

# JOINT AUDIO AND SYMBOLIC CONDITIONING FOR TEMPORALLY CONTROLLED TEXT-TO-MUSIC GENERATION

Or Tal<sup>\*1,2</sup>      Alon Ziv<sup>\*1</sup>      Itai Gat<sup>2</sup>  
 Felix Kreuk<sup>2</sup>      Yossi Adi<sup>1,2</sup>

<sup>1</sup>The Hebrew University of Jerusalem

<sup>2</sup>Meta, FAIR Team

{or.tal1, alon.ziv1}@mail.huji.ac.il

## ABSTRACT

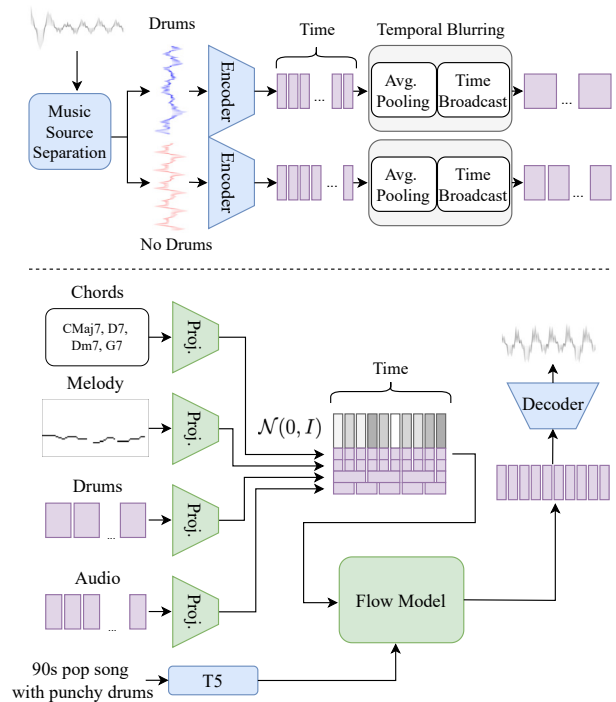
We present JASCO, a temporally controlled text-to-music generation model utilizing both symbolic and audio-based conditions. JASCO can generate high-quality music samples conditioned on global text descriptions along with fine-grained local controls. JASCO is based on the Flow Matching modeling paradigm together with a novel conditioning method that allows for both locally (e.g., chords) and globally (text description) controlled music generation. Specifically, we apply information bottleneck layers in conjunction with temporal blurring to extract relevant information with respect to specific controls. This allows the incorporation of both symbolic and audio-based conditions in the same text-to-music model. We experiment with various symbolic control signals (e.g., chords, melody), as well as with audio representations (e.g., separated drum tracks, full-mix). We evaluate JASCO considering both generation quality and condition adherence using objective metrics and human studies. Results suggest that JASCO is comparable to the evaluated baselines considering generation quality while allowing significantly better and more versatile controls over the generated music. Samples are available on our demo page <https://pages.cs.huji.ac.il/adiyoss-lab/JASCO>

## 1. INTRODUCTION

Conditional music generation has shown a great improvement in recent years, specifically in the task of *text-to-music* generation [1–6]. Such advancements in music generation hold great potential to empower content creators, advertisers, and video game designers. Though presenting highly realistic music samples, most of the prior work is focused on global conditioning only. Such methods mainly consider textual descriptions or melody in the form

\*Equal contribution

© O. Tal, A. Ziv, I. Gat, F. Kreuk, and Y. Adi. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** O. Tal, A. Ziv, I. Gat, F. Kreuk, and Y. Adi, “JOINT AUDIO AND SYMBOLIC CONDITIONING FOR TEMPORALLY CONTROLLED TEXT-TO-MUSIC GENERATION”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.



**Figure 1.** Top figure presents the temporal blurring process, showcasing source separation, pooling and broadcasting. Bottom figure presents a high level presentation of JASCO. Conditions are first being projected to low dimensional representation and are concatenated over the channel dimensions. Green blocks have learnable parameters while blue block are frozen.

of spectral features [3]. However, when considering music production, global controls may not be enough. During the creative process, professional musicians often use chords, melodies, or audio prompts, at the local level, rather than global descriptions. As a result, current models may be limited in their relevancy for music creators.

More recently, several works study text-to-music generation using temporally aligned controls. The authors in [7] suggest adding symbolic beat and dynamics conditions on top of the previously explored melody conditioning. The authors in [8] further explore musical structure conditioning, such as A-part and B-part. Unlike these works, the proposed method provides local controls considering both

symbolic representation and raw audio together with a global textual description. When considering music editing, the authors in [9] propose leveraging chord progression to guide the generation process towards the harmony of the inputs signal. For that, the authors extract an internal representation from stemmed data using a pre-trained chord classification model. The proposed method is different as we focus on generating full musical pieces rather than editing a given one. Specifically, we allow symbolic chord progression conditioning during inference time.

In this work, we present JASCO, a locally controlled Joint Audio and Symbolic COnditioning text-to-music model. JASCO uses time-aligned controls, namely audio prompts, melodies and chord progressions, comprised of either symbolic signals or raw waveforms. We relieve the need for either studio quality stemmed data or supervised datasets by using off-the-shelf pre-trained models to automatically extract the relevant information. JASCO is based on the Flow-Matching [10] modeling paradigm. Figure 1 provides a high level description of the proposed method. Results suggest that JASCO achieves comparable performance in terms of generation quality w.r.t the evaluated baselines while allowing significantly richer set of controls. The main contributions of this work are as follows: (i) We introduce a simple yet effective approach for audio conditioning with high temporal-adherence. (ii) We offer specific evaluation metrics to measure the alignment and accuracy of our suggested controls. (iii) We provide a thorough analysis on the components composing JASCO and compare to several baselines.

## 2. BACKGROUND

**Audio Representation.** Modern audio generative models mostly operate on a latent representation of the audio, commonly obtained from a compression model [11–13]. Compression models such as [14] employ Residual Vector Quantization (RVQ) which results in several parallel streams. Each stream is comprised of discrete tokens originating from different learned codebooks.

Specifically, the authors in [14] introduced EnCodec, a convolutional auto-encoder with a latent space quantized using RVQ [15], and an adversarial reconstruction loss. Given a reference audio signal  $x \in \mathbb{R}^{D \cdot f_s}$  with  $D$  the audio duration and  $f_s$  the sample rate, EnCodec first encodes it into a continuous latent tensor  $z \in \mathbb{R}^{D \cdot f_r \times N_{\text{enc}}}$  with a frame rate  $f_r \ll f_s$  and  $N_{\text{enc}} = 128$ . Then,  $z$  is quantized into  $q \in \{1, \dots, N\}^{D \cdot f_r \times K}$ , with  $K$  being the number of codebooks used in RVQ and  $N$  being the codebook size. After quantization, we are left with  $K$  discrete token sequences, each of length  $T = D \cdot f_r$ , representing the audio signal. In RVQ, each quantizer encodes the quantization error left by the previous quantizer, thus quantized values for different codebooks are in general dependent, where the first codebook holds most of the information. Finally, the quantized representation is decoded back to a time domain signal using the decoder network applied to the sum of the representations learned by the different codebooks.

In JASCO, we use the continuous tensor  $z$  as the latent representation, while leveraging the discrete representation  $q$  for audio conditioning.

**Flow Matching.** The Flow Matching modeling paradigm [10] was recently found to provide impressive results on image [10], speech [16] and environmental sound generation [17]. More specifically, Conditional Flow Matching (CFM) is a novel training technique for Continuous Normalizing Flow models [18], that captures the continuous transformation paths of samples from a basic prior distribution, usually standard normal  $\mathcal{N}(0, 1)$ , to their counterparts in a target data distribution,  $\mathcal{S}$ . The position on this path is denoted by a time parameter  $t$ , starting from the prior state at  $t = 0$  and ending at the data state at  $t = 1$ .

In this work, we focus on Optimal Transport (OT) paths as defined in [10]. The model is trained to predict the vector field of the continuous latent audio variable  $z$ , given  $t$  and a set of conditions  $\mathbf{Y}$ . Formally, the model minimizes the regression loss

$$\mathcal{L}_{\text{CFM}}(\theta; z_0, z_1, t | \mathbf{Y}) = \|v_\theta(z, t | \mathbf{Y}) - (z_1 - (1 - \sigma_{\min}) \cdot z_0)\|^2, \quad (1)$$

where  $z_0 \sim \mathcal{N}(0, I)$  is a sampled noise,  $z_1 \sim \mathcal{S}$  is the latent representation of a data sample, and

$$z = (1 - (1 - \sigma_{\min}) \cdot t) \cdot z_0 + t \cdot z_1, \quad (2)$$

is an interpolation between the noise and the data sample. For numerical stability, we use a small value  $\sigma_{\min} = 10^{-5}$  in both terms. During inference we follow an iterative process, starting with the prior noise  $z \leftarrow z_0 \sim \mathcal{N}(0, 1)$  and with  $t = 0$ . In each step, we translate the estimated vector field  $v_\theta(z, t | \mathbf{Y})$  into an updated latent sequence  $z$ , and gradually converge toward the data distribution.

## 3. METHOD

Given a textual description, and a set of temporal conditions - such as melody, chord progression or drum recording, our goal is to produce high-quality samples that are musically aligned with the given controls, while complying to the arrangement description provided in the text.

JASCO tackles the aforementioned problem by a CFM model, operating on the continuous latent space of EnCodec. JASCO is conditioned on low-dimensional embeddings of melody, chords and audio signals, together with a T5 embedding of the textual description. All local controls are concatenated to the model’s input across the feature dimension, while text is being passed via cross attention. To diminish timbre-related information, JASCO further applies temporal blurring to the audio-based controls, as well as band-pass filtering. See Figure 1 for a visual description, and Section 3.1 for detailed information.

### 3.1 Temporal Controls

**Symbolic.** We use Chordino<sup>1</sup> chord progression model to extract an integer categorical chord label sequence, and a

<sup>1</sup> <https://github.com/ohollo/chord-extractor>

pretrained multi-F0 classifier [19] to obtain melody scores per time step. We resample all features to match EnCodec’s frame rate using nearest-interpolation for chords and linear-interpolation for melody. For Chords, we use a learned embedding table to map the raw integer sequence, denoted as  $\mathbf{c}_{\text{crd}}$ , to its corresponding condition matrix in shape  $T \times d_{\text{crd}}$ . For Melody, we zero out values with a score lower than a pre-defined threshold (0.5). Then, we select the maximal non-zero score per time step from the remaining values, and set it to 1 while setting the rest to 0. This yields a binary matrix  $\mathbf{c}_{\text{mld}} \in \{0, 1\}^{D \cdot f_r^{\text{mld}} \times N_{\text{mld}}}$ . Finally, we linearly project the binary matrix and obtain the melody condition representation in shape  $T \times d_{\text{mld}}$ . We use  $N_{\text{mld}} = 53$  (corresponding to G2-B7 notes), and  $d_{\text{crd}} = d_{\text{mld}} = 16$ .

**Audio.** We consider general audio and separated drum stems. We use a pretrained source separation model [20], to extract the drum stem from a source audio. We pass the waveform through EnCodec to obtain the corresponding quantized discrete representation  $\mathbf{q}$ . We then convert the first token stream back to its continuous latent representation, using EnCodec’s first codebook while discarding all other streams, yielding  $\mathbf{c}_{\text{aud}}, \mathbf{c}_{\text{drm}} \in \mathbb{R}^{T \times N_{\text{enc}}}$ . We chose to use only the first codebook stream to further discard timbre information, stressing the forced information-bottleneck further. Following that, we apply temporal blurring to the reconstructed latent. First, we apply average pooling using non-overlapping windows along the temporal axis. Then, we broadcast the signal to its original temporal dimension. Finally, we linearly project the blurred condition to a low dimensional feature space and obtain the condition matrix. For general audio, we use a window size of 5 and output dimension of 1, while for drums we use a window size of 3 and output dimension of 2.

**Inpainting and Outpainting.** In/Out-painting is the task of filling in a masked region, where in/out refers to the masked segment position in the sequence, be it at the middle (in) or at the end (out). Following prior work [5], we add in/out-painting as an additional condition to the model. We randomly choose between inpainting/outpainting, and mask a random segment of 40-90% from the reference waveform. Then, we use the raw EnCodec latent representation of the masked waveform  $\mathbf{c}_{\text{iop}} \in \mathbb{R}^{T \times N_{\text{enc}}}$  as the condition, with no learned projection.

### 3.2 Model and Optimization

Similarly to prior work [17], our CFM model consists of a Transformer, with U-Net-like residual connections. We replace the standard residual addition with channel-wise concatenation followed by a linear projection. We use learned convolutional positional encoding [21] as well as symmetric bi-directional ALiBi self-attention biases [22]. We use a model scale of 330M parameters, with 24 Transformer layers, 16 attention heads, embedding dimensionality of 1024 and a feed-forward dimension of 4096.

We train our model using the  $\mathcal{L}_{\text{CFM}}$  objective as defined in Section 2. We further experiment with non-uniform loss

weighting as function of  $t$ , and find the following formulation to produce the best overall sample quality:

$$\mathcal{L}_{\text{WeightedCFM}} = \sum_{\substack{t \sim \mathcal{U}(0,1) \\ \mathbf{z}_0 \sim \mathcal{N}(0,1) \\ \mathbf{z}_1 \sim \mathcal{S}}} (1+t) \cdot \mathcal{L}_{\text{CFM}}(\theta; \mathbf{z}_0, \mathbf{z}_1, t | \mathbf{Y}), \quad (3)$$

where  $\mathbf{Y} = \{\mathbf{c}_{\text{crd}}, \mathbf{c}_{\text{mld}}, \mathbf{c}_{\text{aud}}, \mathbf{c}_{\text{drm}}, \mathbf{c}_{\text{iop}}\}$ . We provide an ablation study for this scheme in Section 5.

### 3.3 Inference

During inference, as in [10], we use *dopri5* [23], an off-the-shelf numerical ODE solver, to iteratively solve for  $\mathbf{z}$  given the estimated vector field  $v_\theta$ . Specifically, at each iteration the solver determines the increment to the time parameter  $t$ , resulting in a dynamic scheduling for the inference process. The process halts when an acceptance criterion is met, defined by an error approximation of the solver and a tolerance parameter provided by the user.

**Multi-Source Classifier Free Guidance.** We employ classifier-free guidance (CFG) [24] for the conditional vector field estimation  $v_\theta(\mathbf{z}, t | \mathcal{Y})$ . Since our set of conditioning signals combines both global and local concepts, we further experiment with multi source CFG. While prior work [25] suggest a separate evaluation for each condition, we evaluate the model considering all and partial conditions. During each inference step, we obtain an estimated vector field for each set of conditions  $\mathcal{Y} \in \{\{\text{local}\}, \{\text{text}\}, \{\text{local}, \text{text}\}\}$ . The resulting CFG formulation then follows:

$$\text{CFG}(v_\theta, \mathbf{z}, t) = (1 - \sum_{c \in \mathcal{Y}} \alpha_c) v_\theta(\mathbf{z}, t) + \sum_{c \in \mathcal{Y}} \alpha_c v_\theta(\mathbf{z}, t | c). \quad (4)$$

When following the standard CFG setup ( $\alpha_{\text{text}} = \alpha_{\text{local}} = 0$ ), we observe that the model adheres to the temporal condition while ignoring instrumentation information provided in the text prompt. To increase text influence on guidance, we set a positive weight to the text-only term  $\alpha_{\text{text}} > 0$ . We found that  $\alpha_{\text{text}} = 0.5, \alpha_{\text{local}} = 0, \alpha_{\text{local}, \text{text}} = 1.5$  offer a good trade-off between audio quality, text alignment and temporal controls adherence.

## 4. EXPERIMENTAL SETUP

**Implementation Details.** We follow the same experimental setup as in [3, 6], and use a training dataset consisting of 20K hours of licensed music from the Shutterstock<sup>2</sup> and Pond5<sup>3</sup> data collections with 25K and 365K instrument-only music tracks, respectively. We additionally include a set of proprietary data consisting of 10K high-quality tracks. All datasets are sampled at 32kHz, paired with textual descriptions. We present results on the MusicCaps benchmark [1], comprising 5.5K 10-second samples together with an in-domain test set of 528 tracks.

<sup>2</sup> [shutterstock.com/music](https://shutterstock.com/music)

<sup>3</sup> [pond5.com](https://pond5.com)

We use the official EnCodec model provided by [3, 12], with a frame rate of 50 Hz, and 4 codebooks, each with a size of 2048. For text representation we use a pretrained T5 model [26]. For melody extraction we use the pretrained deep salience multi-F0 detector<sup>4</sup>, for chords extraction we use Chordino, while for drum track extraction we use the Hybrid Demucs model [27].

All single condition models were trained with 40% condition dropout, and in the multi-condition experiments we train the models with 20% condition dropout for all conditions. In the remaining 80% we set 50% dropout for each of the conditions independently excluding the in/out-painting, for which we set 70% dropout.

We experiment with multi-source CFG coefficients in  $(\alpha_{\text{text}}, \alpha_{\text{local}}, \alpha_{\text{text,local}}) \in \{0.0, 0.5\} \times \{0.0, -0.5\} \times \{1.5, 2.0\}$  and report the best overall configuration. All models were trained for 500k steps over audio segments of 10 seconds, with a batch size of 336. We use Adam [28] optimizer with linear learning rate warm-up up to a peak of  $10^{-4}$  during the first 5k steps, followed by a linear decay, and a gradient clipping with a norm threshold of 0.2.

#### 4.1 Evaluation Metrics

We perform a thorough empirical evaluation, using both objective metrics and human studies. We evaluate JASCO on several temporal alignment aspects, namely harmonic matching, rhythmic alignment and melody preservation. Additionally, we measure audio quality and text adherence.

**Objective Evaluations.** We evaluate our method with widely used metrics, namely Fréchet Audio Distance (FAD), Kullback-Leiber Divergence (KL) and CLAP score (CLAP), as well as more specific metrics designed to quantify the adherence of our suggested controls. We report FAD [29] using the official tensorflow implementation where a low FAD score indicates that the generated audio is associated with higher quality. Following [3, 12], we use an audio classifier [30] to compute the KL-divergence over the probabilities of the labels between the original and the generated music. The generated music is expected to share similar concepts with the reference music when the KL is low. Last, CLAP score [31, 32] is computed between the track description and the generated audio, measuring audio-text alignment. We use the official pretrained CLAP model<sup>5</sup>. To evaluate melody compatibility, similar to [3] we use a cosine similarity metric on either a simple quantized chroma representation, or multi-octave melody representation obtained from a pretrained multi-F0 classifier [19]. For beat adherence, as in [7] we evaluate the onset F1 score using *mir eval*<sup>6</sup> considering a 50ms tolerance margin around classified onsets in the reference signal. Lastly, to evaluate chord progression, we use the Chordino model to extract the chord progression from both the reference and the generated signals and compute the intersection over union (IOU) score between the two.

| Model           | FAD↓  | CLAP↑ | Mel Sim.↑ | Mel Acc.↑ |
|-----------------|-------|-------|-----------|-----------|
| MusicGen        | 5.90  | 0.29  | 0.61      | 44.0      |
| MusicControlNet | 10.81 | 0.22  | -         | 47.1      |
| JASCO           | 6.05  | 0.26  | 0.67      | 49.1      |

**Table 1.** Melody conditioning evaluation over MusicCaps. We evaluated MusicGen with 300M parameters.

**Human Study.** We request raters to evaluate three aspects of given audio samples: (i) overall quality; (ii) similarity to text description; and (iii) adherence to either melody or rhythmic pattern from a reference recording. Raters were instructed to rate the recordings on a scale between 0-100 where higher is better. Raters were recruited using the Amazon Mechanical Turk platform. We evaluate randomly sampled files, where each sample was evaluated by at least 5 raters. We use the CrowdMOS package [33] to filter noisy annotations and outliers. We remove annotators who did not listen to the full recordings, annotators who rate the reference recordings less than 90, and the rest of the recommended recipes from [33]. Similarly to [3], for a fair comparison, all samples are normalized at -14dB LUFS [34]. Overall, we 179 raters evaluated the generation quality, 121 raters evaluated the text relevancy, 159 raters evaluated the adherence to rhythm patterns using drum conditioning, and 142 raters evaluated melody conditioning.

## 5. RESULTS

**Melody Conditioning.** We start by evaluating the proposed method considering melody conditioning. We compare JASCO to MusicGen [3] and MusicControlNet [7]. For a fair comparison, we train MusicGen (300M) on 10 second music segments using Audioscraft<sup>7</sup> repository, considering text and melody conditions. For comparison compatibility with [7] we compute melody accuracy score on both JASCO and MusicGen. We experiment with melody conditioning using the commonly used 12-bins chroma representation which is octave invariant. Results are presented in Table 1.

Results suggest that JASCO surpasses the evaluated baselines w.r.t melody adherence. When considering melody accuracy, JASCO provides better alignment to the conditioning melody. Notice, we hypothesize this is due to the conditioning method: both MusicGen and MusicControlNet inject conditions as an additive bias (i.e., cross-attention and zero-convolutions), this is in contrary to JASCO which follows the concatenation approach for melody conditioning (see Section 6 for more experiments).

**Local Controls.** We train a single-condition variant for each observed condition-type as well as two multi-condition models. Under the multi-condition setup, we train models with Drums tracks passed through a Band-Pass-Filter (BPF) over 200-800 Hz frequency range, and

<sup>4</sup> [github.com/rabitt/ismir2017-deepsalience](https://github.com/rabitt/ismir2017-deepsalience)

<sup>5</sup> [github.com/LAION-AI/CLAP](https://github.com/LAION-AI/CLAP)

<sup>6</sup> [github.com/craffel/mir\\_evaluators](https://github.com/craffel/mir_evaluators)

<sup>7</sup> <https://github.com/facebookresearch/audiocraft/blob/main/docs/MUSICGEN.md>

| Local Controls |              |              |              | Objective metrics (MusicCaps / Internal dataset) |                     |                     |                     |                  |                 |                 |
|----------------|--------------|--------------|--------------|--|---------------------|---------------------|---------------------|------------------|-----------------|-----------------|
| Aud            | Drum         | Crds         | Mld          | Mld (clf) sim. $\uparrow$                        | Mld sim. $\uparrow$ | Onset F1 $\uparrow$ | Crds IOU $\uparrow$ | FAD $\downarrow$ | KL $\downarrow$ | CLAP $\uparrow$ |
| -              | -            | -            | -            | 0.13 / 0.13                                      | 0.09 / 0.09         | 0.34 / 0.41         | 0.09 / 0.07         | 6.04 / 0.90      | 1.46 / 0.70     | 0.27 / 0.36     |
| $\checkmark$   | -            | -            | -            | 0.33 / 0.34                                      | 0.38 / 0.47         | 0.62 / 0.81         | 0.23 / 0.27         | 4.47 / 0.86      | 0.92 / 0.81     | 0.30 / 0.31     |
| no drm         | -            | -            | -            | 0.21 / 0.22                                      | 0.38 / 0.31         | 0.62 / 0.58         | 0.23 / 0.18         | 5.68 / 0.92      | 1.79 / 0.75     | 0.19 / 0.33     |
| -              | $\checkmark$ | -            | -            | 0.13 / 0.13                                      | 0.09 / 0.10         | <b>0.62 / 0.73</b>  | 0.09 / 0.08         | 5.85 / 0.94      | 1.68 / 0.78     | 0.23 / 0.35     |
| -              | BPF          | -            | -            | 0.13 / 0.13                                      | 0.10 / 0.10         | 0.45 / 0.74         | 0.10 / 0.07         | 6.31 / 1.61      | 1.52 / 0.65     | 0.26 / 0.37     |
| -              | -            | $\checkmark$ | -            | 0.21 / 0.25                                      | 0.22 / 0.29         | 0.24 / 0.13         | <b>0.59 / 0.61</b>  | 7.23 / 0.95      | 1.16 / 0.68     | 0.28 / 0.36     |
| -              | -            | -            | $\checkmark$ | <b>0.67 / 0.64</b>                               | <b>0.41 / 0.35</b>  | 0.37 / 0.57         | 0.31 / 0.27         | 6.96 / 1.05      | 1.32 / 0.63     | 0.27 / 0.35     |
| -              | BPF          | $\checkmark$ | $\checkmark$ | 0.68 / 0.69                                      | 0.44 / 0.46         | 0.63 / 0.66         | 0.50 / 0.53         | 6.42 / 1.15      | 1.22 / 0.50     | 0.28 / 0.37     |
| no drm         | BPF          | $\checkmark$ | $\checkmark$ | 0.71 / 0.68                                      | 0.50 / 0.55         | 0.54 / 0.75         | 0.51 / 0.55         | 4.78 / 0.80      | 0.93 / 0.41     | 0.30 / 0.37     |

**Table 2.** Objective local controls experiment, observing all suggested controls w.r.t a zero hypothesis (no local controls).

| Model     | Cond. | Q               | T              | M                              | D                              |
|-----------|-------|-----------------|----------------|--------------------------------|--------------------------------|
| Reference | -     | 92.7 $\pm$ 0.66 | 93.7 $\pm$ 0.8 | 96.3 $\pm$ 0.6                 | 97.1 $\pm$ 0.6                 |
| MusicGen  | T     | 84.4 $\pm$ 0.8  | 84.5 $\pm$ 0.9 | 81.5 $\pm$ 1.3                 | 82.1 $\pm$ 1.0                 |
| JASCO     | T     | 83.3 $\pm$ 0.7  | 80.3 $\pm$ 1.3 | 79.7 $\pm$ 1.5                 | 81.5 $\pm$ 1.1                 |
| MusicGen  | T & M | 84.7 $\pm$ 0.7  | 82.5 $\pm$ 1.1 | 83.6 $\pm$ 1.1                 | 82.7 $\pm$ 0.9                 |
| JASCO     | T & M | 84.1 $\pm$ 0.7  | 81.2 $\pm$ 1.2 | <b>89.3<math>\pm</math>0.7</b> | 80.6 $\pm$ 1.2                 |
| JASCO     | T & D | 85.5 $\pm$ 0.8  | 84.1 $\pm$ 1.1 | 81.9 $\pm$ 1.4                 | <b>89.5<math>\pm</math>0.7</b> |

**Table 3.** Human evaluation results. Observing general quality (Q), text match (T) melody match (M) and drums match (D). Evaluated on a 0-100 scale (higher is better).

Audio condition excluding drums. This was found to better disentangle Drums and Audio conditions in preliminary experiments, and allows users to provide different drum beats than the one presented in the Audio. When applying Audio/Drums conditions, we evaluate Melody, Onset F1, and Chord IoU using the reference audio as a condition, while for the computation FAD, KL, and CLAP scores we use a randomly selected audio from the test set.

As there are no open-source relevant baselines available, we compare the proposed method against a text-only condition model. We perform experiments using both the open source MusicCaps dataset, and an internal proprietary dataset, highlighting our model performance on diverse, high quality recordings. Table 2 summarizes the results.

Results depict a systematic improvement considering local control adherence. For instance, chords conditioning shows apparent improvement in Chords IOU metric, improving from 0.09/0.07 to 0.59/0.61. In addition, in spite of being evaluated with randomly selected audio conditions, FAD, KL, CLAP scores mostly remain comparable w.r.t to the baseline. This highlights JASCO’s disentangling property as local controls metrics improve while text adherence and audio quality metrics stay roughly the same.

The lower section of the table presents multi-control setup results. This section draws a similar trend to the single control setups, allowing for multiple controls while preserving FAD, KL, CLAP. This highlights JASCO’s ability to incorporate multiple controls simultaneously with no significant penalty to quality and text alignment.

**Human Study.** Lastly, we perform a human study in or-

der to validate both quality and text alignment as well as local control adherence. We evaluate JASCO vs MusicGen considering: (i) text only; and (ii) both text and melody. We additionally, provide results of the proposed method with text and drums conditions. Results seen on Table 3, indicate that JASCO achieve similar generation quality as MusicGen across all setups. As of text relevancy, MusicGen reaches superior performance to the proposed method, however, when considering melody conditioning, JASCO reaches significantly better scores. Lastly, when conditioned on drums, JASCO provides the best rhythmic pattern similarity scores. This highlights JASCO’s ability to provide better controls over the generated music without sacrificing quality and text alignment. Interestingly, after including melody or drums conditions, as expected, the relevant metrics are improving (i.e., melodic and rhythmic similarity) while the quality and text adherence remain comparable to the unconditioned model.

## 6. ANALYSIS

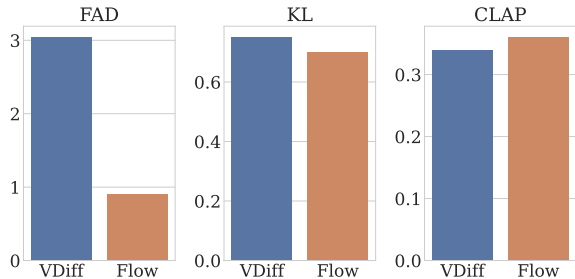
**Condition Injection Method.** We compare the proposed method to two widely used condition injection methods proposed in prior work. Specifically, we perform a controlled experiment in which we evaluate cross-attention as used in MusicGen, and zero-convolution as used in MusicControlNet, considering the same training configuration.

Results shown in Table 4 suggest that the temporal adherence using the concatenation method performs the best overall. This can be seen in both higher Chord IoU, as well as better FAD and KL, where CLAP was 0.36 for all methods. Additionally, the concatenation method allows training from scratch as opposed to zero-convolutions, in which we start from a pretrained model) without a significant increase in the number of trainable parameters.

**Flow vs. Diffusion.** Most of prior work on music generation is mainly based on Diffusion models [2, 4, 5, 35]. In this experiment we evaluate, under controlled settings, both Diffusion (v-Diffusion) and Flow Matching modeling approaches for music generation. We report FAD, KL, and CLAP scores. Results are depicted in Figure 2. As can be seen, the Flow Matching approach is superior across all metrics, with the biggest gap observed in FAD.

| Conditioning | Chord IOU $\uparrow$ | FAD $\downarrow$ | KL $\downarrow$ |
|--------------|----------------------|------------------|-----------------|
| Concat       | 0.6                  | 1.19             | 0.71            |
| Cross Attn.  | 0.59                 | 1.61             | 0.73            |
| Zero Conv    | 0.26                 | 1.64             | 0.74            |

**Table 4.** Ablation for conditioning method. evaluated on internal dataset. All models started from a text-to-music pretrained checkpoint and trained for 500K steps.



**Figure 2.** Comparison of v-Diffusion vs Flow Matching. We report FAD, KL, and CLAP on the internal dataset.

**The Effect of Weighted Loss.** Finally, we evaluate the effect of the proposed modification to the loss function as presented in Equation (3). We compare the proposed objective function against the loss as describe in Equation (1), considering FAD, KL, and CLAP scores in Table 5. Results suggest the new objective function modification improves the generation quality. It provides significantly better FAD while having comparable KL and CLAP scores.

### 7. RELATED WORK

**Flow Matching for Audio Generation.** Flow Matching [10] was recently studied for speech generation. A notable work in this context presented VoiceBox [16], a Flow Matching model, operating on spectrograms, for text-guided multilingual speech generation. More recently, AudioBox [17] was presented, in which self-supervised infilling objectives were leveraged to improve the generalization capabilities of VoiceBox. Similar to our model, AudioBox operates on the continuous latent representations of EnCodec [14]. Though the scope of audio modalities was extended in AudioBox to both speech and environmental sounds, applying a Flow Matching approach for music generation remained less explored.

**Temporally Controlled Music Generation.** Recent work offered several forms of temporally restrictive controls for music generation. Melody conditioned text-to-music was studied in MusicLM [1], in which a melody embedding was trained using a dedicated dataset consists of multiple cover versions of musical tracks paired with aligned singing and humming performances. In MusicGen [3] and Music ControlNet [7], the need for supervised data was relieved, and instead an unsupervised melody extraction was performed using the argmax note of the audio chromagram. Audio-to-audio setups were studied for drum gener-

| Weighted loss schedule | FAD $\downarrow$ | KL $\downarrow$ | CLAP $\uparrow$ |
|------------------------|------------------|-----------------|-----------------|
| $w(t) = 1$             | 1.73             | 0.71            | 0.38            |
| $w(t) = 1 + t$         | 0.99             | 0.73            | 0.37            |

**Table 5.** Ablation for loss weighting method. Evaluated on internal dataset. All models were trained for 500K steps.

ation conditioned on drumless track [36], accompaniment generation given singing voice [37], and single instrument generation given partial mix [25] [9]. Recently, generation conditioned on multiple symbolic controls was studied in Music ControlNet [7], a spectrogram diffusion text-to-music model, fine-tuned using the ControlNet scheme [38], to generation with melody, beat and dynamics controls. In DITTO [8], inference time optimization was explored, for tiding a text-to-music diffusion model to perform several tasks including inpainting, outpainting, loop generation, melody and dynamics conditioned generation, as well as conditioning on musical structures. In [39], classifier guidance was used to perform music inpainting, outpainting and style transfer given a pretrained unconditional latent diffusion model. Inpainting was further explored in [5], [40], and [41]. Style transfer was explored also in [42] and [9].

### 8. DISCUSSION

In this work we present JASCO, a temporally controlled text-to-music generation model, supporting both audio and symbolic conditioning. JASCO is based on the Flow Matching modeling paradigm operating over a dense music latent representation. Through extensive experimentation we empirically show JASCO generates high-fidelity samples that can be conditioned on global textual description together with harmony, melody, rhythmic patterns, and overall musical style. Results suggest JASCO provides comparable generation quality to the evaluated baselines while allowing significantly better control over generation.

**Limitations.** The main limitations of the proposed approach are: (i) Similarly to previous diffusion-based text-to-music models, the length of the generated samples is relatively short ( $\sim 10$  seconds) compared to the auto-regressive alternative. Although this can be extrapolated with overlaps, it may limit the capability of the model in capturing global structure in the generated music; (ii) although generating the whole sequence at once, generation time is slower than auto-regressive alternatives, while not supporting streaming capabilities.

**Future work.** For future work we intend to support additional controls, such as music dynamics, musical structure, etc. together with editing options, e.g., add or replace specific instrument in a given recording. We believe such a research direction, and specifically the proposed approach, holds great potential in empowering musicians, creators, and producers which require richer set of controls during their creative process.

## 9. ETHICAL STATEMENT

The use of large-scale generative models raises several ethical concerns. To mitigate at some of them, we first made sure all the data used for training our models was obtained legally through an agreement with Shutterstock. Another issue is the potential lack of diversity in the dataset, which predominantly consists of western-style music. However, we believe that the proposed method is not tied to any specific genera and can help expand the scope of applications to new datasets.

Moreover, generative models could potentially create an unbalanced competitive environment for artists, a problem that is yet to be solved. We are firm believers in the power of open research to provide all participants with equal opportunities to access these models. By introducing more sophisticated controls, like chords and rhythmic patterns as suggested in this work, we aspire to make these models beneficial for both amateurs and professional musicians.

## 10. REFERENCES

- [1] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, “Musiclm: Generating music from text,” 2023.
- [2] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank, J. Engel, Q. V. Le, W. Chan, Z. Chen, and W. Han, “Noise2music: Text-conditioned music generation with diffusion models,” 2023.
- [3] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” 2023.
- [4] F. Schneider, O. Kamal, Z. Jin, and B. Schölkopf, “Moûsai: Text-to-music generation with long-context latent diffusion,” 2023. [Online]. Available: <https://arxiv.org/abs/2301.11757>
- [5] P. Li, B. Chen, Y. Yao, Y. Wang, A. Wang, and A. Wang, “Jen-1: Text-guided universal music generation with omnidirectional diffusion models,” *arXiv preprint arXiv:2308.04729*, 2023.
- [6] A. Ziv, I. Gat, G. L. Lan, T. Remez, F. Kreuk, A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “Masked audio generation using a single non-autoregressive transformer,” *arXiv preprint arXiv:2401.04577*, 2024.
- [7] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, “Music controlnet: Multiple time-varying controls for music generation,” 2023.
- [8] Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, “Ditto: Diffusion inference-time t-optimization for music generation,” 2024.
- [9] B. Han, J. Dai, W. Hao, X. He, D. Guo, J. Chen, Y. Wang, Y. Qian, and X. Song, “Instructme: An instruction guided music edit and remix framework with latent diffusion models,” 2023.
- [10] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, “Flow matching for generative modeling,” 2023.
- [11] 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 Transactions on Audio, Speech, and Language Processing*, 2023.
- [12] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, “Audiogen: Textually guided audio generation,” *arXiv preprint arXiv:2209.15352*, 2022.
- [13] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, “Diffsound: Discrete diffusion model for text-to-sound generation,” *arXiv preprint arXiv:2207.09983*, 2022.
- [14] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” 2022.
- [15] 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*, 2021.
- [16] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, R. Moritz, M. Williamson, V. Manohar, Y. Adi, J. Mahadeokar, and W.-N. Hsu, “Voicebox: Text-guided multilingual universal speech generation at scale,” 2023.
- [17] A. Vyas, B. Shi, M. Le, A. Tjandra, Y.-C. Wu, B. Guo, J. Zhang, X. Zhang, R. Adkins, W. Ngan, J. Wang, I. Cruz, B. Akula, A. Akinyemi, B. Ellis, R. Moritz, Y. Yungster, A. Rakotoarison, L. Tan, C. Summers, C. Wood, J. Lane, M. Williamson, and W.-N. Hsu, “Audiobox: Unified audio generation with natural language prompts,” 2023.
- [18] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural ordinary differential equations,” 2019.
- [19] R. M. Bittner, B. McFee, J. Salamon, P. Q. Li, and J. P. Bello, “Deep salience representations for f0 estimation in polyphonic music,” in *International Society for Music Information Retrieval Conference*, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:4531539>
- [20] A. Défossez, “Hybrid spectrogram and waveform source separation,” in *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.
- [21] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” 2020.

- [22] O. Press, N. A. Smith, and M. Lewis, “Train short, test long: Attention with linear biases enables input length extrapolation,” 2022.
- [23] J. R. Dormand and P. J. Prince, “A family of embedded runge-kutta formulae,” *Journal of computational and applied mathematics*, vol. 6, no. 1, pp. 19–26, 1980.
- [24] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” 2022.
- [25] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J.-C. Wang, M. Avent, J. Chen, and D. Le, “Stemgen: A music generation model that listens,” 2024.
- [26] 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.
- [27] S. Rouard, F. Massa, and A. Défossez, “Hybrid transformers for music source separation,” 2022.
- [28] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.
- [29] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fr\`echet audio distance: A metric for evaluating music enhancement algorithms,” *arXiv preprint arXiv:1812.08466*, 2018.
- [30] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, “Efficient training of audio transformers with patchout,” *arXiv preprint arXiv:2110.05069*, 2021.
- [31] 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 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [32] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, “Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 13 916–13 932.
- [33] F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer, “Crowdmos: An approach for crowdsourcing mean opinion score studies,” in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 2416–2419.
- [34] T. Sugimoto, “Loudness-level-chasing algorithm for multiformat live audio production,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1290–1304, 2022.
- [35] S. Forsgren and H. Martiros, “Riffusion-stable diffusion for real-time music generation. 2022,” URL <https://riffusion.com/about>.
- [36] Y.-K. Wu, C.-Y. Chiu, and Y.-H. Yang, “Jukedrummer: Conditional beat-aware audio-domain drum accompaniment generation via transformer vq-vae,” 2022.
- [37] C. Donahue, A. Caillon, A. Roberts, E. Manilow, P. Esling, A. Agostinelli, M. Verzetti, I. Simon, O. Pietquin, N. Zeghidour, and J. Engel, “Singsong: Generating musical accompaniments from singing,” 2023.
- [38] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.
- [39] M. Levy, B. D. Giorgi, F. Weers, A. Katharopoulos, and T. Nickson, “Controllable music production with diffusion models and guidance gradients,” 2023.
- [40] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, “Vampnet: Music generation via masked acoustic token modeling,” 2023.
- [41] L. Lin, G. Xia, Y. Zhang, and J. Jiang, “Arrange, in-paint, and refine: Steerable long-term music audio generation and editing via content-based controls,” 2024.
- [42] N. Mor, L. Wolf, A. Polyak, and Y. Taigman, “A universal music translation network,” 2018.