timestamps, which would look like something as follows: *Am at 1.11; E at 4.14; C#maj7 at 7.18*. We again keep the predicted chord time up to 10 seconds and ignore timestamps predicted beyond that.

4 Experiments

We conduct extensive objective and subjective evaluations to answer these research questions: i) How is the audio quality of the music generated by Mustango? ii) Does Mustango generate music with better music quality compared to other baselines? iii) Is Mustango more controllable in terms of music-specific instructions? iv) Is our data augmentation approach effective – can models trained on only this dataset compete with large-scale pre-trained models?

4.1 Baselines and Mustango Variants

We first compare Mustango with Tango since it shares a similar architecture with Mustango, except for the extra conditioning module: MuNet. To judge the efficacy of Mustango, we train the following three models from scratch: i) Tango trained on MusicCaps TrainA, ii) Tango trained on MusicBench, iii) Mustango trained on MusicBench. Additionally, we finetune Tango and Mustango from pre-trained Tango checkpoints: iv) pre-trained Tango fine-tuned on AudioCaps and MusicCaps[3], v) pre-trained Tango fine-tuned on AudioCaps[4], then finetuned on MusicBench, vi) Mustango initialized from pre-trained Tango and finetuned on MusicBench. Furthermore, we compare Mustango with state-of-the-art Text-to-Music model of MusicGen (Copet et al., 2023) and a Text-to-Audio model of AudioLDM2 (Liu et al., 2023b). For MusicGen baselines, we use the small and medium checkpoints. For AudioLDM2, we compare with their music-specific checkpoint.

4.2 Training and Additional Evaluation Set

All the models were trained at a learning rate of $4.5e-5$ using the AdamW (Loshchilov and Hutter, 2017) optimizer until convergence. Our Beat and Chord predictors are also trained on MusicBench. More details on training the classifier-free guidance, and parameters are reported in Appendix B.

Given that some of the fine-tuned models used in our experiments were exposed to the entire MusicCaps dataset in the initial Tango pre-trained checkpoint, we can only fairly evaluate those models on a different and independently created evaluation set. We thus curated 1,000 pseudo-captioned evaluation samples from the music files of Free Music Archive (FMA) (Defferrard et al., 2016), which we refer to as FMACaps. The details of creating FMACaps are reported in Appendix F.

4.3 Inference Settings and Time

In all our experiments, we use 200 diffusion steps with a classifier-free guidance scale of three for all variants of Mustango, Tango, and AudioLDM2. In MusicGen, we generate audio sequences of 10 seconds to match the outputs of the other models. We further performed a simple time measurement experiment to assess computing time of presented models. With batch size of 1, we inferred 20 samples and took the average inference time on a single Tesla V100 GPU. The obtained results are: Tango 34 sec, MusicGen-M 51 sec, Mustango 76 sec.

4.4 Objective Evaluation Methodology

Audio Quality Estimation The quality of the generated audio samples is evaluated using three objective metrics: Fréchet Distance (FD), Fréchet Audio Distance (FAD) (Kilgour et al., 2019), and Kullback-Leibler divergence (KL) as used earlier in AudioLDM and Tango.

Controllability Evaluation We evaluate each model's controllability using TestB (see §2.3) and a version of FMACaps that has all the control sentences for each sample in the prompt. We first generate music based on the text prompts and then extract the musical features mentioned in §2.1. Subsequently, we define nine metrics (all represented in percentage; in the case of binary values, 100 stands for true and 0 stands for false) to evaluate whether the music properties in the generated music match the text prompts. The metrics are:

- **Tempo Bin (TB)**: The predicted beats per minute (bpm) fall into the ground truth tempo bin.
- **Tempo Bin with Tolerance (TBT)**: The predicted bpm falls into the ground truth tempo bin or a neighboring one.
- **Correct Key (CK)**: The predicted key matches the ground truth key.
- **Correct Key with Duplicates (CKD)**: The predicted key matches the ground truth key or an equivalent key (i.e., a major key and its relative minor).
- **Perfect Chord Match (PCM)**: The predicted chord sequence perfectly matches ground truth in terms of length, order, chord root, and chord type.
- **Exact Chord Match (ECM)**: The predicted chord sequence matches the ground truth exactly in terms of order, chord root, and chord type, with tolerance for missing and excess chord instances.
- **Chord Match in any Order (CMO)**: The portion of predicted chord sequence matching the ground truth chord root and type, in any order.
- **Chord Match in any Order major/minor Type (CMOT)**: The portion of predicted chord sequence matching the ground truth in terms of chord root

[3] hf.co/declare-lab/tango-full-ft-audio-music-caps

[4] hf.co/declare-lab/tango-full-ft-audiocaps

Model	Datasets	Pre-trained	#Params	TestA			TestB			FMACaps		
				FD ↓	FAD ↓	KL ↓	FD ↓	FAD ↓	KL ↓	FD ↓	FAD ↓	KL ↓
MusicGen-S	–	✗	300M	35.40	6.82	1.81	36.40	7.54	1.75	23.21	5.13	1.31
MusicGen-M	–	✗	1.5B	36.49	6.98	1.71	35.54	6.99	1.71	22.61	5.01	1.33
AudioLDM2	–	✗	346M	32.76	5.29	1.68	33.66	5.42	1.75	**19.99**	3.01	1.33
Tango	MusicCaps	✗	866M	30.80	2.84	1.34	30.39	2.92	1.33	28.32	3.75	1.22
Tango	MusicCaps	✓	866M	34.87	4.05	1.25	37.85	4.52	1.32	28.81	2.92	1.21
Tango	MusicBench	✗	866M	28.50	2.29	1.33	28.27	2.17	1.32	26.31	**2.31**	1.16
Tango	MusicBench	✓	866M	**25.38**	1.91	**1.19**	**24.60**	1.77	1.13	24.48	2.96	**1.15**
Mustango	MusicBench	✗	1.4B	26.58	2.09	1.21	25.24	**1.57**	1.18	24.24	2.94	1.16
Mustango	MusicBench	✓	1.4B	26.35	**1.46**	1.21	25.97	1.67	**1.12**	25.18	2.34	1.16

Table 1: Objective evaluation results of the models on TestA, TestB, and FMACaps datasets.

and binary major/minor chord type, in any order (e.g., D, D6, D7, Dmaj7 are all considered major).

- **Beat Match (BM)**: The percentage of predicted beat counts that match the ground truth.

4.5 Objective Evaluation Results

Audio Quality: The results for TestA, TestB and FMACaps are presented in Table 1. Both Tango variants trained on MusicCaps are inferior to the other four models, which depicts the efficacy of our augmentation strategy. Pre-trained Tango fine-tuned on MusicBench and Mustango pre-trained seem to perform very similarly in FD and KL, but Mustango pre-trained shows a big improvement in FAD, which suggests better-perceived quality and musicality as FAD is a human perception-inspired metric. Lastly, the performance of Mustango trained from scratch is comparable in FD and KL to both pre-trained versions of Mustango and Tango trained on MusicBench, which shows that training with our augmented dataset can be an alternative to large-scale audio pre-training for music generation. Mustango also outperforms MusicGen and AudioLDM2 in FAD and KL across all three sets.

We note that the results for MusicGen and AudioLDM2 differ from what was reported in their original papers in evaluation on MusicCaps. This is due to MusicBench representing a different, more challenging, split of data than the MusicCaps evaluation set, as described in §2.3. Additionally, we note that the results of both MusicGen and AudioLDM2 show more improvement than Mustango when evaluated on FMACaps as compared to TestA and TestB. This is due to the fact that Mustango was trained on MusicBench, thus TestA and TestB represent similar distributions to the training set, while FMACaps is of a slightly different distribution. The MusicGen and AudioLDM2 models on the other hand, were trained with various large-scale data, hence they perform well on an unseen set.

Controllability: The evaluation results on controllability are shown in Table 2. On TestB, in terms of Tempo metrics, all the models perform comparably, except for MusicGen, which performs better. In Beat metrics, the models perform similarly to each other. Mustango placed second closely behind MusicGen. The similarity in performance among the models could be caused by the MusicCaps dataset already containing enough information about tempo, with words such as "slow", "fast", "moderate", etc. This information being passed through the text encoding might be a sufficient control command. Furthermore, the inaccuracy of the beat extractor combined with the fact that not all music pieces in MusicCaps have clearly audible beats might further contribute to the Beat and Tempo metrics results. Thus, having more open-sourced high-quality music data would greatly benefit development of even more controllable systems.

In Key metrics, we can observe that models trained on MusicBench perform significantly better than the ones trained on MusicCaps. Additionally, Mustango outperforms all the other models on TestB and placed second on FMACaps. Finally, in Chord controllability, Mustango outperforms all the other models by a big margin. On FMACaps, we further see that the Chord metrics are even better for Mustango with CMOT reaching 75.83. Overall, the results gathered from both TestB and modified FMACaps correlate in most aspects. Overall, Mustango performs fine in Beat and Tempo metrics, and it excels in Key and Chord controllability.

4.6 Subjective Evaluation Methodology

We conducted two rounds of subjective evaluation, each consisting of a general and an expert listening test that focuses on controllability. The first round is aimed at comparing Mustango vari-

Model	Datasets	Pre-trained	TestB									FMACaps								
			Tempo		Key		Chord				Beat	Tempo		Key		Chord				Beat
			TB	TBT	CK	CKD	PCM	ECM	CMO	CMOT	BM	TB	TBT	CK	CKD	PCM	ECM	CMO	CMOT	BM
MusicGen-S	–	✗	39.50	56.00	17.5	19.00	3.17	6.03	12.56	21.74	36.75	**45.0**	61.9	19.9	21.1	3.62	6.49	10.99	22.30	42.4
MusicGen-M	–	✗	**41.00**	**60.25**	25.5	26.25	3.97	8.21	14.42	26.76	**45.00**	42.7	**63.5**	23.5	24.3	6.38	10.60	16.24	31.51	**42.9**
AudioLDM2	–	✗	21.25	47.75	6.50	10.25	0.79	2.67	4.87	10.55	39.75	24.2	48.7	5.9	9.9	1.06	1.96	3.27	8.84	38.1
Tango	MusicCaps	✗	26.00	55.25	4.00	7.00	0.53	2.09	4.30	11.13	41.00	22.5	49.6	3.6	8.6	0.64	1.43	4.03	10.82	41.1
Tango	MusicCaps	✓	27.50	52.00	7.75	11.25	1.06	3.07	6.72	13.99	36.75	24.2	48.6	5.9	8.6	1.17	2.74	5.17	12.69	35.4
Tango	MusicBench	✗	24.75	50.75	34.25	34.50	5.56	12.03	21.54	32.21	34.25	25.5	51.0	**38.1**	**38.4**	6.60	13.45	21.18	41.49	36.4
Tango	MusicBench	✓	26.00	48.75	30.25	31.00	6.61	13.33	22.53	39.31	38.50	22.8	45.6	30.6	31.7	7.55	14.72	22.35	44.46	36.0
Mustango	MusicBench	✗	25.50	52.00	**41.75**	**42.50**	**17.99**	**32.61**	**48.74**	**68.46**	42.00	24.1	50.9	36.8	37.3	**23.94**	**35.43**	**49.59**	**75.83**	42.6
Mustango	MusicBench	✓	21.25	48.25	34.50	35.50	11.64	20.82	32.93	50.56	34.75	26.2	52.2	33.9	34.7	15.21	25.48	37.50	61.55	39.1

Table 2: Controllability evaluation results of the models on TestB and full-control variant of FMACaps. Higher numbers indicate better controllability.

ants with Tango and in the second run we compare Mustango with the state-of-the-art models: MusicGen and AudioLDM2.

In the first round of the general listening test, subjects listened to ten generated music samples for each of the four models (pre-trained Mustango, Mustango, Tango trained with MusicCaps and MusicBench) and were provided with the input text caption. The ten text prompts were custom-made by music experts in the style of MusicCaps, and are shown in Table 7 in the Appendix. The participants were asked to rate the: i) audio rendering quality (AQ), ii) relevance of the audio with the input text prompt (REL), iii) overall musical quality (OMQ), iv) rhythm consistency (RC), and v) harmony and consonance of music (HC). For the expert listening test, we added two additional music control-specific aspects to rate the degree to which the chords and tempo from the generated music match the text prompt. We denote them as MCM and MTM (musical chord/tempo match). All the aspects were rated on a 7-point Likert scale using the PsyToolkit interface (Stoet, 2010). The full questions and interface used are shown in Appendix G.

For the expert listening test in the first round, we found experts with at least five years of formal musical training who can identify music attributes from music audio. They were presented with 80 samples generated using 20 custom text prompts for each of the four models as shown in Appendix J. Samples consisted of ten *contrasting* pairs (e.g., same prompts with different chord changes) that aimed to target musical controllability.

In the second round of the general listening test, we used the same 10 captions as in the first run with five additional captions taken from FMACaps. In the expert test, we kept the same 10 contrasting pairs as in the first round. For both of these tests, we downsampled the MusicGen samples to 16 kHz to eliminate audio quality bias in listeners' responses, and we excluded the AQ metric from these tests.

4.7 Subjective Evaluation Results

A total of 48 participants participated in the first round of the general listening test, of which 26 had more than five years of formal musical training. The results in Table 3 show the average ratings for each of the metrics defined above. We can clearly see that the Tango baseline model is outperformed in all metrics by the models trained on MusicBench. Interestingly, Mustango trained from scratch performs the best in terms of audio quality, rhythm presence, and harmony. The differences in ratings are minimal between the three top models, clearly confirming that our augmentation method is effective in furthering the output quality and that Mustango is able to reach state-of-the-art quality.

A total of four experts participated in the controllability listening study. The results of the expert listening study in Table 3 further confirm that both Mustango models outperform the Tango baselines in all metrics, especially in terms of the chords of the generated music matching with the input text caption (Chord Match or MCM). This further supports the controllability results presented in Table 2 and shows that our proposed Mustango model can indeed understand music-specific text prompts.

In the second run, a total of 17 general audience listeners and 4 experts participated. The results are depicted in the lower part of Table 3. We performed a series of paired t-tests on the obtained results and conclude that Mustango outperforms MusicGen and AudioLDM2 in terms of REL, with a statistically significant difference; and performs similarly in OMQ, HC, and MTM to MusicGen, where the t-tests showed no stastically significant differences (both in general audience and expert test). Moreover, Mustango dominates in MCM. In RC, MusicGen outperformed both AudioLDM2 and Mustango.

Model	Datasets	Pre-trained	General audience					Music experts						
			REL	AQ	OMQ	RC	HC	REL	MCM	MTM	AQ	OMQ	RC	HC
Tango	MusicCaps	✓	4.09	3.68	3.55	3.91	3.80	4.35	2.75	3.88	3.35	2.83	3.95	3.84
Tango	MusicBench	✓	**4.96**	**4.26**	**4.40**	4.49	4.61	4.91	3.61	3.86	3.88	3.54	4.01	4.34
Mustango	MusicBench	✓	4.85	4.10	4.02	4.24	4.43	5.49	5.76	4.98	4.30	4.28	4.65	5.18
Mustango	MusicBench	✗	4.79	4.20	4.23	**4.51**	**4.63**	**5.75**	**6.06**	**5.11**	**4.80**	**4.80**	**4.75**	**5.59**
MusicGen-M	-	-	4.55	-	**4.40**	**5.11**	**4.63**	4.41	2.99	4.83	-	**5.01**	**5.61**	**5.31**
AudioLDM2	-	-	3.99	-	3.89	4.38	4.11	3.71	2.48	3.53	-	3.29	3.84	3.40
Mustango	MusicBench	✗	**5.18**	-	4.15	4.31	4.47	**5.79**	**6.10**	**4.84**	-	4.53	4.14	5.11

Table 3: Average ratings for each metric in the general and expert listening study. Top part of the table shows the first run of listening tests, the bottom part represents the second round of comparison with MusicGen and AudioLDM2.

4.8 Ablation Study

Although without explicitly ablating one module at a time due to resource constraints, we are able to answer the following research questions:

Is Pre-training Mustango Necessary? In one of the experiment settings in §4.1, we initialized Mustango with a pre-trained Tango checkpoint and subsequently fine-tune it using the AudioCaps dataset. This Tango model was pre-trained using 1.2 million text-audio paired samples and it encapsulates a broad understanding of general audio and text. However, we observed this did not prove beneficial for music generation (see Tables 1 to 3). Nevertheless, these checkpoints may find utility in composing music with soundscapes, such as "Hip-hop music with a lion's roar in the background."

Is MuNet Helpful? We prove the effectiveness of MuNet in §4.5 and §4.7, we show that the use of MuNet significantly enhances the performance of Mustango in terms of controllability under both objective and subjective evaluations. Moreover, several objective metrics which are not explicitly targeted at controllability (i.e., FD, FAD, and KL-divergence), consistently show superior performance when MuNet is incorporated. With classier-free guidance, MuNet does not compromise the overall quality of the generated music when the control sentences in the prompts are absent.

4.9 Discussions

As both objective and subjective evaluation results show, Mustango gives state-of-the-art performance in music quality and drastic improvement in music controllability, despite being trained on a publicly available dataset of relatively small size as compared to other available text-to-music systems such as MusicGen, which are usually trained on private large-scale licensed dataset. Although these text-to-music systems usually generate music with better audio quality or longer-term structure, which sheds light on further improvement direction of Mustango.

5 Conclusion

In conclusion, Mustango presents a significant advancement in the field of controllable text-to-music generation. Mustango is a controllable diffusion-based text-to-music system inspired by music-domain knowledge which is able to generate music that follows certain music properties embedded within user-specified text prompts. The integration of the MuNet module within Mustango enables greater music controllability over state-of-the-art text-to-music systems such as Tango, AudioLDM2 and MusicGen. We also made our dataset MusicBench and model publicly available. MusicBench contains 11 times more data than the original MusicCaps dataset and includes text prompts that contain music-theory-based description and augmented music audio.

6 Limitations

Our music generation method is limited to Western music in terms of controllability since the control information mentioned in the paper (e.g., chord, key) might be missing or appear in a different form in other non-Western music (e.g., Indian or Chinese classical music). We also assumed the availability of paired text captions of music, which was used to train our model. Mustango is also currently limited to generating music of up to ten seconds due to computational constraints. Adapting Mustango for generating long-form music is left for future work.

7 Ethical Considerations

Our training data is based on the MusicCaps dataset (Agostinelli et al., 2023). The 5.5k music samples in Music-Caps are sourced from Youtube under Creative Commons license. We perform our custom data augmentation strategies solely on this dataset. We did not use any other privately-licensed dataset.

Our listening tests involved human annotators for which the data collection protocol was approved by an independent ethics review board. More details can be found in the Appendix G.

Acknowledgements

This project has received funding from SUTD MOE grant SKI 2021_04_06 and AcRF MoE Tier-2 grant (Project no. T2MOE2008, and Grantor reference no. MOE-T2EP20220-0017) titled: "CSK NLP: Leveraging Commonsense Knowledge for NLP", for the support. This work is also supported by the Microsoft Research Accelerate Foundation Models Academic Research program. We also thank Huggingface for sponsoring the live demo on Huggingface Space. Zixun Guo is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported jointly by UK Research and Innovation [Grant number EP/S022694/1] and Queen Mary University of London.

References

Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. 2023. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*.

Dmitry Bogdanov, Nicolas Wack, Emilia Gómez Gutiérrez, Sankalp Gulati, Herrera Boyer, Oscar Mayor, Gerard Roma Trepat, Justin Salamon, José Ricardo Zapata González, Xavier Serra, et al. 2013. Essentia: An audio analysis library for music information retrieval. In *Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8.* International Society for Music Information Retrieval (ISMIR).

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. 2023. Audiolm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. w2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2021, Cartagena, Colombia, December 13-17, 2021*, pages 244–250. IEEE.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. 2023. Simple and controllable music generation. *arXiv preprint arXiv:2306.05284*.

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. 2016. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*.

Josh Gardner, Simon Durand, Daniel Stoller, and Rachel M Bittner. 2023. Llark: A multimodal foundation model for music. *arXiv preprint arXiv:2310.07160*.

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE.

Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria. 2023. Text-to-audio generation using instruction tuned llm and latent diffusion model. *arXiv preprint arXiv:2304.13731*.

Zixun Guo, J. Kang, and D. Herremans. 2023. A domain-knowledge-inspired music embedding space and a novel attention mechanism for symbolic music modeling. In *Proc. of the 37th AAAI Conference on Artificial Intelligence*, Washington DC. AAAI, AAAI.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2022. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Dorien Herremans, Ching-Hua Chuan, and Elaine Chew. 2017. A functional taxonomy of music generation systems. *ACM Computing Surveys (CSUR)*, 50(5):1–30.

Mojtaba Heydari, Frank Cwitkowitz, and Zhiyao Duan. 2021. Beatnet: Crnn and particle filtering for online joint beat downbeat and meter tracking.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.

Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis. 2022. Mulan: A joint embedding of music audio and natural language. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, pages 559–566.

Qingqing Huang, Daniel S Park, Tao Wang, Timo I Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Frank, et al. 2023. Noise2music: Text-conditioned music generation with diffusion models. *arXiv preprint arXiv:2302.03917*.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. 2019. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In *INTERSPEECH*, pages 2350–2354.

Peike Li, Boyu Chen, Yao Yao, Yikai Wang, Allen Wang, and Alex Wang. 2023. Jen-1: Text-guided universal music generation with omnidirectional diffusion models. *arXiv preprint arXiv:2308.04729*.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. 2023a. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*.

Haohe Liu, Qiao Tian, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. 2023b. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *arXiv preprint arXiv:2308.05734*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Matthias Mauch and Simon Dixon. 2010. Approximate note transcription for the improved identification of difficult chords. In *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010, Utrecht, Netherlands, August 9-13, 2010*, pages 135–140. International Society for Music Information Retrieval.

OpenAI. 2023a. DALL·E 2.

OpenAI. 2023b. Introducing ChatGPT.

Marco Pasini and Jan Schlüter. 2022. Musika! fast infinite waveform music generation. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, pages 543–550.

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 18*, pages 234–241. Springer.

Flavio Schneider, Zhijing Jin, and Bernhard Schölkopf. 2023. Mo\^ usai: Text-to-music generation with long-context latent diffusion. *arXiv preprint arXiv:2301.11757*.

Gijsbert Stoet. 2010. Psytoolkit: A software package for programming psychological experiments using linux. *Behavior research methods*, 42:1096–1104.

Kun Su, Judith Yue Li, Qingqing Huang, Dima Kuzmin, Joonseok Lee, Chris Donahue, Fei Sha, Aren Jansen, Yu Wang, Mauro Verzetti, et al. 2023. V2meow: Meowing to the visual beat via music generation. *arXiv preprint arXiv:2305.06594*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2022. Soundstream: An end-to-end neural audio codec. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30:495–507.

Pengfei Zhu, Chao Pang, Shuohuan Wang, Yekun Chai, Yu Sun, Hao Tian, and Hua Wu. 2023. Ernie-music: Text-to-waveform music generation with diffusion models. *ArXiv*, abs/2302.04456.

A Reverse Diffusion Process

The reverse process to iteratively reconstruct z_0 is as following:

$$p_\theta^{mus}(z_{n-1}|z_n, \mathcal{C}) = \mathcal{N}(\mu_\theta^{(n)}(z_n, \mathcal{C}), \tilde{\beta}^{(n)}), \tag{5}$$

$$\mu_\theta^{(n)}(z_n, \mathcal{C}) = \frac{1}{\sqrt{\alpha_n}}[z_n - \frac{1 - \alpha_n}{\sqrt{1 - \bar{\alpha}_n}}\hat{\epsilon}_\theta^{(n)}(z_n, \mathcal{C})], \tag{6}$$

$$\tilde{\beta}^{(n)} = \frac{1 - \bar{\alpha}_{n-1}}{1 - \bar{\alpha}_n}\beta_n, \tag{7}$$

$$\alpha_n = 1 - \beta_n, \tag{8}$$

$$\overline{\alpha}_n = \prod_{i=1}^{n} \alpha_n, \tag{9}$$

$$\hat{\epsilon}_\theta^{(n)}(z_n, \mathcal{C}) = w\, \epsilon_\theta^{(n)}(z_n, \mathcal{C}) + (1 - w)\epsilon_\theta^{(n)}(z_n), \tag{10}$$

where w is the guidance scale in Eq. (10) used during inference. During training however, $\epsilon_\theta^{(n)}(z_n, \mathcal{C})$ is directly used for noise estimation where the conditions $\mathcal{C}$ are randomly dropped as specified in §4.2.

B Training Details

To further improve the robustness of the classifier-free guidance in Mustango, we use these three dropouts during training:

1. With 5% probability, drop all the inputs (text, beats, and chords);
2. With 5% probability, drop an input feature (applied to each of the inputs separately);
3. We determine the probability of masking a prompt as $\min(100, 10\frac{N}{M})\%$, where N represents the number of sentences in the current prompt, and M is the average number of sentences per prompt. Once a prompt is chosen for masking, we randomly draw an integer X from a uniform distribution in the range [20, 50] and proceed to remove $X\%$ of the input sentences in the prompt.

The idea behind the first two dropouts is to enable the model to work with incomplete, faulty, or missing input information. The third dropout is aimed at improving robustness for short text inputs. We apply these dropouts to Tango as well, with a small modification: since Tango does not use music feature inputs, we replace the first two dropouts with a single 10 % probability of dropping all text.

To train Mustango and Tango baselines, we used various GPU resources: 4 Nvidia Tesla V100 GPUs, and 8 Quadro RTX 8000 GPUs. Training time ranged from 5 to 10 days with effective batch size of 32.

C Performance of the Predictors

During the inference phase, we utilize pre-trained predictors for chord and beat predictions based on textual prompts. These predictors exhibit exceptional performance when the prompts explicitly contain chord and beat information, achieving accuracy of 94.5 % on the TestB dataset. However, our interest extends to evaluating their performance in scenarios where control sentences are absent from the prompt—essentially, do these predictors generate noisy chords and beats? The concern is that such noise might propagate from the predictors to Mustango, significantly impacting the overall quality of the generated music.

In our experiments, TestA serves as a scenario where control sentences are not included in the textual prompts. Upon comparing the performance (Table 1) of Tango and Mustango on TestA, we observe that the latter outperforms the former across most metrics. This observation indicates that the control predictors do not compromise the performance of Mustango relative to Tango. The adaptability of these predictors to specific themes or styles in the absence of control sentences remains a potential avenue for future exploration, a topic we briefly touch upon below.

First, we investigate the effect of the Chord predictor on the generated output in a little comparison experiment. We take both TestA and TestB samples synthesized by Mustango and extract features from them. Then, we evaluate the chord control metrics of PCM, ECM, CMO, and CMOT using chords predicted by chord predictor vs chords detected in the audio from feature extraction. The metrics on TestA are PCM - 16.15, ECM - 33.95, CMO - 39.81, and CMOT - 47.82. The metrics on TestB are PCM - 17.75, ECM - 32.07, CMO - 47.36, and CMOT - 66.80. These results show that Mustango tends to follow the chords predicted by the chord predictor quite often. While the results on TestA are a bit lower than on TestB, they are still higher than Tango results on TestB as shown in Table 2.

Second, we take a look at some specific examples:

Prompt: "This folk song features a female voice singing the main melody. This is accompanied by a tabla playing the percussion. A guitar strums chords. For most parts of the song, only one chord is played. At the last bar, a different chord is played. This song has minimal instruments. This song has a story-telling mood. This song can be played in a village scene in an Indian movie. *The chord sequence is Bbm, Ab. The beat is 3. The tempo of this song is Allegro. The key of this song is Bb minor.*"

Without control sentences in italics (TestA): **chords predicted**: ["G", "C", "G", "C", "G", "C"], **chords predicted time**: [0.46, 1.21, 3.25, 5.48, 7.24, 8.92]. **chords extracted from audio**: ["G6", "C", "G", "C", "G", "Cmaj7"], **chords time extracted from audio**: [0.46, 1.58, 3.07, 5.94, 7.62, 9.66]

With control sentences in italics (TestB): **chords predicted**: ["Bbm", "Ab"], **chords predicted time**: [0.46, 7.24], **chords extracted from audio**: ["F#maj7", "Ab"], **chords time extracted from audio**: [0.46, 7.43].

Prompt: "A female singer sings this bluesy melody. The song is medium tempo with minimal guitar accompaniment and no other instrumentation. The song's medium tempo is very emotional and passionate. The song is a modern pop hit but with poor audio quality. *The key of this song is G minor. The time signature is 3/4. This song goes at 168.0 beats per minute. The chord progression in this song is Am7, G7, Cm, G, A7.*"

Without control sentences in italics (TestA): **chords predicted**: ["C#m7", "C#m7", "C#m7", "C#m7", "C#m7"], **chords predicted time**: [0.46, 3.25, 6.32, 8.17, 9.29], **chords extracted from audio**: ["F#", "C#m", "F#m", "C#m7"], **chords time extracted from audio**: [0.46, 1.21, 4.55, 5.39]

With control sentences in italics (TestB): **chords predicted**: ["Am7", "G7", "Cm", "G", "A7"], **chords predicted time**: [0.46, 1.67, 3.53, 5.48, 8.92], **chords extracted from audio**: ["Am", "G", "C", "Gmaj7", "A6", "Gmaj7"], **chords time extracted from audio**: [0.46, 1.67, 3.72, 5.94, 8.73, 9.85]

The two depicted samples give us some specific insights into the predicted chords and chords detected in the generated audio. Most of the time, Mustango follows the chords provided by the chord predictor in most cases. We can observe some substitutions in the actual chords detected from the audio compared to the predicted chords, e.g., G became G6, C became Cmaj7, and C#m7 became C#m. These chord substitutions are very close musically and could even be a consequence of the feature extraction system not being 100% accurate. The substitution of Bbm for F#maj7 is more of a change at first glance, but given that 2 out of 3 notes in Bbm are also contained in the 4-note F#maj Mustango, we see this substitution as understandable too. However, we note that this substitution would not be considered a valid one in any of our proposed chord control metrics.

Last but not least, in the absence of explicit control sentences in the prompt, we observe that the chords predicted by the chord predictor usually follow specific patterns. The generated samples follow a pattern of two chords that alternate (A, B, A, B, A, B). Another type of an observed pattern is one chord repeated (A, A, A, A, A, A). A more elaborate study on the Chord predictor behavior should be a topic for future work.

D Insights from the Human Annotation

Here, we take a look at some generated examples from the expert listening test, specifically a blues sample with the following prompt: "An instrumental blues melody played by a lead guitar and a strumming acoustic guitar. The acoustic guitarist's strumming keeps the rhythm steady. The chord sequence is G7, F7, C7, G7. This song goes at 100 beats per minute."

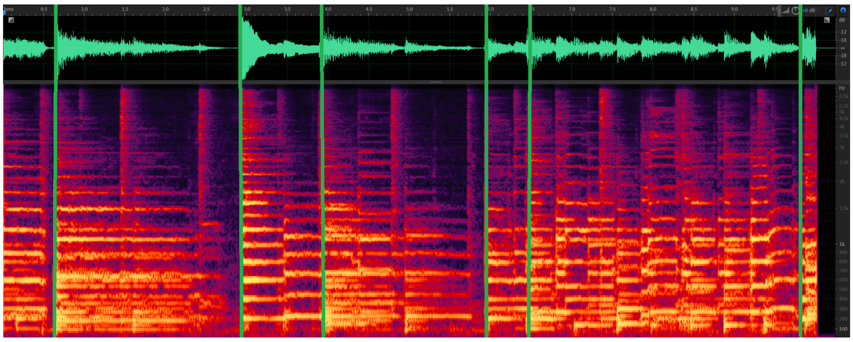


Figure 3: Mel-spectrogram of a blues sample generated by Tango trained on MusicBench.

In Figure 3 we can see the mel-spectrogram generated by pre-trained Tango finetuned on MusicBench. As is clear from the spectrogram and the waveform attached, the music appears a bit abruptly in contrast to the sample generated by Mustango depicted in Figure 4 where the rhythm is very consistent. This seems to reflect the results of our expert listening study from Table 3. The predicted beat timestamps by our Beat predictor that condition the diffusion process are as follows: **beats predicted**: [[0.26, 0.87, 1.52, 2.09, 2.76, 3.41, 4.0, 4.57, 5.1, 5.65, 6.22, 6.79, 7.36, 7.79, 8.3, 8.8, 9.3, 9.75], 3]. **These predicted beat timestamps show that there is a beat roughly every 0.6 seconds, which corresponds to 100 beats per minute tempo. This is the tempo ordered and properly predicted to condition the model.**

When it comes to chords, Tango would sometimes not follow the chords, make them sound unclear, or not give them enough time to sound through. On the other hand, Mustango seems to follow the predicted chords as well as their starting time. We take a look at the same blues example. The predicted chord condition from the Chord predictor is as follows: **chords predicted**: ["G7", "F7", "C7", "G7"], **chords predicted time**: [0.46, 2.04, 4.37, 8.17]. We can see that the chord onset time is nicely spread in time. This is also clear from listening to the sample and seeing the spectrogram with perceived chord starts in Figure 4. To confirm this, we extracted the chord features from the generated audio to compare. The chord feature extracted from the audio sample generated by Mustango is: **chords**: ["G7", "F7", "C", "G7"], **chords time**: [0.46, 1.76, 4.74, 8.45] Interestingly, the match of timing and chord sequence is very clear here. The substitution of the C7 chord for C can be a minor mistake either on the generation part or the feature extraction part. If we consider the chord metrics from the controllability evaluation in §4.4, this would yield a score of 100 for CMOT and a score of 75 for CMO and ECM. In contrast, the sample generated by pre-trained Tango finetuned on MusicBench sounds more unstable and does not give enough time to chords to sound through. The chord feature extracted from the audio sample generated by pre-trained Tango finetuned on MusicBench is: **chords**: ["Fm6", "G", "Dm", "G", "C", "Gm"], **chords time**: [0.46, 2.69, 3.53, 5.76, 6.69, 9.66]. We can see that there are 6 chords extracted from the audio sample instead of the ordered 4, and they do not match too well, as we see a minor type of F chord instead of a major; G also appears in a minor variant once; and there is an additional Dm chord too. This would yield a CMOT score of 75, but CMO and ECM scores of 0. The perceived chord starts can be seen in Figure 3.

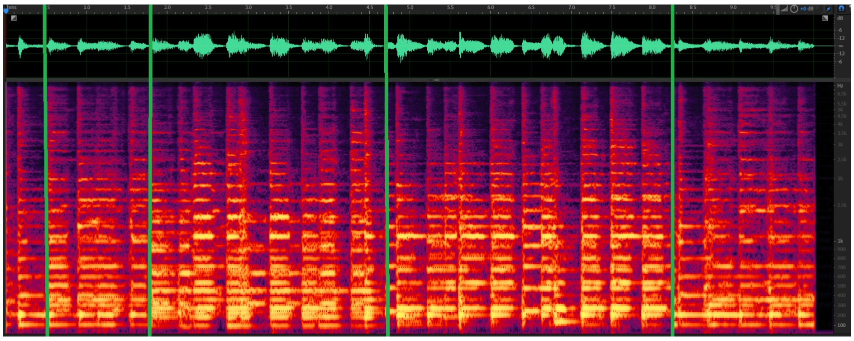


Figure 4: Mel-spectrogram of a blues sample generated by `Mustango` with vertical lines showing perceived chord starts.

Model	Dataset	Core Architecture	Area of Focus
MusicLM (Agostinelli et al., 2023)	Private Dataset including an open-sourced test set: MusiCaps	Hierarchical Seq2Seq Modeling	Audio Quality, Text-Music Relevance
Noise2Music (Huang et al., 2023)	Private Dataset obtained via pseudo labelling	2-stage Diffusion	Audio Quality, Text-Music Relevance
Ernie-Music (Zhu et al., 2023)	Private Dataset consisting of online music and corresponding comments	Diffusion (without using Audio Latent)	Audio Quality, Text-Music Relevance, Diversity
MusicGEN (Copet et al., 2023)	Private Dataset	Autoregressive Transformer	Audio Quality, Text-Music Relevance, Music Quality, Controllability (Follows given melodies)
Mousai (Schneider et al., 2023)	Private Dataset; Data Collection Pipeline partially open-sourced	2-Stage Latent Diffusion	Audio Quality, Text-Music Relevance, Music Quality, Efficiency, Long-Term Structure, Diversity
JEN1 (Li et al., 2023)	Private Dataset	Latent Diffusion, Multi-Task Learning	Audio Quality, Text-Music Relevance, Music Quality, Efficiency
`Mustango` (ours)	**Public Dataset** + Music-Domain-Knowledge-Enhanced Data Augmentation	Latent Diffusion	Audio Quality, Text-Music Relevance, Music Quality, **Music Controllability** (Follows user-specific text prompts including tempo, chord changes, etc)

Table 4: High-level comparison among various recent text-to-music models.

E Related Works

In this section, we describe existing state-of-the-art research on text-to-audio generation, followed by the more specific domain of text-to-music generation. For audio generation, AudioLM (Borsos et al., 2023) uses the state-of-the-art semantic model w2v-Bert (Chung et al., 2021) to generate the semantic tokens from audio prompt. These tokens condition the generation of acoustic tokens that are decoded using acoustic model SoundStream (Zeghidour et al., 2022) to generate audio.

AudioLDM (Liu et al., 2023a) is a text-to-audio framework that leverages CLAP (Wu et al., 2023), a joint audio-text representation model, and a latent diffusion model (LDM). Specifically, an LDM is trained to generate the latent representations of melspectrograms which are obtained using a VAE. During diffusion, the CLAP embeddings are utilized to guide the generation. `Tango` (Ghosal et al., 2023) leverages the pre-trained VAE from AudioLDM and replaces the CLAP model with an instruction fine-tuned large language model: FLAN-T5 to achieve comparable or better results while training with a much smaller dataset.

In the field of music generation, there is a long history of generated MIDI music (Herremans et al., 2017). Using MIDI may be useful for producers to work with in Digital Audio Workstations, yet it has the disadvantage that datasets are extremely limited. In recent years, the focus ofconditional music generation within the audio domain has centered around musical conditions, such as note intensity or tempo (Pasini and Schlüter, 2022). More recently, however, models that directly generate *audio* music from text captions have emerged. A summary of these papers are provided in Table 4. MusicLM (Agostinelli et al., 2023) uses two pre-trained models, MuLan (Huang et al., 2022), a joint text-music embedding model, and w2v-Bert (Chung et al., 2021), a masked language model to address the challenge of maintaining both synthesizing quality and coherence during music generation. These two pre-trained models are then utilized to condition the acoustic model SoundStream (Zeghidour et al., 2022) which in turn can generate acoustic tokens autoregressively. These acoustic tokens are then decoded by SoundStream to become the final audio output. MusicLM outperforms two existing commercially available text-to-music software: Mubert[5] and Riffusion[6] in terms of Frechet Audio Distance, Faithfulness to the text description, KL divergence, and Mulan Cycle Consistency. Since no publications are linked to these latter two systems, the model details are not available.

Another text-to-music model is Noise2Music (Huang et al., 2023). To obtain training data for the model, the authors propose a method to obtain a large amount of paired music and text data in which LaMDA-LF (Thoppilan et al., 2022), a large language model, is used to generate multiple generic candidate text descriptions. The aforementioned joint text-music embedding MuLan is then utilized to select the best candidates for existing music data. The obtained music and text pairs are then used to train a two-stage diffusion model, where the first diffusion model generates an intermediate representation and the second generates the final audio output.

Ernie-Music (Zhu et al., 2023) uses a diffusion model to generate music audio from free-form text. It is trained using a private dataset which consists of online music and the top-rated comments from the comment section. The authors recruited 10 casual music listeners to participate in a listening study. Results showed that Ernie-Music outperforms two non-diffusion-based generative systems.

In recent months, a number of text-to-music models have come out. Schneider et al. (2023) proposes a 2-stage diffusion model in which the first diffusion magnitude autoencoder (DMAE) learns a meaningful latent representation of music (64 times smaller than the input), while in the second diffusion model, text condition along with the latent acquired at the first stage is included to guide the final music generation. MusicGen (Copet et al., 2023) utilizes a single-stage transformer LM with efficient token interleaving patterns to achieve high-quality generation and better controlabillity over the output. MusicGen can be conditioned by a text prompt, or by an audio fragment in the form of a chromagram. The system was trained with a licensed dataset. The JEN-1 model (Li et al., 2023) is an omnidirectional diffusion model designed to perform various tasks such as text-guided music generation, music inpainting, and continuation. Another interesting recent model is that of Su et al. (2023), which focuses on generating music pieces to complement video, conditioned on both video and text inputs. Unlike text, video conditioning can contain a lot of temporal information, such as beats and emotions, which are important for music.

F FMACaps dataset creation

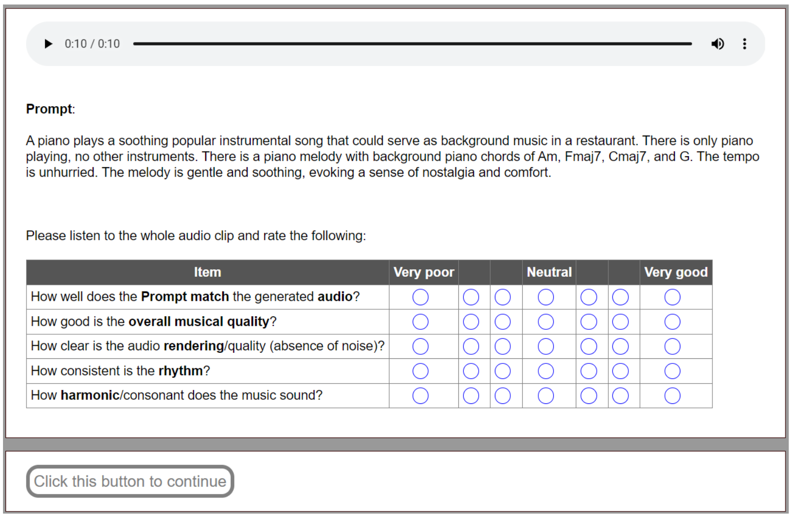
We source the new music files from the Free Music Archive (FMA) (Defferrard et al., 2016), a large dataset of popular songs. In particular, we took 1,000 random samples from FMA-large and clipped out a random 10-second fragment from each of them. Then, we used Essentia's tagging models (Bogdanov et al., 2013) to assign tags to audio. Specifically, we used the models for general auto-tagging, mood, genre, instrumentation, voice, and voice gender which provide us with a rich set of tags along with their probabilities. Then, a music expert wrote text descriptions for 25 of the samples based on the audio as well as the extracted tags. Next, we instructed ChatGPT to perform an in-context learning task to get pseudo-prompts from tags for the rest of the dataset. Finally, we added relevant control sentences to the prompts after extracting relevant music features, as described in §2.1. Similar to our training set, we added 0/1/2/3/4 control sentences with a probability of 25/30/20/15/10% respectively. We refer to this evaluation set as `FMACaps`.

G User Interface and Questions used for Listening Studies

The human evaluation participants for the listening tests were recruited through email contacts of various music research and machine learning communities. Their age ranged from 15 to 60+ years. No specific information on the country of residence was collected. Participation was free of reward and voluntary. All the participants were informed about this before their participation. The data collection protocol for the listening test was approved by an independent ethics review board.

[5] https://github.com/MubertAI/Mubert-Text-to-Music

[6] https://www.riffusion.com/



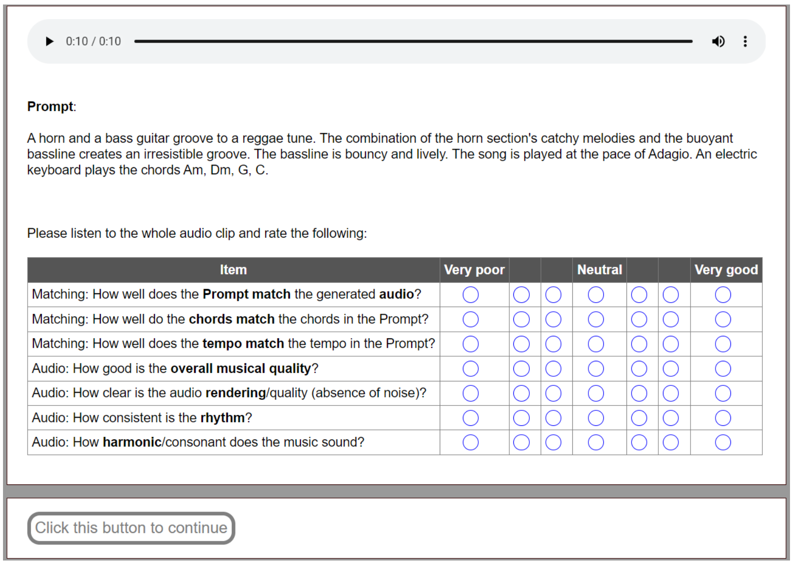
Prompt:

A piano plays a soothing popular instrumental song that could serve as background music in a restaurant. There is only piano playing, no other instruments. There is a piano melody with background piano chords of Am, Fmaj7, Cmaj7, and G. The tempo is unhurried. The melody is gentle and soothing, evoking a sense of nostalgia and comfort.

Please listen to the whole audio clip and rate the following:

Item	Very poor			Neutral			Very good
How well does the **Prompt match** the generated **audio**?	○	○	○	○	○	○	○
How good is the **overall musical quality**?	○	○	○	○	○	○	○
How clear is the audio **rendering**/quality (absence of noise)?	○	○	○	○	○	○	○
How consistent is the **rhythm**?	○	○	○	○	○	○	○
How **harmonic**/consonant does the music sound?	○	○	○	○	○	○	○

Figure 5: Question interface used for the general listening test.



Prompt:

A horn and a bass guitar groove to a reggae tune. The combination of the horn section's catchy melodies and the buoyant bassline creates an irresistible groove. The bassline is bouncy and lively. The song is played at the pace of Adagio. An electric keyboard plays the chords Am, Dm, G, C.

Please listen to the whole audio clip and rate the following:

Item	Very poor			Neutral			Very good
Matching: How well does the **Prompt match** the generated **audio**?	○	○	○	○	○	○	○
Matching: How well do the **chords match** the chords in the Prompt?	○	○	○	○	○	○	○
Matching: How well does the **tempo match** the tempo in the Prompt?	○	○	○	○	○	○	○
Audio: How good is the **overall musical quality**?	○	○	○	○	○	○	○
Audio: How clear is the audio **rendering**/quality (absence of noise)?	○	○	○	○	○	○	○
Audio: How consistent is the **rhythm**?	○	○	○	○	○	○	○
Audio: How **harmonic**/consonant does the music sound?	○	○	○	○	○	○	○

Figure 6: Question interface used for the controllability listening test with music experts.

H ChatGPT Rephrasing

In our data augmentation pipeline, we rephrase captions using ChatGPT. The instructions to the API are as follows:

"I have a song for which the caption is the following:" *caption* "I have made some changes to the audio file which are optionally described towards the end of the caption. Can you rephrase the caption more naturally in a single paragraph using all the musical terms provided above? You should generate only the caption and nothing else. Do not use the word modification in your generation. The length of the new caption should be no more than eight sentences."

Examples of rephrasing:

Original

This folk song features a male voice singing the main melody in an emotional mood. This is accompanied by an accordion playing fills in the background. A violin plays a droning melody. There is no percussion in this song. This song can be played at a Central Asian classical concert. The chord sequence is B, Dmaj7, B, Gbm. This song goes at 95.0 beats per minute. The beat counts to 2. This song is in the key of B minor.

Rephrased

A male voice sings the main melody with emotional intensity in this Central Asian folk song. The accordion fills the background while a violin provides a droning melody. There is no percussion in this piece, which can be played at a classical concert in Central Asia. With a chord sequence of B, Dmaj7, B, Gbm, the song has a beat count of 2 at a tempo of 95.0 beats per minute. The song is in the key of B minor.

Original

This folk song features a male voice singing the main melody in an emotional mood. This is accompanied by an accordion playing fills in the background. A violin plays a droning melody. There is no percussion in this song. This song can be played at a Central Asian classical concert. The key is C minor. The chord progression in this song is C, D#maj7, C, Gm.

Rephrased

This emotional folk song, perfect for a Central Asian classical concert, showcases a male voice singing the main melody accompanied by a droning violin and accordion fills in the background. With no percussion present, the key of C minor sets the tone, and the chord progression follows suit with C, D#maj7, C, Gm.

I Control-Sentence Templates to Enhance the Prompts

Feature	Input	Output sentences
Tempo	int i	• The bpm is i. • The tempo of this song is i beats per minute. • This song goes at i beats per minute.
Tempo	string $w \in$ [‘Grave’, ‘Largo’, ‘Adagio’, ‘Andante’, ‘Moderato’, ‘Allegro’, ‘Vivace’, ‘Presto’, ‘Prestissimo’]	• This song is in w. • The tempo of this song is w. • This song is played in w. • The song is played at the pace of w.
Beat count	int b	• The time signature is $\frac{b}{4}$. • The beat is b. • The beat counts to b.
Chords	text list of chords s	• The chord sequence is s. • The chord progression in this song is s.
Key	string $rootnote$ string $m \in$ [‘major’, ‘minor’]	• The key is $rootnote\ m$ • The key of this song is $rootnote\ m$. • This song is in the key of $rootnote\ m$
Volume change	float f indicating start/end time of crescendo/decrescendo, string $w \in$ [’crescendo’, ’decrescendo’], and $u \in$ [’increase’, ’decrease’]	• There is a w from start until f seconds • The song starts with a w. • u the volume progressively! • There is a w from f seconds on. • At seconds f, the song starts to gradually u in volume. • Midway through the song, a w starts.

Table 5: Rules used to create text sentences from input parameters detected from the data (key, chords, beats, tempo), and those used to augment the data (crescendo, etc.). Note that the tempo strings w were assigned based on music-theory binning in terms of bpm: Grave (0, 40], Largo (40, 60], Adagio (60, 70], Andante (70, 90], Moderato (90, 110], Allegro (110, 140], Vivace (140, 160], Presto (160, 210], Prestissimo (210, ∞).
.

J Custom Captions used for Listening Studies

1	This piece is an instrumental reggae song that is very chill and slow. There is no singer. It is relaxing to hear the groove with the bass guitar. The song includes reggae electric guitar, horn, and percussion like bongos. The keyboard provides lush chords. The time signature is 4/4. The chord progression is G, F, C.
2	This instrumental blues song goes very slow at a bpm of 50. You can hear the bass, harmonica and guitar grooving. The harmonica plays a solo over the harmonious guitar and bass.
3	This classical piece is a waltz played by a string quartet. It includes two violins, a viola, and a cello, the beat counts to 3. It sounds elegant, and has a strong first beat. It has a natural and danceable rhythm. The mood is romantic. The chord progression is Em, Am, D, G.
4	African drums are playing a complex rhythm while a male vocalist chants a ritual. The atmosphere is mesmerizing. The complex drumming pattern is a mesmerizing blend of syncopation, polyrhythms, and intricate patterns. It takes place somewhere in the wilderness, or in an indigenous village.
5	This rock piece with guitars and drums is loud but fades out later on and becomes softer. It sounds powerful yet melancholic. It is instrumental only. A bass guitar provides a steady beat, enhancing the groove and energy of the song.
6	A single bass instrument is playing a running baseline. It has a jazzy feeling to it and sounds mellow. This could be played in a jazz club. The tempo is 120 bpm.
7	This is a hip hop song. It has two rappers taking turns, one female and one male. An electronic synth melody sample in the background keeps on looping. We can hear electronic beats and sometimes record-scratching sound effects.
8	A smooth jazz song with saxophone, drums and guitar with a chord progression of Dm7, G7, Cmaj7. The song is relaxed and slow. There are no vocals, it is instrumental only. The saxophone produces a velvety tone that delivers an emotive melody.
9	A piano plays a soothing popular instrumental song that could serve as background music in a restaurant. There is only piano playing, no other instruments. There is a piano melody with background piano chords of Am, Fmaj7, Cmaj7, and G. The tempo is unhurried. The melody is gentle and soothing, evoking a sense of nostalgia and comfort.
10	Indian folk music with a sitar and female vocals. It evokes a sense of zen and elevation. A sitar player begins with a gentle and melodic introduction, plucking the strings with precision and emotion. There are rhythmic beats of traditional hand percussion instruments, such as the tabla. It could be played at a cultural festival to showcase Indian culture.
11	This is a melodic and energetic rock ballad with a male vocalist. It has a country vibe and is of alternative or popfolk genre. The electric and acoustic guitars and the bass create the background, while the drums give a regular beat. The singer's voice is complemented by a piano.
12	This is a slow classical piece with violins and pianos. It has a film score feel and is instrumental only. The orchestration is soft, with strings and flutes.
13	This fast and energetic rock song is performed by a male singer. The genre is alternative or punk rock. The background is formed by a guitar, an electric guitar, bass, and drums. There is also a synthesizer.
14	This is a slow and ambient instrumental piece with a soundscape that feels like space. The atmosphere is meditative and relaxing but with a certain darkness to it. The genre is electronic soundtrack, and the music is completely instrumental with a synthesizer, bass, and drums forming the background. This song goes at 167.0 beats per minute.
15	This is an instrumental piece with Indian and classical elements. The sitar, violin, and flute play prominent roles in creating a meditative and relaxing mood. The percussion and guitar provide a background rhythm to this world and jazz fusion.

Table 6: Custom captions used for the general listening test. Captions in the top part were used in both first and second runs, captions in the bottom part were used in the second run only.

Table 7 presents the 20 text prompts used for the expert listening studies. They consist of 10 contrasting pairs written by music experts. Care was given to make sure that they were realistic and that there were no contradicting elements in the prompts. For instance, caption 1 in Table 7 contrasts with caption 2. They share the same original caption *"An instrumental blues melody played by a lead guitar and a strumming acoustic guitar. The acoustic guitarist's strumming keeps the rhythm steady."*. However, the control sentences are different: *"The chord sequence is G7, F7, C7, G7. This song goes at 100 beats per minute."* versus *"The chord sequence is Dm, Am, Em. This song goes at 60 beats per minute."*. Both chord sequences come from blues progressions, but they belong to a different key/mode. The tempo of caption 2 is significantly slower. Such captions are ideally suited to test if the control sentences influence the generated music.

1	An instrumental blues melody played by a lead guitar and a strumming acoustic guitar. The acoustic guitarist's strumming keeps the rhythm steady. The chord sequence is G7, F7, C7, G7. This song goes at 100 beats per minute.
2	An instrumental blues melody played by a lead guitar and a strumming acoustic guitar. The acoustic guitarist's strumming keeps the rhythm steady. The chord sequence is Dm, Am, Em. This song goes at 60 beats per minute.
3	A piano plays a popular melody over the chords of Am, Fmaj7, Cmaj7, G. There is only piano playing, no other instruments or voice. The tempo is Adagio.
4	A piano plays a popular melody over the chords of Gm, Bb, Eb. There is only piano playing, no other instruments or voice. The tempo is Vivace.
5	This is an intense and loud punk song with guitars and drums. It is instrumental only. It is very energetic and powerful. The thunderous beats of the drummer provide a pounding rhythm. A guitar solo melody emerges from the chaotic background of the chords. The chord progression is A, D, E. The tempo of the song is 160 bpm.
6	This is an intense and loud punk song with guitars and drums. It is instrumental only. It is very energetic and powerful. The thunderous beats of the drummer provide a pounding rhythm. A guitar solo melody emerges from the chaotic background of the chords.The chord progression is C, B, A, G. The tempo of the song is 100 bpm.
7	A slow paced jazz song played by a saxophone, piano, guitar and drums follows a chord progression of Em7b5, A7, Dm7. The pianist produces delicate harmonies and subtle embellishments. The drummer provides a brushed rhythm. The guitar strums softly, while the saxophone plays a solo over the chords. This song goes at 80 beats per minute.
8	A slow paced jazz song played by a saxophone, piano, guitar and drums follows a chord progression of B7, G7, E7, C7. The drummer provides a brushed rhythm. The guitar strums softly, while the saxophone plays a solo over the chords. This song goes at 115 beats per minute.
9	This is a techno piece with drums and beats and a leading melody. A synth plays chords. The music kicks off with a powerful and relentless drumbeat. Over the pounding beats, a leading melody emerges. It has strong danceability and can be played in a club. The tempo is 120 bpm. The chords played by the synth are Am, Cm, Dm, Gm.
10	This is a techno piece with drums and beats and a leading melody. A synth plays chords. The music kicks off with a powerful and relentless drumbeat. Over the pounding beats, a leading melody emerges. It has strong danceability and can be played in a club. The tempo is 160 bpm. The chords played by the synth are C, F, G.
11	A horn and a bass guitar groove to a reggae tune. The combination of the horn section's catchy melodies and the buoyant bassline creates an irresistible groove. The bassline is bouncy and lively. The song is played at the pace of Adagio. An electric keyboard plays the chords Am, Dm, G, C.
12	A horn and a bass guitar groove to a reggae tune. The combination of the horn section's catchy melodies and the buoyant bassline creates an irresistible groove. The bassline is bouncy and lively. The song is played at the pace of Moderato. An electric keyboard plays the chords E, B, A.
13	This is a metal song with a guitar, drums and bass guitar. The bassist, wielding a solid-bodied bass guitar, adds depth and power to the sonic landscape. The drummer commands a massive drum kit. With a relentless force, they pound out thunderous rhythms, driving the music forward. As the song begins, the guitar roars to life, delivering a series of distorted chords. It follows the chords of Em, C, G, D. The tempo is 120 bpm.
14	This is a metal song with a guitar, drums and bass guitar. The bassist, wielding a solid-bodied bass guitar, adds depth and power to the sonic landscape. The drummer commands a massive drum kit. With a relentless force, they pound out thunderous rhythms, driving the music forward. As the song begins, the guitar roars to life, delivering a series of distorted chords. It follows the chords of A, F#m, D, E. The tempo is 170 bpm.
15	A man sings a captivating folk song while strumming chords on an acoustic guitar. This fits a campfire evening happening. The chord progression is G, C, D, G. The tempo is 100 beats per minute.
16	A man sings a captivating folk song while strumming chords on an acoustic guitar. This fits a campfire evening happening. The chord progression is Am, Em, Dm, Am. The tempo is 70 beats per minute.
17	This is a classical music piece played by a string trio. The instruments involved are violin, viola, and cello. The violin plays the lead melody. The cello's soulful and melodic contributions add depth and gravitas to the performance. The time signature is ¾. The tempo of this song is Presto. The chord sequence is E, C#m, A, B.
18	This is a classical music piece played by a string trio. The instruments involved are violin, viola, and cello. The violin plays the lead melody. The cello's soulful and melodic contributions add depth and gravitas to the performance. The time signature is 4/4. The tempo of this song is Andante. The chord sequence is Am, Dm, E7, Am.
19	This is a pop song with a female singer singing the leading melody and synthesizers looping samples as background. These loops provide the song's electronic foundation, creating a rich and layered sonic landscape. The charismatic female singer has a dynamic and emotive voice. The tempo is Moderato. The chord sequence is C, G, Am, F.
20	This is a pop song with a female singer singing the leading melody and synthesizers looping samples as background. These loops provide the song's electronic foundation, creating a rich and layered sonic landscape. The charismatic female singer has a dynamic and emotive voice. The tempo is Presto. The key is A minor and the chord sequences are Am, Dm, E.

Table 7: Custom opposing captions created for the control experiment.

K Additional Examples of Generated Music

Here we show additional samples generated from pre-trained Tango fine-tuned on MusicCaps, Tango finetuned on MusicBench, Mustango, MusicGen-M, and AudioLDM2, all generated from the same prompts.

Prompt: A horn and a bass guitar groove to a reggae tune. The combination of the horn section's catchy melodies and the buoyant bassline creates an irresistible groove. The bassline is bouncy and lively. The song is played at the pace of Adagio. An electric keyboard plays the chords Am, Dm, G, and C.

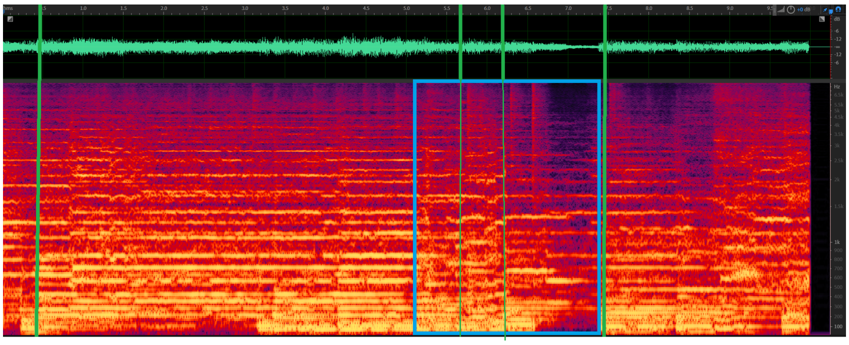


Figure 7: Mel-spectrogram of a reggae sample generated by pre-trained Tango fine-tuned on MusicCaps with vertical lines showing perceived chord starts. The blue box shows an area of dissonance in the music. Overall, the audio is a bit noisy.

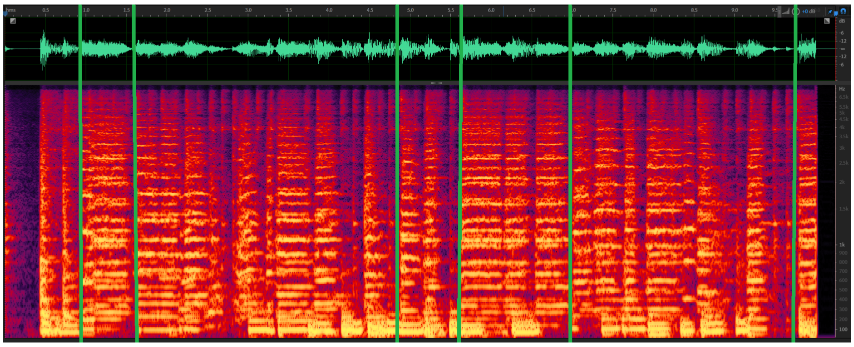


Figure 8: Mel-spectrogram of a reggae sample generated by pre-trained Tango fine-tuned on MusicBench with vertical lines showing perceived chord starts. There are too many chords here.

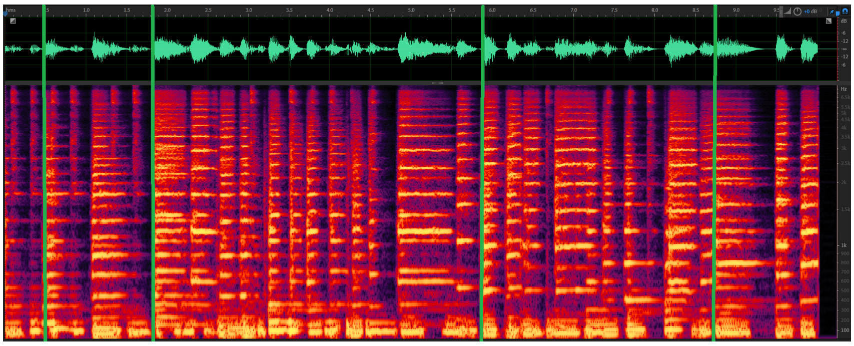


Figure 9: Mel-spectrogram of a reggae sample generated by Mustango with vertical lines showing perceived chord starts. The chords match the prompt.

Prompt: This is a metal song with a guitar, drums and bass guitar. The bassist, wielding a solid-bodied bass guitar, adds depth and power to the sonic landscape. The drummer commands a massive drum kit. With a relentless force, they pound out thunderous rhythms, driving the music forward. As the song begins, the guitar roars to life, delivering a series of distorted chords. It follows the chords of Em, C, G, D. The tempo is 120 bpm.

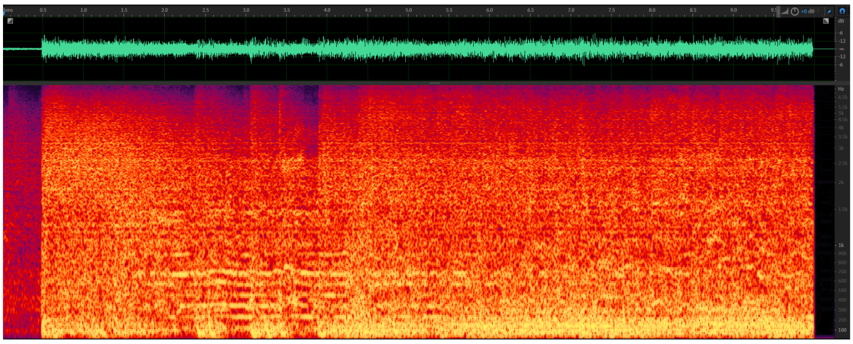


Figure 10: Mel-spectrogram of a metal song sample generated by pre-trained Tango fine-tuned on MusicCaps. It is very noisy from the very start.

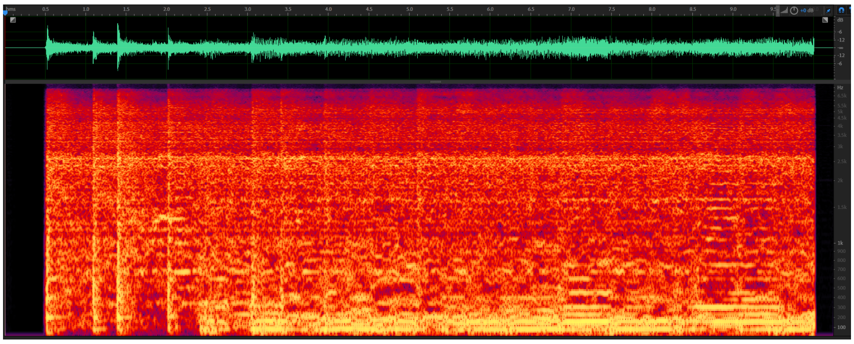


Figure 11: Mel-spectrogram of a metal song sample generated by pre-trained Tango fine-tuned on MusicBench. The song starts with 4 beats from the drummer, but there is a bit of noise from the start.

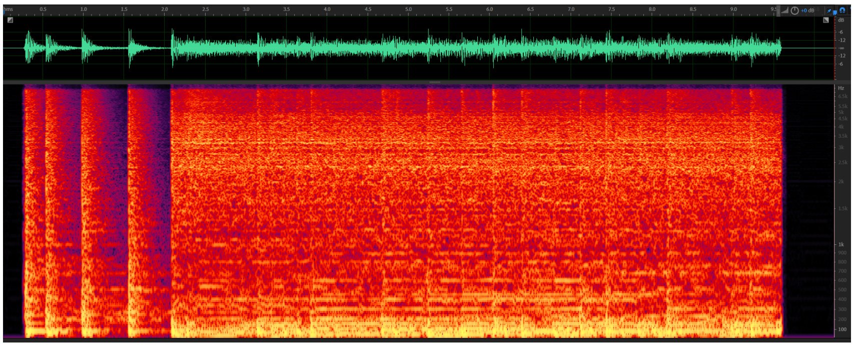


Figure 12: Mel-spectrogram of a metal sample generated by Mustango. The song starts with 4 distinguishable beats from the drummer, then the guitars join.

Prompt: This is a classical music piece. There are violins playing a lead theme, with a double bass and cymbals in the background. It is a melancholic, rather sad piece. The music builds up in volume gradually. The key is A minor. The chord sequence is Am, C, Am.

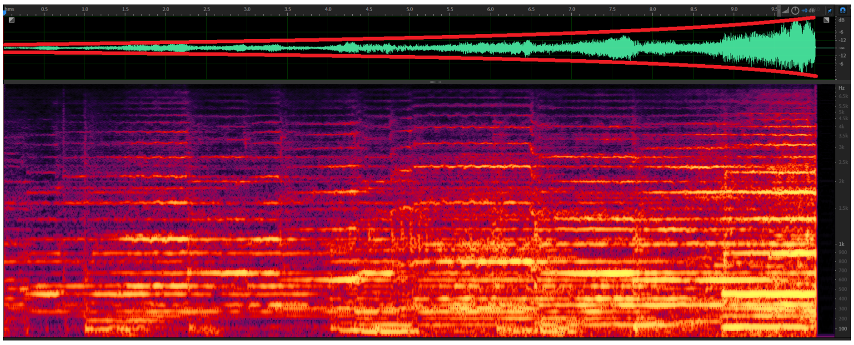


Figure 13: Mel-spectrogram of a classical music piece generated by `Mustango`. The effect of gradual volume increase (crescendo) is apparent (red color envelope around waveform).

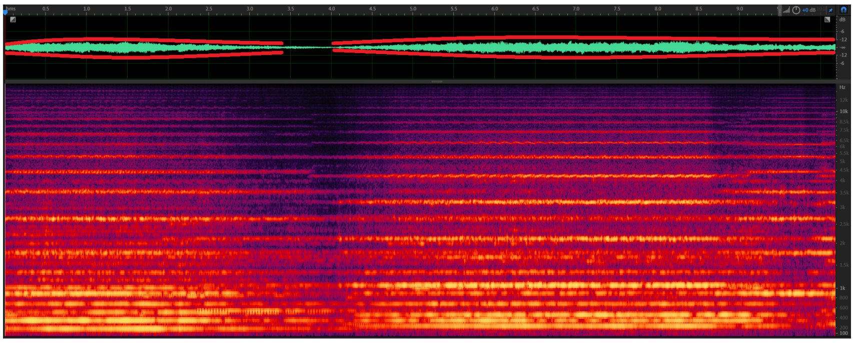


Figure 14: Mel-spectrogram of a classical music piece generated by `MusicGen-M`. The effect of gradual volume increase (crescendo) is not clear (red color envelope around waveform).

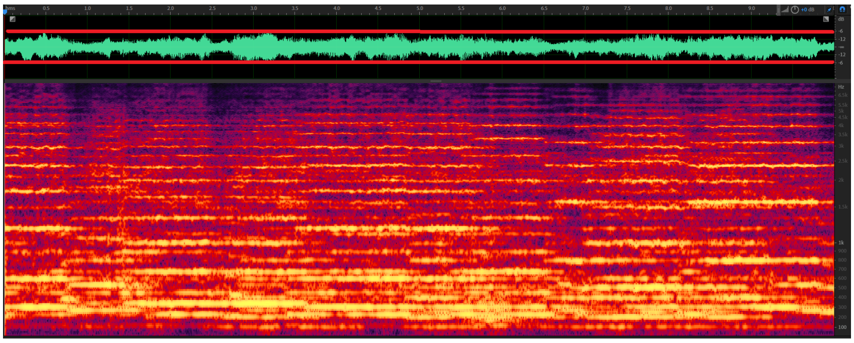


Figure 15: Mel-spectrogram of a classical music piece generated by `AudioLDM-2`. The effect of gradual volume increase (crescendo) is not present (red color envelope around waveform).