# DITTO-2: DISTILLED DIFFUSION INFERENCE-TIME T-OPTIMIZATION FOR MUSIC GENERATION

**Zachary Novack**[1] **Julian McAuley**[1] **Taylor Berg-Kirkpatrick**[1] **Nicholas J. Bryan**[2]
[1]University of California – San Diego [2]Adobe Research
znovack@ucsd.edu, njb@ieee.org

## ABSTRACT

Controllable music generation methods are critical for human-centered AI-based music creation, but are currently limited by speed, quality, and control design trade-offs. Diffusion inference-time T-optimization (DITTO), in particular, offers state-of-the-art results, but is over 10x slower than real-time, limiting practical use. We propose **D**istilled **D**iffusion **I**nference-**T**ime $\boldsymbol{T}$-**O**ptimization (or DITTO-2), a new method to speed up inference-time optimization-based control and unlock faster-than-real-time generation for a wide-variety of applications such as music inpainting, outpainting, intensity, melody, and musical structure control. Our method works by (1) distilling a pre-trained diffusion model for fast sampling via an efficient, modified consistency or consistency trajectory distillation process (2) performing inference-time optimization using our distilled model with one-step sampling as an efficient surrogate optimization task and (3) running a final multi-step sampling generation (decoding) using our estimated noise latents for best-quality, fast, controllable generation. Through thorough evaluation, we find our method not only speeds up generation over 10-20x, but simultaneously improves control adherence and generation quality all at once. Furthermore, we apply our approach to a new application of maximizing text adherence (CLAP score) and show we can convert an unconditional diffusion model without text inputs into a model that yields state-of-the-art text control. Sound examples can be found at https://ditto-music.github.io/ditto2/.

## 1. INTRODUCTION

Audio-domain text-to-music (TTM) methods [1–6] have seen rapid development in recent years and show great promise for music creation. Such progress has been made possible through the development of diffusion models [7–9], language models [1, 2], latent representations of audio [10–13] and text-based control [4, 14]. Such control, however, can be limiting for creative human-centered AI music applications, motivating more diverse and advanced control (e.g., melody) that target's fine-grained aspects of musical composition.

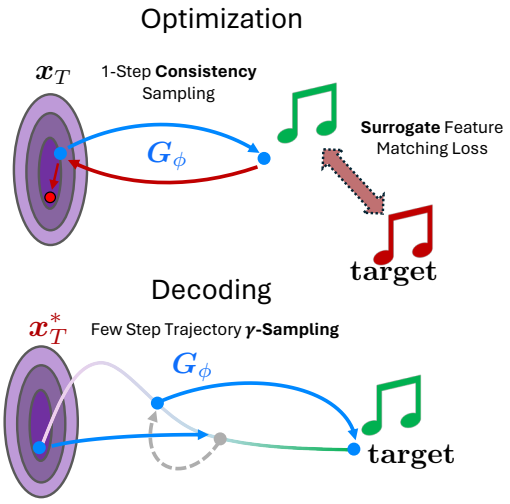


**Figure 1**: DITTO-2: **D**istilled **D**iffusion **I**nference-**T**ime $\boldsymbol{T}$-**O**ptimization. We speed up diffusion inference-time optimization-based music generation by 10-20x while improving control and audio quality. (Top) We use diffusion distillation to speed up performance (optimize with 1-step sampling). (Bottom) We then run multi-step sampling for final higher-quality generation (decoding).

Recent control methods that go beyond text-control fall into training-based and training-free methods. Training-based methods like Music-ControlNet [15] fine-tune DMs with additional adaptor modules that can add time-dependent controls over melody, harmony, and rhythm, offering strong control at the cost of hundreds of GPU hours of fine-tuning for each control. With training-free methods, in particular the class of inference-time *guidance* methods [16,17], the diffusion sampling process is guided at each step using the gradients of a target control $\nabla_{\boldsymbol{x}_t}\mathcal{L}(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t))$, where $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$ is a 1-step approximation of the final output. While training-free, the reliance on approximate gradients limits performance [18]. Finally, inference-time optimization (ITO) methods [19, 20] like DITTO [18] offer state-of-the-art (SOTA) control without the need for large-scale fine-tuning via optimizing for noise latents, but suffer from slow inference speeds (10-20x slower than real-time) [18].

In this work, we propose **D**istilled **D**iffusion **I**nference-**T**ime $\boldsymbol{T}$-**O**ptimization (or DITTO-2), a new method for speeding up ITO-based methods by over an order of magni-

tude for faster-than-real-time generation for a wide-variety of controllable generation tasks including inpainting, outpainting, intensity, melody, and musical structure control. Our method works via 1) distilling a pre-trained diffusion model for fast sampling via an efficient, modified consistency model (CM) [21] or consistency trajectory model (CTM) [22] distillation process (only 32 GPU hours on a 40GB A100), (2) performing inference-time optimization using our distilled model with a 1-step *surrogate* objective, and (3) running a final multi-step sampling generation (decoding) using our estimated noise latents for final best-quality results as shown in Fig. 1. We find our approach accelerates optimization 10-20x, improves control, and improves audio quality at all once. Furthermore, we apply our approach to maximize text adherence (CLAP score) and show how an unconditional diffusion model trained without text inputs can yield SOTA text control.

## 2. BACKGROUND

### 2.1 Diffusion-Based Music Generation

Audio-domain music generation has become tractable through diffusion-based methods, popularized with models such as Riffusion [3], MusicLDM [23], and Stable Audio [4]. Diffusion Models (DMs) [8, 24] are defined using a closed-form forward process, where input audio is iteratively noised according to a Gaussian Markov Chain. DMs then learn to approximate the score of the probability distribution of the reverse process $\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t)$ using a noise prediction model $\boldsymbol{\epsilon}_\theta$, which progressively denoises a random initial latent $\boldsymbol{x}_T \sim \mathcal{N}(0, I)$ to generate new data $\boldsymbol{x}_0$. For audio-domain DMs, diffusion is performed over spectrograms [15] or on the latent representations of an audio-based VAE [3, 4, 23, 25], with an external vocoder used to translate spectrograms back to the time domain. Though DMs are efficiently trained by a simple MSE score matching objective [8, 26], sampling from DMs typically requires running the denoising process for 100s of iterations (calls to $\boldsymbol{\epsilon}_\theta$), and have slower inference than VAEs or GANs [27].

### 2.2 Fast Diffusion Sampling

Fast diffusion sampling is critical. DDIM [8] or DPM-Solver [28] accelerates DMs to sample in only 10-50 sampling steps. To truly increase speed, however, *distillation* can be used to produce a model that can sample in a *single* step [21, 22, 29, 30]. Two promising DM distillation methods include consistency models (CM) [21] and consistency trajectory models [22]. The goal of CMs is to distill a base DM $\boldsymbol{\epsilon}_\theta$ into a new 1-step network $\boldsymbol{x}_0 = \boldsymbol{G}_\phi(\boldsymbol{x}_t, \boldsymbol{c})$ that satisfies the consistency property $\forall t, t' \in [T, 0], \boldsymbol{G}_\phi(\boldsymbol{x}_t, \boldsymbol{c}) = \boldsymbol{G}_\phi(\boldsymbol{x}_{t'}, \boldsymbol{c})$ or that every point along the diffusion trajectory maps to the same output. Formally, CMs are distilled by enforcing local consistency between the learnable $\boldsymbol{G}_\phi$ and an exponential moving average (EMA) copy $\boldsymbol{G}_{\phi^-}$:

$$\mathbb{E}_{t \sim T\,(\boldsymbol{x}\,\boldsymbol{c}) \sim \mathcal{D}} \| \boldsymbol{G}_\phi(\boldsymbol{x}_t, \boldsymbol{c}) - \boldsymbol{G}_{\phi^-}(\Theta(\boldsymbol{\epsilon}_\theta, \boldsymbol{x}_t, \boldsymbol{c}), \boldsymbol{c}) \|_2^2, \quad (1)$$

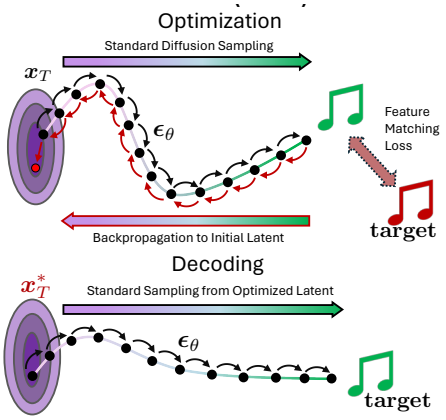


**Figure 2**: (Top) Baseline DITTO runs optimization over a multi-step sampling process to find an initial noise latent to achieve a desired stylized output, incurring a large speed cost. (Bottom) When generating the final output (decoding), the same multi-step diffusion sampling process is used.

where $\Theta(\boldsymbol{\epsilon}_\theta, \boldsymbol{x}_t, \boldsymbol{c})$ denotes one sampling step from $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t-1}$ using the *frozen* teacher model $\boldsymbol{\epsilon}_\theta$ and some sampling algorithm (e.g. DDIM).

CMs are not perfect, however, and one-step performance lags behind DM quality [29]. Multi-step "ping-pong" sampling [21] also does not reliably increase quality due to compound approx. errors in each renoising step. CTMs [22], on the other hand, are designed to fix this problem. CTMs bridge the gap between CMs and DMs by distilling a model $\boldsymbol{x}_s = \boldsymbol{G}_\phi(\boldsymbol{x}_t, \boldsymbol{c}, t, s)$ that can jump from *anywhere* $t$ *to anywhere* $s$ along the diffusion trajectory as shown in Fig. 3. CTMs then use $\gamma$-sampling to interpolate between few-step deterministic sampling along the trajectory ($\gamma = 0$) and CM's "ping-pong" sampling ($\gamma = 1$), allowing a way to balance sampling stochasticity with overall quality. To our knowledge, CTM distillation is unexplored for audio, and CM distillation has only been applied to general audio [31].

### 2.3 Diffusion Inference-time Optimization

Diffusion inference-time optimization (DITTO) [18–20] is a general-purpose framework to control diffusion models at inference-time. The work is based on the observation that the *initial noise latent* $\boldsymbol{x}_T$, traditionally thought of as a random seed, encodes a large proportion of the semantic content in the generation outputs [18, 32]. Thus, we can search for an initial noise latent of the diffusion generation process via optimization to achieve a desired stylized output as shown in Fig. 2. We do this by defining a differentiable feature extraction function (e.g. chroma-based melody extraction) $f(\cdot)$, a matching loss function $\mathcal{L}$ (e.g. cross entropy), a target feature $\boldsymbol{y}$, and optimize $\boldsymbol{x}_T$:

$$\boldsymbol{x}_T^* = \arg\min_{\boldsymbol{x}_T} \mathcal{L}\left(f(\boldsymbol{x}_0), \boldsymbol{y}\right) \quad (2)$$

$$\boldsymbol{x}_0 = \Theta_T(\boldsymbol{\epsilon}_\theta, \boldsymbol{x}_T, \boldsymbol{c}), \quad (3)$$

where $\Theta_T(\boldsymbol{\epsilon}_\theta, \boldsymbol{x}_T, \boldsymbol{c})$ denotes $T$ calls of the model using any sampler $\Theta$. In practice, DITTO is run with a fixed budget of $K$ optimization steps using a standard optimizer (i.e. Adam). This approach allows for any control that can

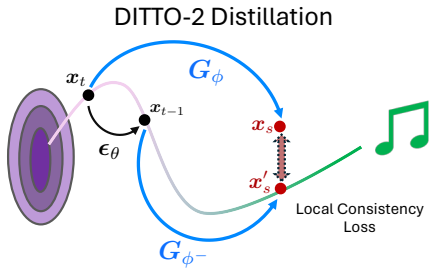


**Figure 3**: CTM Distillation for DITTO-2. We distill $\boldsymbol{G}_\phi$ by minimizing the distance between the jump from $\boldsymbol{x}_t$ to $\boldsymbol{x}_s$ and $\boldsymbol{x}_{t-1}$ to $\boldsymbol{x}_s$, where $\boldsymbol{x}_{t-1}$ is generated by sampling with the base model $\boldsymbol{\epsilon}_\theta$.

be parameterized differentiably, including melody, intensity, and musical structure, as well as editing tasks like inpainting and outpainting. For brevity, we combine Eq. 2 and Eq. 3 into the shorthand $\boldsymbol{x}_T^* = \arg\min_{\boldsymbol{x}_T} \mathcal{L}_\theta^{(T)}(\boldsymbol{x}_T)$.

The downside of DITTO, however, is that it is slow. We need to backpropagate through the *entire* sampling process for each of the $K$ optimization steps and use memory management techniques like gradient checkpointing [33] or invertible networks [19] to handle large memory use that slows down generation. The overall cost of running a single ITO generation is on the order of $4KT$: $T$-step diffusion chain for $K$ opt. steps, with a factor of 2 from gradient management and 2 from using classifier-free guidance (CFG) [34] to improve quality.

## 3. METHOD

### 3.1 Overview

We seek to dramatically speed up the diffusion ITO process to achieve controllable music generation for near-interactive rate music co-creation. To do so, we focus on three critical methodological improvements. First, we leverage **diffusion distillation** to significantly speed up diffusion sampling with an efficient, modified distillation process designed to be used together with ITO methods. Second, we introduce ***surrogate* optimization**, or the idea of decoupling the task of estimating noise latents from the task of rendering a final output or decoding, which allows us to leverage both fast sampling for optimization for control estimation and multi-step sampling for final, high-quality generation. Third, we combine diffusion distillation with surrogate optimization within the DITTO framework and produce a new, more efficient diffusion inference-optimization algorithm (no gradient checkpointing) as found in Section 3.4.

### 3.2 Acceleration through Diffusion Distillation

The clearest way to speed up ITO is to simply reduce the number of diffusion sampling steps $T$. From initial experiments, however, we found that (1) reducing the number of sampling steps $T$ degrades overall generation quality [8], (2) quality degradation makes the optimization gradients weaker (as the outputs are less semantically coherent) leading to control degradation, and (3) achieving close to real-time performance requires $< 4$ sampling steps, which produce fully incoherent results on standard DMs. Thus, we employ distillation to speed up the diffusion process [29].

First, we develop CM distillation [21, 29, 31] for ITO-based controllable music generation. For CM distillation, we follow past work [29] for our training recipe, optimizing (1), and also learn an explicit embedding for the CFG scale $w$ in the model $\boldsymbol{G}_\phi(\boldsymbol{x}_t, \boldsymbol{c}, w)$ during distillation following [29]. By distilling CFG, we are able to half the number of total model calls per distilled diffusion sampling step. Once distilled, $\boldsymbol{G}_\phi$ jumps from $\boldsymbol{x}_T$ to $\boldsymbol{x}_0$, allowing for deterministic 1-step sampling and stochastic multi-step-sampling by repeatedly *renoising* with some $\epsilon \sim N(0, I)$ back to $\boldsymbol{x}_{t-1}$.

Second, we develop CTM distillation [22] for ITO-based controllable music generation. CTM distillation offers more advantageous speed vs. quality design trade-offs, but comes at a cost of a more complex training procedure. In more detail, CTM distillation normally involves an expensive soft-consistency loss in the data domain with added GAN and score-matching loss terms. As we aim to distill our base model for *surrogate* optimization (see Sec. 3.3), we are able to simplify and speed up the CTM distillation process. First, we remove the image-domain GAN loss to reduce complexity of developing an audio-based GAN loss. Second, we use the consistency term from CTM in local-consistency form [35]:

$$\mathbb{E}_{t\ s\sim T\ (\boldsymbol{x}\ \boldsymbol{c})\sim\mathcal{D}} \| \boldsymbol{G}_\phi(\boldsymbol{x}_t, \boldsymbol{c}, w, t, s) - \boldsymbol{G}_{\phi^-}(\Theta(\boldsymbol{\epsilon}_\theta, \boldsymbol{x}_t, \boldsymbol{c}), \boldsymbol{c}, w, t-1, s) \|_2^2. \quad (4)$$

Third, we use the 1-step Euler parameterization of $\boldsymbol{G}_\phi$ from [22]'s Appendix, which avoids explicitly learning additional parameters for the target step $s$ in order to accelerate distillation. These changes reduce the number of per-training step model calls from 10-30 to 3, leading to a near order-of-magnitude speed up in wall clock time for performing the distillation process. Finally, we upgrade CTM's unconditional framing for conditional diffusion by incorporating $\boldsymbol{c}$ into the distillation procedure, and adding $w$ directly into the model to distill the CFG weight into an explicit parameter following past work [29], resulting in CFG control at inference but without double the complexity. In total, we perform distillation in as few as 32 GPU hours on an A100, the fastest trajectory-based distillation to our knowledge [22, 35].

### 3.3 Surrogate Optimization

Given a distilled CM or CTM model, we seek to minimize our inference runtime and maximize control adherence and audio quality. The obvious choice to minimize runtime is to use our distilled model with one-step sampling, but this results in limited audio quality and text-control. To solve this, we first split the ITO process into two separate phases: **optimization**, i.e. the nested loop of optimizing the initial latent over $M$-step multi-step sampling, and **decoding**, i.e. the final $T$-step sampling process from the optimized latent $\boldsymbol{x}_T^*$, where $M = T$ in all past work [18, 19]. In this light, it is clear that the optimization phase is mostly respon-

**Algorithm 1** **D**istilled **D**iffusion **I**nference-**T**ime ***T***-**O**ptimization (DITTO-2)

**input** : $\boldsymbol{G}_\phi$, feature extractor $f$, loss $\mathcal{L}$, target $\boldsymbol{y}$, starting latent $\boldsymbol{x}_T$, text $\boldsymbol{c}$, optimization steps $K$, optimizer $g$, decoding steps $\{\tau_0, \ldots, \tau_M\}$, $\gamma$, CFG weight $w$

// Optimization Loop
**for** $K$ iterations **do**
  $\boldsymbol{x}_0 = \boldsymbol{G}_\phi(\boldsymbol{x}_T, \boldsymbol{c}, w, T, 0)$
  $\hat{\boldsymbol{y}} = f(\boldsymbol{x}_0)$
  $\boldsymbol{x}_T \leftarrow \boldsymbol{x}_T - g(\nabla_{\boldsymbol{x}_T}\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}))$
**end for**
// Decoding Loop
$\boldsymbol{x}_t \leftarrow \boldsymbol{x}_T$
**for** $t = M$ to 1 **do**
  $\hat{\tau}_{t-1} = \sqrt{1-\gamma^2}\tau_{t-1}$
  $\boldsymbol{x}_{t-1} = \boldsymbol{G}_\phi(\boldsymbol{x}_{t-1}, \boldsymbol{c}, w, \tau_t, \hat{\tau}_{t-1}) + \gamma\tau_{t-1}\epsilon$
**end for**

**output** : $\boldsymbol{x}_0$

sible for the control strength and runtime, while decoding is generally responsible for final output quality.

Thus, we fix our final decoding process as multi-step sampling with $T$ steps. Then, we perform control optimization over a **surrogate** objective $\hat{\boldsymbol{x}}_T^* = \arg\min_{\boldsymbol{x}_T} \mathcal{L}_\phi^{(M)}(\boldsymbol{x}_T)$ using some model $\boldsymbol{\epsilon}_\phi$ and $M \ll T$, where our surrogate is more efficient but yields approx. equal latents to our original objective

$$\arg\min_{\boldsymbol{x}_T} \mathcal{L}_\phi^{(M)}(\boldsymbol{x}_T) \approx \arg\min_{\boldsymbol{x}_T} \mathcal{L}_\theta^{(T)}(\boldsymbol{x}_T). \quad (5)$$

A natural candidate for a surrogate model would be the base DM $\boldsymbol{\epsilon}_\theta$ with fewer sampling steps. DM performance, however, becomes fully incoherent as $M \to 1$ [21, 22], causing a significant domain-gap when $M < T$. Alternatively, our distilled models are naturally strong surrogates:

- One-step outputs are generally coherent unlike in base DMs, resulting in more stable gradients when $M = 1$.
- Since distilled models excel at few-step sampling (i.e. $< 8$) [21,22], the control domain gap between $M$ and $T$ can be reduced while ensuring coherent outputs.
- CTMs can increase quality with more sampling.

As a result, we use a CM or CTM $\boldsymbol{G}_\phi$ as our surrogate, optimize with $M = 1$, and decode with $T \in [1, 8]$.

### 3.4 Complete Algorithm

Given our efficient CTM-based distillation process, and our surrogate objective, we propose a new ITO algorithm for controllable music generation in Alg. 1. Here, we run optimization to estimate control parameters using our surrogate 1-step objective. Then, we use the optimized latent $x_T^*$ and decode from our surrogate model with $T$ steps using either multi-step CM Sampling (i.e. $\gamma = 1$) or CTM $\gamma$-sampling ($\gamma < 1$). Beyond decoupling optimization and decoding, we also eliminate the need for gradient checkpointing found in the original DITTO method [18]. In total, we reduce the ITO speed from $4KT$ costly operations for DITTO to $K + T$.

## 4. EXPERIMENTS

To evaluate our proposed method, we follow the evaluation protocol used for DITTO [18] for intensity, melody, music structure, inpainting, and outpainting as described below. Before the full breadth of application tests, however, we explore our design space by comparing different distillation techniques and surrogate options on the task of intensity control. We further conclude with an experiment showing an adaptive sampling surrogate scheme as well a new experiment on maximizing text-adherence (CLAP score).

### 4.1 Controllable Generation Evaluation Protocol

We benchmark our method on five controllable music generation tasks from DITTO [18] including:

- **Intensity Control** [15, 18]: Here, we control the time-varying volume and overall semantic density to some target intensity curve $\boldsymbol{y}$ using the extractor $f(\boldsymbol{x}_0) \coloneqq \boldsymbol{w} * 20\log_{10}(\texttt{RMS}(\mathbf{V}(\boldsymbol{x}_0)))$ (i.e. the RMS energy of the vocoder $\mathbf{V}$ outputs smoothed with a Savitsky-Golay filter $\boldsymbol{w}$) and $\mathcal{L} = ||f(\boldsymbol{x}_0) - \boldsymbol{y}||_2^2$.
- **Melody Control** [2, 15, 18]: We control the model outputs to match a given target melody $\boldsymbol{y} \in \{1, \ldots, 12\}^{N\times 1}$ (where $N$ is number of frames) using the chromagram of the model outputs $f(\boldsymbol{x}_0) = \log(\mathbf{C}(\mathbf{V}(\boldsymbol{x}_0)))$ and $\mathcal{L} = \text{NLLLoss}(f(\boldsymbol{x}_0), \boldsymbol{y})$.
- **Musical Structure Control** [18]: We control the overall timbral structure of the model outputs by regressing the self-similarity (SS) matrix $f(\boldsymbol{x}_0) = \mathbf{T}(\boldsymbol{x}_0)\mathbf{T}(\boldsymbol{x}_0)^\top$ of Mel Frequency Cepstrum Coefficients (MFCC) $\mathbf{T}$ against a target SS matrix $\boldsymbol{y}$ like "ABA" form with $\mathcal{L} = ||f(\boldsymbol{x}_0) - \boldsymbol{y}||_2^2$.
- **Inpainting and Outpainting** [16, 18]: Given music $\boldsymbol{x}_{\text{ref}}$, we can continue (outpainting) or infill (inpainting) $\boldsymbol{x}_{\text{ref}}$ by matching the model outputs over $o$-length overlap regions $f(\boldsymbol{x}_0) \coloneqq \mathbf{M}_{\text{gen}} \odot \boldsymbol{x}_0$ to the reference $\boldsymbol{y} = \mathbf{M}_{\text{ref}} \odot \boldsymbol{x}_{\text{ref}}$ (where $\mathbf{M}_{\text{ref}}$ and $\mathbf{M}_{\text{gen}}$ denote the overlap masks) and $\mathcal{L} = ||f(\boldsymbol{x}_0) - \boldsymbol{y}||_2^2$.

For brevity, we focus on the $o = 1$ case for outpainting and inpainting (i.e. a gap of 4 seconds) and omit looping given its equivalence. See [18] for a more thorough description.

### 4.2 Pre-training and Distillation Details

For our base DM, we follow a similar setup and model design to DITTO [18], using the same base model and vocoder. Specifically, we train a 41M parameter Stable Diffusion-style 2D UNet directly over 6-second mel-spectrograms trained on $\approx$ 1800 hours of licensed music, and MusicHiFi [36] as the vocoder. The base model is trained with genre, mood, and tempo tags similar to [37] rather

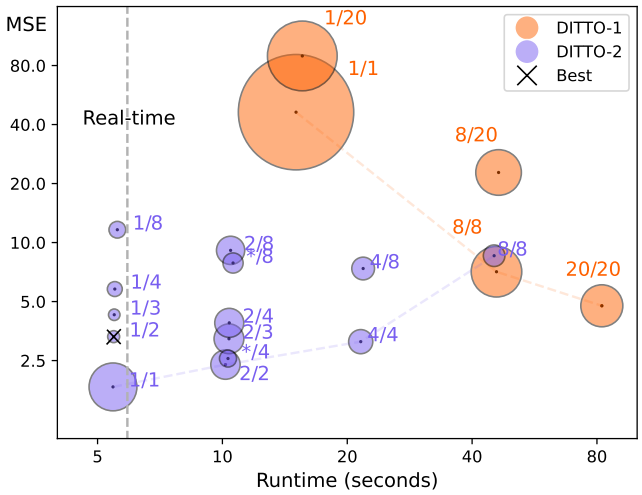


**Figure 4**: **DITTO-2 inference speed vs. control MSE vs. audio quality (FAD, denoted by size, smaller is better)**. Dashed line denotes the cutoff for real-time performance, color denotes ITO method, and subscripts denote number of sampling steps during optimization / final decoding. Applied to intensity control. Trends also hold for CLAP score.

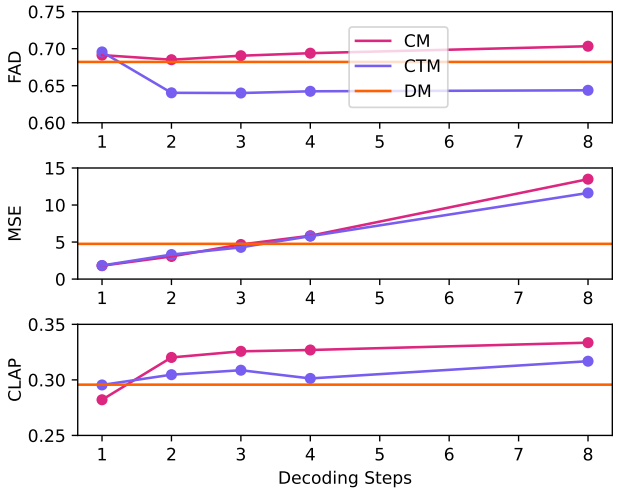


**Figure 5**: FAD, MSE, and CLAP results on Intensity Control for 1-step optimization, where orange lines denote baseline 20-step performance. MSE increases with more decoding steps for both CM/CTM given the domain gap though beats the baseline with $< 4$ steps. CM is unable to beat baseline quality due to accumulated errors in multi-step sampling, while multi-step CTM achieves SOTA quality.

than full text descriptions. Both the CM and CTM surrogate models are distilled using a maximum of $T = 20$ sampling steps, evenly spaced across the trajectory for 4 hours across 8 A100 40GB GPUs on the same data. For DITTO-2, we use Adam. During CTM $\gamma$-sampling, we set $\gamma \in [0.05, 0.35]$ as empirically we found that using deterministic $\gamma = 0$ resulted in noticeable audio artifacts that degrade overall quality.

### 4.3 Metrics

For all tasks, we report the Frechet Audio Distance (FAD) and CLAP Score with the CLAP [38] music backbone (as for FAD the standard VGGish backbone poorly correlates with human perception [39]), which measure overall audio quality and text relevance respectively across 2.5K generations. FAD is calculated with MusicCaps as the reference [1] dataset. Since our base model uses tags rather than captions, we convert each tag set into captions for CLAP Score calculation using the format *"A [mood] [genre] song at [tempo] beats per minute."* Additionally, we report the MSE to the control target for intensity and structure control, and the overall accuracy for melody control.

## 5. RESULTS

### 5.1 Design Exploration Results

We study our design space for ITO via a case study on the task of intensity control. Notably, we compare DITTO with our proposed approach using CM and CTM distilled models in Fig. 4. We show runtime in seconds (x-axis), control MSE (y-axis), and FAD (point size) for an array of $(M, T)$ combinations for our base DM, as well as our distilled CTM models (i.e. DM-8/20 corresponds to the base DM with $M = 8, T = 20$). We find that our distilled models are over 10x faster than the standard DITTO (20, 20) configuration, while simultaneously achieving both better audio quality and control adherence. To understand these trends more in depth, specifically as we vary the number of decoding steps for our distilled models, we show both FAD (top), MSE (middle), and CLAP (bottom) in Fig. 5 as a function on number of decoding steps with $M = 1$, where the orange line denotes the baseline DITTO results with $M = T = 20$.

A few key points of DITTO-2 are visible here. Notably, both CM and CTM distilled models are able to achieve *better* control adherence than the base performance, as the shorter optimization process allows convergence to happen more effectively. Additionally, we find that CTM is clearly stronger than CM in terms of quality, as CTM is able to cleanly trade-off quality for control adherence (as sampling with more steps with $M = 1$ introduces a domain gap) and even *improve* baseline quality, while CM exhibits no real quality trend when sampling more due to its accumulated errors in multi-step sampling. In particular, CTM with $M = 1, T = 2$ achieves SOTA control adherence and FAD with faster than real-time. CM and CTM multi-step sampling also improves text relevance above the base DM.

### 5.2 Benchmark Results

We show full results on our suite of controllable music generation benchmarking results in Table 1. Here, we compare baseline DITTO with DITTO-2, where we display results for the best performing $(M, T)$ setup in each experiment for CM and CTM, where best performance was chosen by finding the setup with the lowest latency (and thus best control adherence) while roughly matching the best overall FAD. As a whole, DITTO-2 achieves comparable or better performance than DITTO on all tasks with an 10-20x speedup, clearing the way for near real-time inference-time

| **Intensity** | Time (s) | FAD | CLAP | MSE |
|---|---|---|---|---|
| DITTO | 82.192 | 0.682 | 0.296 | 4.758 |
| DITTO-2 (CM) | **5.206** | 0.685 | **0.320** | **3.055** |
| DITTO-2 (CTM) | 5.467 | **0.640** | 0.309 | 3.311 |
| **Melody** | Time (s) | FAD | CLAP | Acc. |
| DITTO | 230.780 | 0.699 | 0.283 | 82.625 |
| DITTO-2 (CM) | **21.867** | **0.697** | **0.303** | 81.577 |
| DITTO-2 (CTM) | 22.501 | 0.698 | 0.273 | **85.226** |
| **Musical Structure** | Time (s) | FAD | CLAP | MSE |
| DITTO | 245.295 | **0.632** | **0.281** | 0.024 |
| DITTO-2 (CM) | **11.381** | 0.669 | 0.234 | **0.020** |
| DITTO-2 (CTM) | 11.749 | 0.658 | 0.226 | 0.022 |
| **Outpainting** | Time (s) | FAD | CLAP | |
| DITTO | 144.437 | 0.716 | 0.343 | |
| DITTO-2 (CM) | **6.658** | 0.694 | 0.319 | |
| DITTO-2 (CTM) | 7.098 | **0.680** | **0.347** | |
| **Inpainting** | Time (s) | FAD | CLAP | |
| DITTO | 145.486 | 0.690 | 0.339 | |
| DITTO-2 (CM) | **6.744** | 0.689 | **0.358** | |
| DITTO-2 (CTM) | 6.814 | **0.660** | 0.337 | |

**Table 1**: Controllable generation benchmark results. Best performing configuration for each DITTO-2 setup across five unique tasks. Both CM and CTM results yield excellent results with 10-20x speed ups.

| $M$ | $T$ | Runtime | FAD | CLAP | MSE |
|---|---|---|---|---|---|
| 1 | 1 | 5.447 | 0.696 | 0.295 | 1.835 |
| 1 | 2 | 5.467 | 0.640 | 0.307 | 3.311 |
| 1 | 4 | 5.502 | 0.643 | 0.301 | 5.792 |
| 2 | 2 | 10.171 | 0.659 | 0.281 | 2.384 |
| 2 | 4 | 10.387 | 0.658 | 0.296 | 3.894 |
| Adaptive | 4 | 10.315 | 0.644 | 0.296 | 2.561 |

**Table 2**: Intensity control results with various $(M, T)$ options including an adaptive sampling during optimization.

controllable music generation. Specifically, we find that CTM outperforms CM, showing noticeably better quality with similar runtime and control adherence.

### 5.3 Variable Compute Budget Optimization

Though we are primarily interested in real-time performance (i.e. as fast as possible), we additionally investigated how we can use a varying compute budget during optimization (in terms of runtime). As simply increasing $M$ predictably increases runtime by a multiplicative factor, we designed an *adaptive* schedule for $M$ (denoted as $*$ in Fig. 4) in order to improve downstream decoding performance without increasing runtime significantly. Formally, for $K$ optimization steps, we set the adaptive budget as using $M = 1$ for $\lfloor \frac{K}{2} \rfloor$ iterations, then $M = 2$ for $\lfloor \frac{3K}{8} \rfloor$, and finally $M = 4$ for $\lfloor \frac{K}{8} \rfloor$ iterations, thus allowing a coarse-to-fine optimization process. In Table 2, using the adaptive schedule exhibits the runtime of the $M = 2$ case yet achieves much better FAD and similar control adherence. This shows that given a more flexible compute budget, using an adaptive $M$ schedule balances downstream performance better than simply modifying a fixed $M$, and allows smoother objective tradeoff between audio quality and control strength.

| Method | Condition | FAD | CLAP |
|---|---|---|---|
| Base TTM | Tags | 0.488 | 0.167 |
| DITTO-2 | Tags | 0.456 | 0.317 |
| DITTO-2 | N/A | 0.440 | 0.341 |
| U-DITTO-2 | N/A | **0.430** | **0.347** |
| MusicGen (1.5B) | Caption | 0.444 | 0.237 |
| MusicGen (3.3B) | Caption | 0.437 | 0.226 |

**Table 3**: Text similarity results. We use DITTO-2 to maximize CLAP similarity using a fully unconditional pretrained diffusion model and yield a 54% relative improvement over past SOTA CLAP score (MusicGen).

### 5.4 Inference-time Optimization of Text-Control

Past ITO methods for music generation use simple feature extractors $f(\cdot)$ (i.e. chroma or RMS energy) [18] to minimize runtime speed. Given that our method is much faster, however, we can introduce new bespoke control applications with neural network-based feature extractors. Thus, we propose the task of inference-time **text similarity** control. We extract the normalized CLAP audio embedding [38] of our model outputs $f(\cdot) = \text{CLAP}(\boldsymbol{x}_0)$ and, given some natural language caption $\boldsymbol{y}$, calculate the cosine distance between the output and the normalized CLAP text embedding of the caption $\mathcal{L}(\boldsymbol{x}_0) = 1 - f(\boldsymbol{x}_0)^\top f(\boldsymbol{y})$.

Using FAD and CLAP score as metrics, we benchmark several configurations including our base DM model with tag inputs, DITTO-2 method with tag inputs, DITTO-2 method with null tag inputs, MusicGen w/melody (1.5B) [2], and MusicGen w.o./melody (3.5B) [2]. For models that take input text, we use captions from MusicCaps [1] as input and for models with tag inputs, we convert MusicCaps captions to tags via GPT-4 as done in past work [15] with tempo extracted from audio. Furthermore, to ablate whether any part of the tag-conditioned training process influences downstream DITTO-2 CLAP control, we retrain and distill our base model *without any text input*, which we denoted U-DITTO-2. In Table 3, we see that DITTO-2 enables SOTA text relevance compared to MusicGen by an over 54% relative improvement (large), thus showing the benefits of ITO-based approaches which allow us to directly optimize for desired downstream metrics, and notably enables fully-unconditional models to have text control *with no paired music-text training*.

## 6. CONCLUSION

We present DITTO-2: **D**istilled **D**iffusion **I**nference-**T**ime ***T*-O**ptimization, a new efficient method for accelerating inference-time optimization for fast controllable music generation. By utilizing a modified consistency or consistency trajectory distillation process and performing inference-time optimization on efficient surrogate objectives, we speed up past ITO methods by over 10-20x while simultaneously improving audio quality and text control. Furthermore, we find we can leverage the efficiency of our method on new, more complex tasks like text-adherence and show we can convert a fully unconditional diffusion model into a TTM model that yields SOTA results on evaluated metrics.

## 7. REFERENCES


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

[2] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," in *Neural Information Processing Systems (NeurIPS)*, 2023.

[3] S. Forsgren and H. Martiros, "Riffusion: Stable diffusion for real-time music generation," 2022. [Online]. Available: https://riffusion.com/about

[4] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, "Fast timing-conditioned latent audio diffusion," *International Conference on Machine Learning (ICML)*, 2024.

[5] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "AudioLDM: Text-to-audio generation with latent diffusion models," in *International Conference on Machine Learning (ICML)*, 2023.

[6] H. Liu, Y. Yuan, X. Liu, X. Mei, Q. Kong, Q. Tian, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, "Audioldm 2: Learning holistic audio generation with self-supervised pretraining," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[7] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Neural Information Processing Systems (NeurIPS)*, 2020.

[8] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations (ICLR)*, 2020.

[9] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[10] K. Kumar, R. Kumar, T. De Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. De Brebisson, Y. Bengio, and A. C. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," *Neural Information Processing Systems (NeurIPS)*, 2019.

[11] 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 (TASLP)*, 2021.

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

[13] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved RVQGAN," in *Neural Information Processing Systems (NeurIPS)*, 2023.

[14] H. Manor and T. Michaeli, "Zero-shot unsupervised and text-based audio editing using ddpm inversion," *International Conference on Machine Learning (ICML)*, 2024.

[15] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, "Music ControlNet: Multiple time-varying controls for music generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 2024.

[16] M. Levy, B. D. Giorgi, F. Weers, A. Katharopoulos, and T. Nickson, "Controllable music production with diffusion models and guidance gradients," in *Diffusion Models Workshop at NeurIPS*, 2023.

[17] J. Yu, Y. Wang, C. Zhao, B. Ghanem, and J. Zhang, "FreeDoM: Training-free energy-guided conditional diffusion model," *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[18] Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, "DITTO: Diffusion inference-time T-optimization for music generation," in *International Conference on Machine Learning (ICML)*, 2024.

[19] B. Wallace, A. Gokul, S. Ermon, and N. V. Naik, "End-to-end diffusion latent optimization improves classifier guidance," *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[20] K. Karunratanakul, K. Preechakul, E. Aksan, T. Beeler, S. Suwajanakorn, and S. Tang, "Optimizing diffusion noise can serve as universal motion priors," *arXiv preprint arXiv:2312.11994*, 2023.

[21] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," in *International Conference on Machine Learning (ICML)*, 2023.

[22] D. Kim, C.-H. Lai, W.-H. Liao, N. Murata, Y. Takida, T. Uesaka, Y. He, Y. Mitsufuji, and S. Ermon, "Consistency trajectory models: Learning probability flow ODE trajectory of diffusion," in *International Conference on Learning Representations (ICLR)*, 2023.

[23] K. Chen, Y. Wu, H. Liu, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, "MusicLDM: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies," in *IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2024.

[24] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," *Neural Information Processing Systems (NeurIPS)*, 2021.

[25] M. W. Y. Lam, Q. Tian, T.-C. Li, Z. Yin, S. Feng, M. Tu, Y. Ji, R. Xia, M. Ma, X. Song, J. Chen, Y. Wang, and Y. Wang, "Efficient neural music generation," in *Neural Information Processing Systems (NeurIPS)*, 2023.

[26] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[27] M. Pasini and J. Schlüter, "Musika! fast infinite waveform music generation," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

[28] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models," *ArXiv*, vol. abs/2211.01095, 2022.

[29] S. Luo, Y. Tan, L. Huang, J. Li, and H. Zhao, "Latent consistency models: Synthesizing high-resolution images with few-step inference," *arXiv preprint arXiv:2310.04378*, 2023.

[30] A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach, "Adversarial diffusion distillation," *ArXiv*, vol. abs/2311.17042, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:265466173

[31] Y. Bai, T. Dang, D. Tran, K. Koishida, and S. Sojoudi, "Accelerating diffusion-based text-to-audio generation with consistency distillation," in *Interspeech*, 2024.

[32] C. Si, Z. Huang, Y. Jiang, and Z. Liu, "FreeU: Free lunch in diffusion U-Net," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[33] T. Chen, B. Xu, C. Zhang, and C. Guestrin, "Training deep nets with sublinear memory cost," *arXiv preprint arXiv:1604.06174*, 2016.

[34] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *NeurIPS Workshop on Deep Gen. Models and Downstream Applications*, 2021.

[35] J. Zheng, M. Hu, Z. Fan, C. Wang, C. Ding, D. Tao, and T.-J. Cham, "Trajectory consistency distillation: Improved latent consistency distillation by semi-linear consistency function with trajectory mapping," 2024. [Online]. Available: https://arxiv.org/abs/2402.19159

[36] G. Zhu, J.-P. Caceres, Z. Duan, and N. J. Bryan, "MusicHiFi: Fast high-fidelity stereo vocoding," *IEEE Signal Processing Letters (SPL)*, 2024.

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

[38] 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 *IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2023.

[39] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, "Adapting Frechet Audio Distance for generative music evaluation," in *IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2024.