arXiv:2506.00736v1 [eess.AS] 31 May 2025

IMPACT: Iterative Mask-based Parallel Decoding for Text-to-Audio Generation with Diffusion Modeling

Kuan-Po Huang [1 2 †] **Shu-wen Yang** [1 2 †] **Huy Phan** [2] **Bo-Ru Lu** [2] **Byeonggeun Kim** [2] **Sashank Macha** [2] **Qingming Tang** [2] **Shalini Ghosh** [2] **Hung-yi Lee** [1] **Chieh-Chi Kao** [2] **Chao Wang** [2]

Abstract

Text-to-audio generation synthesizes realistic sounds or music given a natural language prompt. Diffusion-based frameworks, including the Tango and the AudioLDM series, represent the state-of-the-art in text-to-audio generation. Despite achieving high audio fidelity, they incur significant inference latency due to the slow diffusion sampling process. MAGNET, a mask-based model operating on discrete tokens, addresses slow inference through iterative mask-based parallel decoding. However, its audio quality still lags behind that of diffusion-based models. In this work, we introduce IMPACT, a text-to-audio generation framework that achieves high performance in audio quality and fidelity while ensuring fast inference. IMPACT utilizes iterative mask-based parallel decoding in a continuous latent space powered by diffusion modeling. This approach eliminates the fidelity constraints of discrete tokens while maintaining competitive inference speed. Results on AudioCaps demonstrate that IMPACT achieves state-of-the-art performance on key metrics including Fréchet Distance (FD) and Fréchet Audio Distance (FAD) while significantly reducing latency compared to prior models. The project website is available at https://audio-impact.github.io/.

1. Introduction

The text-to-audio generation task aims to synthesize high-quality and high-fidelity audio that aligns semantically with a given textual prompt. This task holds immense potential for applications ranging from audio content creation and video gaming to marketing and advertising. The current state-of-the-art in text-to-audio generation is represented by the Tango (Ghosal et al., 2023; Kong et al., 2024; Majumder et al., 2024) and AudioLDM (Liu et al., 2023; 2024) series, which leverage diffusion-based models to achieve high-quality audio synthesis. All these models employ computationally heavy network architectures with attention layers as the backbone for their diffusion models. However, this design results in high latency due to slow inference speed, as the iterative denoising steps of the diffusion sampling process combined with the model's complexity significantly increase the time required for generating outputs.

To address the issue of slow inference speed, MAGNET (Ziv et al., 2024), a masked-based generative model (MGM), utilizes iterative mask-based parallel decoding to achieve efficient audio generation. During inference, the model progressively predicts and refines discrete audio tokens across multiple decoding iterations, leveraging parallelism to predict multiple tokens simultaneously at each step. This parallel decoding strategy not only delivers significantly faster inference compared to traditional autoregressive models like MusicGen (Copet et al., 2024) and AudioGen (Kreuk et al., 2023), but also surpasses the inference speed of diffusion-based models by eliminating the need for time-consuming diffusion sampling.

While MAGNET leverages discrete tokens for efficient and structured audio generation, its performance on text-to-audio generation tasks remains inferior to current state-of-the-art models. Given the observed superiority of continuous representations over discrete tokens in tasks such as text-to-image generation (Fan et al., 2024), speech large language models (Yuan et al., 2024), and automatic speech recognition (Xu et al., 2024), one intuitive way to enhance the generation performance of MAGNET is to replace its discrete tokens with continuous representations. However, based on our preliminary experiments, this intuitive modification resulted in significantly worse performance compared to the original MAGNET model.

Knowing that latent diffusion models (LDMs) are good at modeling continuous representations (Ghosal et al., 2023; Huang et al., 2023; Kong et al., 2024; Majumder et al., 2024;

[†]Work done during internship at Amazon in 2024. [1]National Taiwan University, Taipei, Taiwan [2]Amazon AGI, United States. Correspondence to: Kuan-Po Huang <gerber861017@gmail.com>, Chieh-Chi Kao <chiehchi@amazon.com>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

Liu et al., 2023; 2024; Hai et al., 2024), we propose to integrate iterative mask-based parallel decoding with LDMs to better model continuous representations for the text-to-audio generation task. LDMs require a multi-step diffusion sampling process, which inherently imposes high computational costs and slows inference if used independently. However, by integrating iterative mask-based parallel decoding, we can replace the heavy attention-based layers typically used in LDMs with a lightweight MLP-based diffusion head, substantially reducing sampling time while maintaining audio quality and fidelity. In addition, we introduce an unconditional pre-training phase before text-conditional training on paired text-audio data, a step shown to be indispensable for this task. Our experimental results confirm that this design enables low-latency inference while preserving high audio fidelity, quality, and text relevance.

In summary, we state our contributions as follows:

- We pioneer the use of iterative mask-based parallel decoding on continuous latent representations, powered by LDMs, for text-to-audio generation.
- We propose an unconditional pre-training phase preceding conditional training during the MGM training process and demonstrate its effectiveness.
- Our model achieves state-of-the-art performance on objective metrics FD and FAD, and subjective evaluations REL and OVL, while remaining competitive with the fastest existing text-to-audio generation model, MAGNET-S, in terms of inference speed.

2. Related Work

2.1. Mask-based Generative Models (MGM)

Mask-based generative modeling (MGM) has emerged as a powerful technique in discrete-token-based sequence modeling to deal with tasks such as audio (Soundstorm (Borsos et al., 2024), MAGNET (Ziv et al., 2024)), music (VampNet (Garcia et al., 2023), MAGNET (Ziv et al., 2024)), and image generation (MaskGIT (Chang et al., 2022), MUSE (Chang et al., 2023), MAGE (Li et al., 2023)). This approach, which involves masking portions of the input token sequence and training a model to reconstruct the missing information, offers advantages in terms of efficiency and parallelization by employing iterative mask-based parallel decoding. During inference, unlike traditional autoregressive models, such as AudioGen (Kreuk et al., 2023), which generate tokens one at a time, the iterative mask-based parallel decoding process starts with a fully empty sequence and unravels a set of tokens at each decoding iteration to progressively build up a sequence of tokens. The rationale behind this approach is that the generation process starts without prior content. In early iterations, the model has limited context to inform token predictions. As decoding progresses and more tokens are generated, the model gains additional context, enhancing its predictive capabilities in subsequent iterations.

2.2. Latent Diffusion Models for Generation

Latent Diffusion Models (LDMs) are widely employed in text-to-audio tasks due to their ability to operate within a continuous latent space (Liu et al., 2023; 2024; Ghosal et al., 2023; Kong et al., 2024; Majumder et al., 2024; Hai et al., 2024). By encoding audio signals with variational autoencoders (VAEs) (Liu et al., 2023), LDMs surpass discrete-token-based approaches such as MAGNET (Ziv et al., 2024) in both quality and fidelity. However, due to the iterative nature of diffusion sampling, large LDMs can incur substantial inference overhead.

In the field of computer vision, MAR (Li et al., 2024), an MGM-based model, achieved state-of-the-art performance in class-conditional image generation. Unlike earlier MGM-based models such as MaskGIT (Chang et al., 2022), MAGE (Li et al., 2023) and MUSE (Chang et al., 2023) that predict discrete tokens, MAR introduces iterative mask-based parallel decoding directly on continuous representations using LDMs. This choice not only improves image quality and fidelity but also speeds up the diffusion sampling process by replacing heavy attention layers with a lightweight MLP diffusion head.

Inspired by MAR, we present a text-to-audio generation approach that likewise leverages iterative mask-based parallel decoding over continuous latent representations driven by LDMs. We further introduce an unconditional pre-training phase before text-conditional training on paired text-audio data, which is shown to be critical for this task. The results demonstrate that our approach achieves low-latency inference while preserving high audio quality, fidelity, and text relevancy.

3. IMPACT

IMPACT employs the MGM approach with a conditional LDM. During training, the method first undergoes unconditional pre-training, where no text is used. This phase teaches the model to reconstruct audio latents from partially masked inputs, leveraging large unlabeled datasets to learn the basics of audio generation, which is essential for this task. Next, the model is trained with text conditions by concatenating audio latents with a text condition vector sequence and further encoded by a Transformer-based latent encoder. Here, a small diffusion head predicts the noise used to corrupt masked audio latents, thereby learning to generate audio consistent with text prompts. During inference, the method applies iterative mask-based parallel decoding, start-

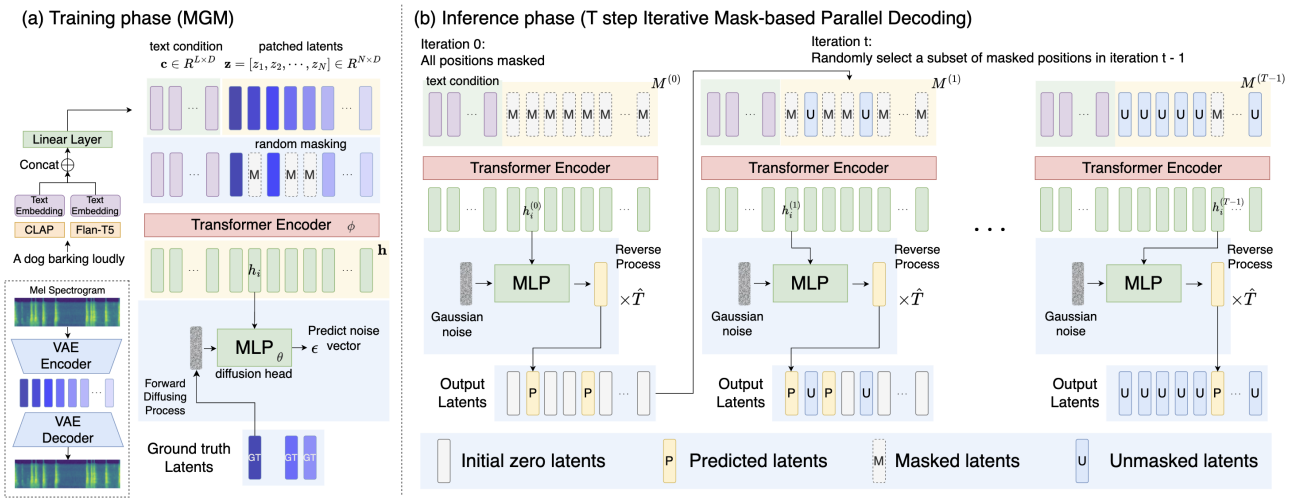

Figure 1. Diagram of our IMPACT framework for the training phase and inference phase.

ing with a fully empty sequence and gradually generating latents at each iteration with the small diffusion head, saving much time since the diffusion sampling loop operates on a lightweight model.

3.1. Training Phase

3.1.1. Text-conditional Training with MGM

Figure 1(a) illustrates the MGM training procedure combined with LDMs. Given an audio input, we extract its representation using an audio VAE and arrange it into a sequence of audio latents $\mathbf{z} = [z_1, z_2, \cdots, z_N]$. To train with MGM, a number of $q \cdot N$ latents are masked by a binary mask $M \in \{0,1\}^N$, where q is the masking percentage factor and $M[i] = 1$ indicates that the i-th audio latent is masked. The text condition vector sequence $\mathbf{c}$ is concatenated with the remaining unmasked latents and forwarded to a Transformer-based latent encoder Enc_ϕ to produce the hidden representation $\mathbf{h}$, as described by Eq. (1):

$$\mathbf{h} = \text{Enc}_\phi(\text{concat}(\mathbf{c}, \mathbf{z} \odot \bar{M})), \mathbf{h} \in \mathbb{R}^{(L+N)\times D} \quad (1)$$

where $\bar{M}$ is the complement of M, L denotes the sequence length of $\mathbf{c}$, and D is the encoder embedding dimension. The goal of our framework is to have a diffusion head to predict these masked audio latents based on input $\mathbf{h}$. To train such a framework, the latents that were masked are corrupted with noise forming $z_i^{\hat{t}} = \sqrt{\bar{\alpha}_{\hat{t}}} z_i + \sqrt{1 - \bar{\alpha}_{\hat{t}}}\epsilon$ based on the closed-form solution of the forward diffusing process (Ho et al., 2020; Nichol & Dhariwal, 2021), where $\epsilon \in \mathbb{R}^D$ denotes a noise vector sampled from the multivariate normal distribution $\mathcal{N}(0, \mathbf{I})$, $\bar{\alpha}_{\hat{t}}$ is the noise schedule, and $\hat{t}$ represents the time step of the noise schedule, also known as the diffusion step. A diffusion head ϵ_θ, parameterized by θ, takes h_i as a condition to predict the noise ϵ used to corrupt the latents. Eq. (2) shows the training objective,

$$\arg\min_{\{\phi,\theta\}} \sum_{\{i|M[i]=1\}} \left\| \epsilon - \epsilon_\theta(z_i^{\hat{t}} \mid \hat{t}, h_i) \right\|^2 \quad (2)$$

where $\{i \mid M[i] = 1\}$ is the set of indices i of M such that $M[i] = 1$, representing the positions of $\mathbf{z}$ that are masked. $\hat{t}$ denotes the integer-valued diffusion step, sampled from the interval $[0, \hat{T}_{\text{max}}]$. $\{\phi, \theta\}$ denotes the set of parameters of the Transformer-based latent encoder and the diffusion head. Both modules are jointly optimized with the objective shown in Eq. (2).

3.1.2. Unconditional Pre-training

Unconditional pre-training is performed in a similar manner mentioned in Section 3.1.1 but without text conditions. It performs mask-based generative modeling without the text condition vector sequence $\mathbf{c}$. This process serves as a preliminary training phase before we have the model to learn how to follow text descriptions during audio generation. The benefits of this come in twofold: (1) it allows the model to first gain generative abilities for audio generation and relieves the burden of having to learn both generation and text relevancy in the same phase, (2) it allows us to utilize unpaired data since some audio datasets like AudioSet do not have text descriptions for all audio samples.

3.2. Inference Phase

Figure 1(b) illustrates the iterative mask-based parallel decoding method performed during the inference phase. The decoding process starts from a fully empty sequence $\mathbf{z}^{(0)}$ and a full mask $M^{(0)} = [1, 1, \cdots, 1] \in \{0,1\}^N$. The decoding process is composed of three main stages: random position selection, mask scheduling, and diffusion sampling.

We elaborate on them in the following sections.

3.2.1. Random Position Selection

Previous mask-based generative models for audio, such as Soundstorm, VampNet, and MAGNET, rely on discrete token representations and selectively predict token positions with low confidence scores at each decoding iteration. In contrast, our approach operates on continuous representations, making it infeasible to compute confidence scores. Therefore, at each decoding iteration t, we randomly select a subset of unpredicted positions in $\mathbf{z}^{(t)}$. We denote the set of indices of the positions to be predicted at each decoding iteration t as $M_{\text{pred}}^{(t)}$. These selected positions are generated in parallel rather than sequentially to reduce inference time.

3.2.2. Mask Scheduling

At each decoding iteration t, a masking scheduler determines the number of latents to remain masked, denoted as $\mu^{(t)}$, based on a fraction $\gamma^{(t)}$ of the N latents: $\mu^{(t)} = \lfloor \gamma^{(t)} \cdot N \rfloor$. The fraction $\gamma^{(t)}$ decreases over iterations following a predefined masking schedule, commonly defined as $\gamma^{(t)} = \cos\left(\frac{\pi}{2} \cdot \frac{t}{T}\right)$, where T is the total number of decoding iterations. This cosine masking schedule ensures that the number of masked latents decreases with each iteration, progressively increasing the amount of information available to the model for the next decoding iteration. The details of masking implementation are listed in Appendix D.1 and D.2.

3.2.3. Diffusion Sampling

At each decoding iteration t, the model generates latents for the positions specified by $M_{\text{pred}}^{(t)}$. Given the hidden representation $\mathbf{h}^{(t)}$ produced by the Transformer-based latent encoder, latents $z_i = z_i^0$ are sampled by following the reverse process (Ho et al., 2020; Nichol & Dhariwal, 2021) shown in Eq. (3),

$$z_i^{\hat{t}-1} = \frac{1}{\sqrt{\alpha_{\hat{t}}}}\left(z_i^{\hat{t}} - \frac{1-\alpha_{\hat{t}}}{\sqrt{1-\overline{\alpha}_{\hat{t}}}}\epsilon_\theta(z_i^{\hat{t}} \mid \hat{t}, h_i^{(t)})\right) + \sigma_{\hat{t}}\delta, \quad (3)$$

where $\sigma_{\hat{t}}$ denotes the noise level at diffusion sampling step $\hat{t}$ and δ denotes a vector drawn from $\mathcal{N}(0, \mathbf{I})$. Note that the diffusion sampling step $\hat{t}$ is distinct from the decoding iteration t.

During the iterative decoding process, classifier free guidance (cfg) (Ho & Salimans, 2021) is adopted to balance the text relevancy and audio fidelity. More details of cfg are elaborated in Appendix A.

4. Experimental Setup

4.1. Dataset Configurations

Multiple audio datasets are involved in this work. Specifically, we employ the AudioCaps (AC) (Kim et al., 2019) training split, which contains 145 hours of audio, and a combined dataset of AudioCaps (AC) and WavCaps (WC) (Mei et al., 2024), totaling 1200 hours of audio. Although AudioSet (AS) (Gemmeke et al., 2017) is currently the largest audio dataset which has about 5500 hours of audio data, since most of the audio samples in AS do not have text descriptions, this dataset is only used for unconditional pre-training. For evaluation, we evaluate our text-to-audio generation model on the AC evaluation set. There are 5 text descriptions for each sample in AC, and we follow AudioLDM by randomly selecting one text description as the text condition. For data preprocessing, we follow AudioLDM's recipe by segmenting each audio sample into 10 seconds and extracting the Mel spectrogram.

4.2. Implementation Details and Model Configurations

Our IMPACT model is composed of three main components, the VAE module, the Transformer-based latent encoder, and the diffusion head. For the VAE module, we directly adopt the VAE of AudioLDM to extract raw audio latents $\mathbf{z}' \in \mathbb{R}^{H \times W \times \text{ch}}$ from Mel spectrograms, where $H = 256$, $W = 16$ and $\text{ch} = 8$. A patching operation with factor $p = 4$ is performed to reduce the height and width dimension of $\mathbf{z}'$ into $\mathbf{z}'' \in \mathbb{R}^{\frac{H}{p} \times \frac{W}{p} \times \text{ch} * p^2}$. The patched latents $\mathbf{z}''$ are then flattened and projected to an embedding dimension of D resulting in $\mathbf{z} \in \mathbb{R}^{N \times D}$, where $N = \frac{H}{p} \times \frac{W}{p} = 256$. The patching operation reduces the sequence length to N and hence reduces computation during the generation process. For the text condition, concatenating CLAP (Wu et al., 2023) and Flan-T5 (Chung et al., 2024) embeddings on the time dimension results in a length of $L = 78$. The embedding dimension of the text conditioning vector sequence is also projected to D resulting in $\mathbf{c} \in \mathbb{R}^{L \times D}$. In this work, we develop two configurations of the IMPACT model, differing in the size of the Transformer-based latent encoder Enc: a base configuration and a large configuration. The base configuration uses an embedding dimension D of 768 and incorporates 24 transformer layers in the latent encoder. In contrast, the large configuration increases the embedding dimension to $D = 1024$ and employs 32 transformer layers in the encoder. For the diffusion head, we employ a multi-layer perceptron (MLP) adopted from MAR (Li et al., 2024) to effectively incorporate conditioning information from both diffusion steps and conditioning vectors h_i. No heavy parameterized attention-based layers are used in the diffusion head to avoid high computational costs during the diffusion loop of the diffusion sampling process at the inference phase. In both the base and large configurations,

the diffusion head remains identical, as scaling it up would substantially degrade inference speed. During training, the maximum number of diffusion steps $\hat{T}_{\text{max}}$ is set to 1000. During inference, the total number of diffusion sampling steps $\hat{T}$ is set to 100 unless specified explicitly. Additional information on the Transformer-based latent encoder and diffusion head can be found in Appendix D.3 and D.4.

During text-conditional training, we set the masking percentage factor $q = 0.7$. For all model training, we adopt the AdamW optimizer and set the learning rate to $5\text{e}{-5}$. For inference, by default, the total number of decoding iterations T is set to 64 unless otherwise specified. For classifier free guidance, we list the details in Appendix A.

4.3. Evaluation Metrics

In this work, generated audios are evaluated on five objective metrics: Fréchet Distance (FD) (Heusel et al., 2017), Fréchet Audio Distance (FAD) (Kilgour et al., 2019), Kullback–Leibler divergence (KL), Inception Score (IS) (Salimans et al., 2016), and Contrastive Language-Audio Pretraining (CLAP). The meanings and implementation details of these metrics are elaborated in Appendix F.

For subjective evaluation, following a similar approach to Tango (Ghosal et al., 2023), we assessed 30 generated audio samples based on text relevance (REL) and overall quality (OVL) but used a 1-to-5 rating scale instead of a 1-to-100 scale. Each sample was rated by at least 10 participants. See Appendix G for more details on the rating platform.

To evaluate inference speed, we measure the latency, also referred to as inference time, reported in seconds for generating a batch of audio samples on a single Tesla V100 GPU with 32GB VRAM.

5. Results

We evaluate our proposed text-to-audio framework across three key aspects. First, Table 1 reports results on the AudioCaps benchmark, using the objective and subjective metrics defined in Section 4.3, and contrasts our models against the current state of the art, while Table 2 presents ablations over key training configurations. Second, Figure 2 and Table 5 analyze the trade-off between objective performance and latency, focusing on the two primary parameters affecting inference time: the number of decoding iterations and diffusion steps. Finally, Figure 4 compares its latency and throughput with MAGNET-S, the fastest existing text-to-audio model.

5.1. Overall System-level Performance

Table 1 shows the system-level performance of text-to-audio generation on the AudioCaps evaluation set measured in both objective and subjective metrics. We compare our proposed models with current state-of-the-art models AudioGen (Kreuk et al., 2023), Tango (Ghosal et al., 2023), Tango-full-ft (Ghosal et al., 2023), Tango-AF&AC-FT-AC (Kong et al., 2024), Tango 2 (Majumder et al., 2024), EzAudio (Hai et al., 2024), MAGNET (Ziv et al., 2024), Make-an-Audio 2 (Huang et al., 2023), and AudioLDM2 (Liu et al., 2024). As shown in Table 1, our proposed IMPACT model outperforms current state-of-the-art models on the FD and FAD metrics. Specifically, while EzAudio-XL achieves the lowest FD score of 14.98 among existing baselines, Tango attains an FAD score of 1.73, our proposed IMPACT base and large models surpass these results on the respective metrics. For the KL metric, Tango-full-ft and Tango 2 both achieve a KL score of 1.12. In contrast, our proposed IMPACT base and large models attain better KL scores, falling only slightly behind the non-public model AudioLDM2-AC-large. Regarding the CLAP metric, our IMPACT models achieve a score of 0.364, slightly lower than Tango 2's 0.375, which may be attributed to Tango 2's use of the Direct Preference Optimization (DPO) (Rafailov et al., 2023) method for training. Note that EzAudio poses a huge gap in CLAP score compared to all other text-to-audio models. This is likely due to applying a data filtering method derived from CapFilt (Li et al., 2022) to discard audio-text pairs with CLAP scores lower than a certain threshold. Nonetheless, subjective evaluations indicate that our IMPACT models surpass Tango 2 and EzAudio-XL in text relevancy (REL), demonstrating their competitiveness in ensuring the generated audio closely corresponds to the provided text descriptions from human perspectives.

The last column of Table 1 compares the inference speed, also known as latency, of IMPACT with that of all other baseline models, each configured using hyperparameters that yield their overall optimal performance on objective metrics. While our IMPACT model with 32 decoding iterations achieves the second-lowest latency, just behind MAGNET-S, it significantly outperforms MAGNET-S across multiple objective metrics. Ablation studies on the two key factors that influence latency, the number of decoding iterations and diffusion steps, are presented in Section 5.3.

5.2. Training Configurations

5.2.1. Unconditional Pre-training

We investigate the benefits of unconditional pre-training by comparing three IMPACT models in Table 2: (a) trained exclusively with text-conditional training, and (b) and (c) additionally pre-trained unconditionally using different data scales. Compared to model (a), model (c) exhibits consistent gains across objective metrics. This improvement can be attributed to the unconditional pre-training phase, which enables the model to ensure the audio quality and fidelity for

Table 1. System level comparison of various text-to-audio generation models and their performance on objective and subjective metrics. Models publicly available were inferenced by us with parameters adjusted to achieve the best possible performance. Results for models that are not publicly available (marked with *) are directly reported from their original papers or from other existing works that have documented them. The notations "pt. data" and "tc. data" represent the training data durations, measured in hours, for pre-training and text-conditional training, respectively. For IMPACT models "pt. data" specifically refers to the duration of training data for unconditional pre-training. Detailed information on the training data is listed in Appendix E. The abbreviation "diff." denotes the number of diffusion sampling steps used for inference. The abbreviation "Lat." represents the latency of generating a batch of 8 audios measured in seconds. Best performance values are marked in bold. Second-best performance values are underscored.

AudioCaps	**pt. data**	**tc. data**	**# para**	**FD** ↓	**FAD** ↓	**KL** ↓	**IS** ↑	**CLAP** ↑	**REL** ↑	**OVL** ↑	**diff.**	**Lat.** ↓
Ground Truth	-	-	-	-	-	-	-	0.373	4.43	3.57	-	-
AudioGen*	-	$\approx$ 4000	1B	-	1.82	1.69	-	-	-	-	-	-
AudioGen v2	-	$\approx$ 4000	1.5B	16.51	2.11	1.54	9.64	0.315	-	-	-	37.2
Tango	-	145	866M	24.42	1.73	1.27	7.70	0.313	-	-	200	182.6
Tango-full-ft	$\approx$ 3333	145	866M	18.93	2.19	1.12	8.80	0.340	-	-	200	181.6
Tango-AF&AC-FT-AC	$\approx$ 400	145	866M	21.84	2.35	1.32	9.59	0.343	-	-	200	182.6
Tango 2	$\approx$ 3333	$\approx$ 80	866M	20.66	2.63	1.12	9.09	0.375	4.13	3.37	200	182.3
EzAudio-L (24kHz)	$>$ 5500	145	596M	15.59	2.25	1.38	<u>11.35</u>	**0.391**	4.05	3.44	50	29.1
EzAudio-XL (24kHz)	$>$ 5500	145	874M	14.98	3.01	1.29	**11.38**	<u>0.387</u>	4.00	3.35	50	40.2
MAGNET-S	-	$\approx$ 4000	300M	23.02	3.22	1.42	9.72	0.287	3.83	2.84	-	**6.9**
MAGNET-L	-	$\approx$ 4000	1.5B	26.19	2.36	1.64	9.10	0.253	-	-	-	24.8
Make-an-Audio 2	-	3700	160M	16.23	2.03	1.29	9.95	0.345	-	-	100	15.9
AudioLDM2-AC-large*	-	145	712M	-	1.42	**0.98**	-	-	-	-	-	-
AudioLDM2-full	-	29510	346M	32.14	2.17	1.62	6.92	0.273	3.74	3.19	200	96.1
AudioLDM2-full-large	-	1150k	712M	33.18	2.12	1.54	8.29	0.281	-	-	200	195.7
IMPACT base, dec iter 32	1200	1200	193M	<u>14.90</u>	**1.07**	<u>1.05</u>	10.06	0.364	-	-	100	<u>11.2</u>
IMPACT base, dec iter 64	5500	1200	193M	**14.72**	<u>1.13</u>	1.09	10.03	0.353	<u>4.31</u>	**3.51**	100	22.2
IMPACT large, dec iter 64	5500	1200	427M	**14.72**	1.17	1.07	10.53	0.364	**4.39**	<u>3.47</u>	100	28.4

generation. The subsequent text-conditional training then aligns the model to follow text descriptions more effectively while further enhancing audio quality and fidelity. Furthermore, although increasing the pre-training data from model (b) to model (c) slightly degrades KL, IS, and CLAP, subjective evaluations consistently favor model (c), underscoring the value of larger-scale data for unconditional pre-training. These findings demonstrate the importance of the unconditional pre-training phase and how it contributes to the performance of IMPACT models.

5.2.2. Text-conditional Training Data

By comparing models (c) and (d) in the IMPACT base configuration, we observe notable improvements in key metrics as the amount of text-conditioning training data is increased. Both models were unconditionally pre-trained with 5500 hours of data; however, model (d) was conditionally trained on 145 hours of data, while model (c) was conditionally trained on 1200 hours of data. This increase in text-conditioning training data led to substantial performance gains across several metrics. Model (c) achieved a lower FAD of 1.13 compared to 1.38 for model (d). Additionally, model (c) demonstrated a reduced KL divergence, scoring 1.09 versus 1.16 for model (d). Model (c) also surpassed model (d) in CLAP score, achieving 0.353 compared to 0.340. These findings confirm that increasing text-conditional training data enhances the performance of generated audio on objective metrics.

To replicate the training dataset configuration of Tango-full-ft (Ghosal et al., 2023), which was initially pre-trained on a large-scale dataset and subsequently fine-tuned on AC, we derive model (c') by performing text-conditional training on model (c) again using the AC training set. This additional training improves the performance on FD and FAD metrics, reflecting improved audio fidelity, but modestly degrades IS and CLAP scores, suggesting a potential trade-off between audio fidelity and quality, and also semantic consistency between the generated audio and the provided text prompt.

5.2.3. CLAP module

In this work, we adopt two text encoders, CLAP and FlanT5, to encode text into conditions. To assess the contribution of CLAP-based conditioning, in Table 3, we conduct ablation studies by removing the CLAP encoder during training while continuing to evaluate text relevance using CLAP scores. The results show that even without the CLAP module, the model's performance remains relatively stable across key metrics, with only minor differences observed. For example, the FAD and KL scores remained stable for IMPACT model (c), and the CLAP score showed only a negligible decrease from 0.353 to 0.352. These findings suggest

Table 2. Model performance comparisons between various IMPACT models on objective and subjective metrics. The † notation specifies that model (c') is trained with text-conditional training twice. Detailed latency values of baseline models and IMPACT are listed in Appendix B.

	AudioCaps	pt. data	tc. data	# para	FD ↓	FAD ↓	KL ↓	IS ↑	CLAP ↑	REL ↑	OVL ↑	Lat. ↓
(a)	IMPACT base, dec iter 64	-	1200	193M	14.86	1.35	1.17	9.75	0.346	-	-	22.2
(b)	IMPACT base, dec iter 64	1200	1200	193M	15.36	1.13	1.04	10.37	0.361	4.15	3.45	22.2
(b')	IMPACT base, dec iter 32	1200	1200	193M	14.90	**1.07**	1.05	10.06	0.364	-	-	11.2
(c)	IMPACT base, dec iter 64	5500	1200	193M	14.72	1.13	1.09	10.03	0.353	4.31	**3.51**	22.2
(c')	IMPACT base, dec iter 64	5500	1200†	193M	**13.86**	1.12	1.11	9.41	0.347	-	-	22.2
(d)	IMPACT base, dec iter 64	5500	145	193M	15.18	1.38	1.16	9.19	0.340	-	-	22.2
(e)	IMPACT large, dec iter 64	5500	145	427M	14.50	1.50	1.13	9.53	0.343	-	-	28.4
(f)	IMPACT large, dec iter 64	5500	1200	427M	14.72	1.17	1.07	10.53	0.364	**4.39**	3.47	28.4

Table 3. Performance comparisons of IMPACT models training with or without (w/o) CLAP embeddings.

dec iter 64	FAD ↓	KL ↓	IS ↑	CLAP ↑
IMPACT model (c)	1.13	1.09	10.03	0.353
w/o CLAP	1.13	1.10	10.22	0.352
IMPACT model (d)	1.38	1.16	9.19	0.340
w/o CLAP	1.49	1.10	9.19	0.344

that the model's performance, particularly in terms of semantic alignment measured by the CLAP score, is robust to the removal of the CLAP module, indicating that the model can effectively learn text-audio alignment without explicit CLAP-based input, relying solely on FlanT5 embeddings. Similarly, for IMPACT model (d), removing CLAP slightly increased the FAD from 1.38 to 1.49 and decreased the KL score from 1.16 to 1.10. However, the CLAP score still remained similar while removing the CLAP embeddings, indicating that semantic alignment between generated audio and text was not significantly impacted.

Table 4. Performance of IMPACT with single-pass decoding and 32-iteration decoding. Masking percentage factor q is set to 1.0 for training to yield the best performance for the single-pass model.

	FAD ↓	KL ↓	IS ↑	CLAP ↑
single-pass	12.26	2.57	2.84	0.125
dec iter 32	**1.07**	**1.05**	**10.06**	**0.364**

Table 5. Performance of IMPACT model (b) using different diffusion steps $\hat{T}$ for inference. Total decoding iterations set to 64.

$\hat{T}$	FAD ↓	KL ↓	IS ↑	CLAP ↑	Lat. ↓
50	1.57	1.15	10.02	0.342	**12.1**
100	**1.13**	**1.04**	**10.37**	0.361	22.2
150	1.30	1.06	10.31	0.363	31.6
200	1.19	1.06	10.36	**0.364**	41.9

5.3. Inference Configurations

5.3.1. Decoding Iterations

Table 4 studies the effectiveness of iterative decoding. The single-pass model, trained to generate audio latents in a single-pass, performs much worse than the model with 32 steps of iterative decoding. This showcases the importance of gradually generating the audio latents in an iterative manner since further iterations can leverage the previously generated latents as the condition for generation. Figure 2 shows the objective performance versus latency (inference time) of the baseline models and IMPACT model (b) using different decoding iterations during the inference phase. With 16 decoding iterations, IMPACT surpasses all publicly available baselines on the FAD metric, while only 8 iterations are sufficient to exceed these baselines in terms of the KL score. For IS scores, we observe an upward trend as the number of decoding iterations increases. For KL and CLAP scores, the gain of performance with 16 decoding iterations or more is little. When using up to 16 decoding iterations, we observe a strong correlation between performance and decoding iterations across all four objective metrics, indicating a trade-off between audio quality, fidelity, and inference speed, as more decoding iterations require more inference time. Notably, with 16 iterations, IMPACT not only surpasses all baselines on FAD and KL but also outperforms most of them on IS, all within a latency of just 5.7 seconds, which is significantly lower than any of the baseline models.

5.3.2. Diffusion Steps

Table 5 analyzes the effect of using different numbers of diffusion steps for inference. For diffusion steps 100, 150, and 200, the KL, IS, and CLAP scores are similar. For FAD, KL and IS scores, the best performance happens when using 100 steps. Given the fact that more diffusion steps result in higher latency, we conclude that using 100 diffusion steps is the most suitable parameter to achieve high performance on the objective metrics and to have low latency.

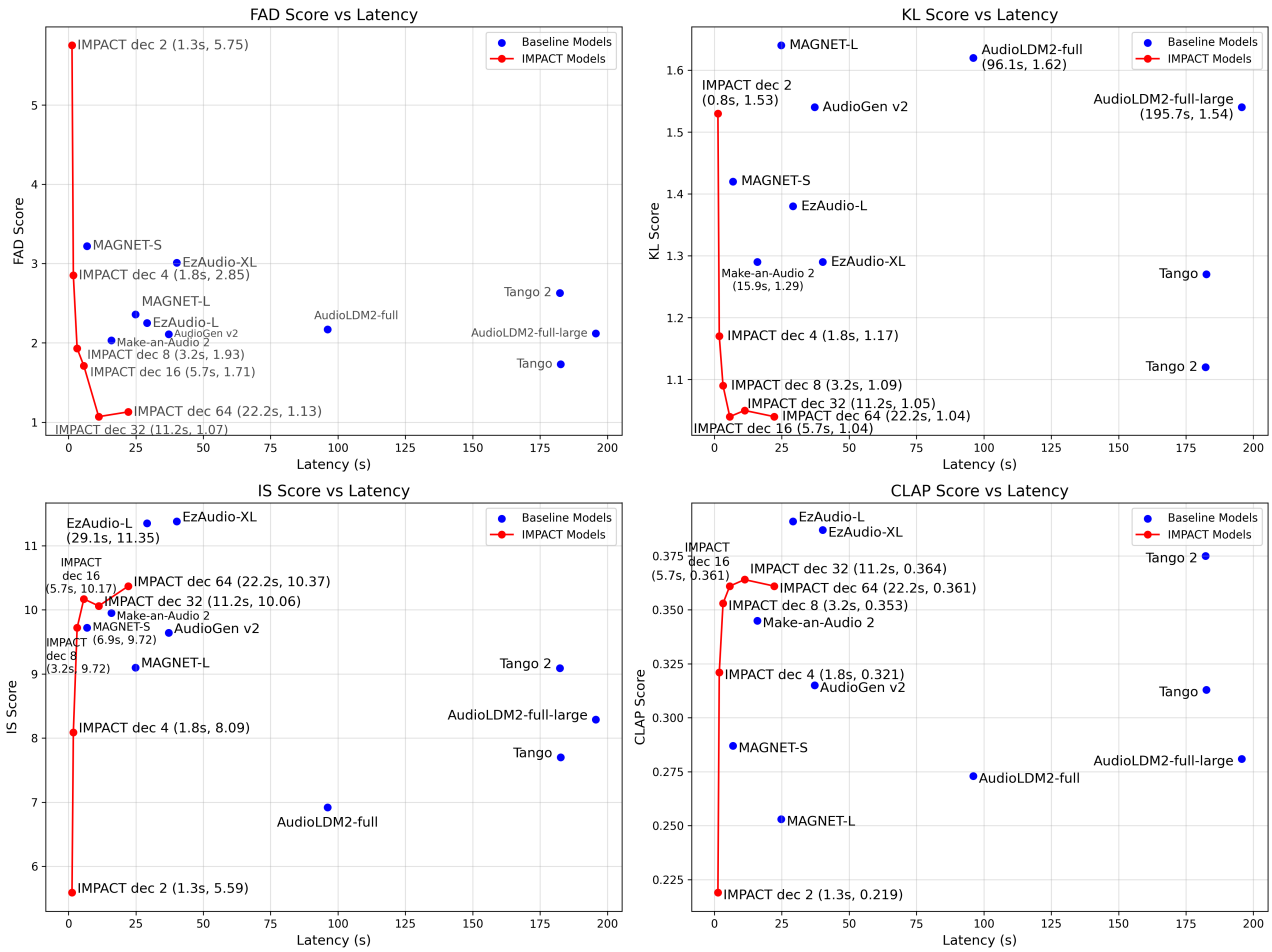

Figure 2. Performance of the baseline models and our proposed IMPACT model (b) with varying decoding iterations (dec iter), visualized by plotting objective metrics (FAD, IS, KL, and CLAP) against latency. Data points of the IMPACT model are plotted in red curves.

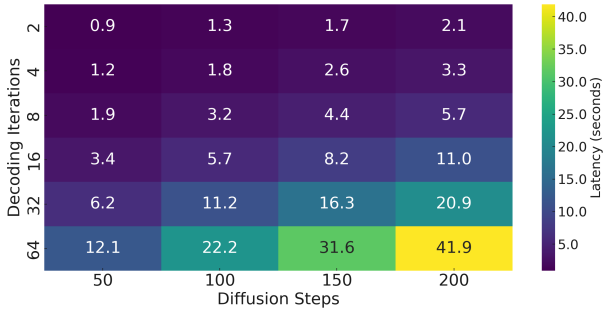

Figure 3. Heatmap visualizing latency measurements for IMPACT model (b) under varying decoding iterations and diffusion steps at a batch size of 8. Latency is depicted by color intensity, with brighter colors indicating higher values. Detailed objective performance values are provided in Appendix C.

5.3.3. Optimal Parameters

Figure 3 analyzes how decoding iterations and diffusion steps together affect the latency of IMPACT. It is observed that the impact of increasing diffusion steps on latency becomes more severe at higher decoding iterations. The latency increase due to decoding iterations is more tolerable at lower diffusion steps. Based on the results of Sections 5.3.1 and 5.3.2, we conclude that using 100 diffusion steps along with 16 to 64 decoding iterations is the optimal range of parameters for our IMPACT models.

5.4. Latency and Throughput Compared to MAGNET

IMPACT can flexibly adjust the number of decoding iterations to balance objective performance metrics and inference efficiency. To analyze this trade-off against the fastest existing baseline, Figure 4 presents the latency and throughput of MAGNET-S and our IMPACT base model with 4 decoding iterations across various batch sizes. Here, we define latency as the time in seconds to generate each batch of audio, and throughput as the number of audio samples generated per second. As reported in Table 6, even with just 4 decoding iterations, IMPACT already surpasses MAGNET-S in FAD, KL, and CLAP. IMPACT achieves faster inference

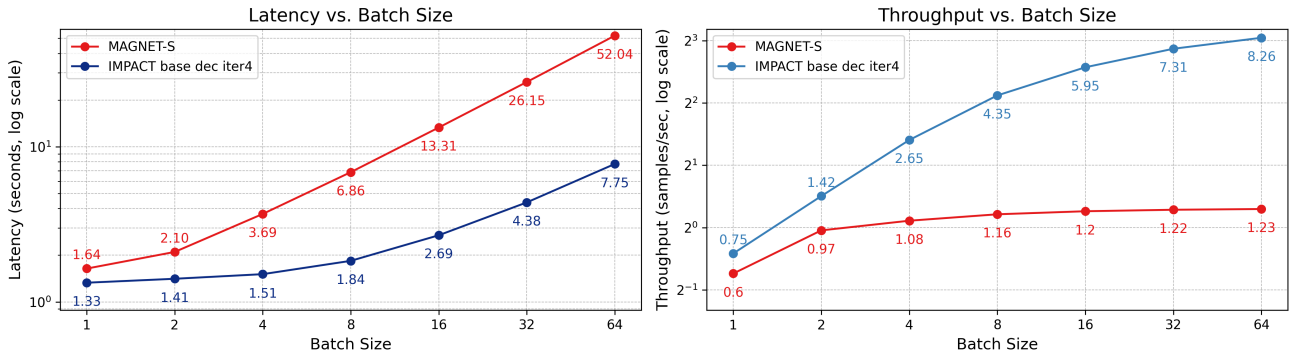

Figure 4. Batch latency and throughput comparisons between MAGNET-S and our IMPACT base model (b) with 4 decoding iterations.

than MAGNET-S, even with looping through the diffusion sampling process, due to its efficiency in reaching high objective performance with fewer decoding iterations. In contrast, MAGNET requires decoding across four levels of codebooks, which needs 50 iterations in total. Furthermore, when examining performance on batch sizes ranging from 1 to 64, IMPACT demonstrates consistently lower inference time and higher throughput, indicating that it scales better with larger batch sizes. For example, at a batch size of 64, IMPACT completes inference in 7.75 seconds with a throughput of 8.26 samples per second, compared to MAGNET-S's 52.04 seconds and 1.23 samples per second, respectively. This improvement highlights IMPACT's scalability and speed, making it a compelling option for applications that demand both high throughput and low latency on single-GPU systems, while also offering competitive or superior output quality.

Table 6. Comparison between MAGNET-S and IMPACT. The number of diffusion steps is set to 100 for IMPACT model (b).

	dec iter	**FAD** ↓	**KL** ↓	**CLAP** ↑
MAGNET-S	50	3.22	1.42	0.287
IMPACT model (b)	4	**2.85**	**1.17**	**0.321**

6. Conclusions

In this paper, we introduce IMPACT for text-to-audio generation, which combines the advantages of iterative mask-based parallel decoding and continuous latent representations through LDMs. By leveraging continuous latents with LDMs, IMPACT overcomes the limitations of discrete-token-based methods, offering superior audio fidelity and semantic alignment. Moreover, its mask-based decoding mechanism and adoption of a small diffusion head for generation ensure efficient inference, achieving faster inference speed than multiple baseline models and competitive latency compared to the fastest text-to-audio model, MAGNET-S. Apart from these methods, we propose unconditional pre-training and demonstrate the importance of it to achieve state-of-the-art performance on objective metrics such as FD and FAD. Our extensive experiments on the AudioCaps evaluation set highlight IMPACT's ability to balance audio quality, fidelity, text relevancy, and inference speed, addressing the trade-offs present in prior approaches. Additionally, human evaluations reaffirm its effectiveness in generating high-quality and contextually accurate audio content. These advancements position IMPACT as a robust solution for applications requiring both high fidelity and low latency.

Impact Statement

This paper presents an approach to enhance text-to-audio generation. The key contributions include improving audio fidelity, quality, and inference speed through iterative mask-based parallel decoding applied to continuous latents within LDMs. Additionally, the method incorporates an unconditional pre-training phase, enabling the utilization of unlabeled audio data, which significantly enhances the fidelity, quality, and text relevancy of the generated audio.

References

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. The million song dataset. 2011.

Borsos, Z., Sharifi, M., Vincent, D., Kharitonov, E., Zeghidour, N., and Tagliasacchi, M. Soundstorm: Efficient parallel audio generation, 2024. URL https://openreview.net/forum?id=KknWbD5j95.

Chang, H., Zhang, H., Jiang, L., Liu, C., and Freeman, W. T. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, 2022.

Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M.-H., Murphy, K., Freeman, W. T.,

Rubinstein, M., et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

Chen, G., Chai, S., Wang, G., Du, J., Zhang, W.-Q., Weng, C., Su, D., Povey, D., Trmal, J., Zhang, J., et al. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*, 2021.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., Adi, Y., and Défossez, A. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Defferrard, M., Benzi, K., Vandergheynst, P., and Bresson, X. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*, 2016.

Deshmukh, S., Elizalde, B., and Wang, H. Audio retrieval with wavtext5k and clap training. *arXiv preprint arXiv:2209.14275*, 2022.

Drossos, K., Lipping, S., and Virtanen, T. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 736–740. IEEE, 2020.

Fan, L., Li, T., Qin, S., Li, Y., Sun, C., Rubinstein, M., Sun, D., He, K., and Tian, Y. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. *arXiv preprint arXiv:2410.13863*, 2024.

Garcia, H. F., Seetharaman, P., Kumar, R., and Pardo, B. Vampnet: Music generation via masked acoustic token modeling. *arXiv preprint arXiv:2307.04686*, 2023.

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

Ghosal, D., Majumder, N., Mehrish, A., and Poria, S. Text-to-audio generation using instruction-tuned llm and latent diffusion model. *arXiv preprint arXiv:2304.13731*, 2023.

Hai, J., Xu, Y., Zhang, H., Li, C., Wang, H., Elhilali, M., and Yu, D. Ezaudio: Enhancing text-to-audio generation with efficient diffusion transformer. *arXiv preprint arXiv:2409.10819*, 2024.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Huang, J., Ren, Y., Huang, R., Yang, D., Ye, Z., Zhang, C., Liu, J., Yin, X., Ma, Z., and Zhao, Z. Make-an-audio 2: Temporal-enhanced text-to-audio generation. *arXiv preprint arXiv:2305.18474*, 2023.

Kilgour, K., Zuluaga, M., Roblek, D., and Sharifi, M. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In *Interspeech 2019*, pp. 2350–2354, 2019. doi: 10.21437/Interspeech.2019-2219.

Kim, C. D., Kim, B., Lee, H., and Kim, G. AudioCaps: Generating captions for audios in the wild. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 119–132, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1011. URL https://aclanthology.org/N19-1011.

Kong, Z., gil Lee, S., Ghosal, D., Majumder, N., Mehrish, A., Valle, R., Poria, S., and Catanzaro, B. Improving text-to-audio models with synthetic captions. In *Synthetic Data's Transformative Role in Foundational Speech Models*, pp. 1–5, 2024. doi: 10.21437/SynData4GenAI.2024-1.

Kreuk, F., Synnaeve, G., Polyak, A., Singer, U., Défossez, A., Copet, J., Parikh, D., Taigman, Y., and Adi, Y. Audiogen: Textually guided audio generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CYK7RfcOzQ4.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Li, T., Chang, H., Mishra, S., Zhang, H., Katabi, D., and Krishnan, D. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2142–2152, 2023.

Li, T., Tian, Y., Li, H., Deng, M., and He, K. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024.

Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., and Plumbley, M. D. AudioLDM: Text-to-audio generation with latent diffusion models. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21450–21474. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/liu23f.html.

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

Majumder, N., Hung, C.-Y., Ghosal, D., Hsu, W.-N., Mihalcea, R., and Poria, S. Tango 2: Aligning diffusion-based text-to-audio generative models through direct preference optimization. In *ACM Multimedia 2024*, 2024. URL https://openreview.net/forum?id=7lqptq5dLG.

Martín-Morató, I. and Mesaros, A. What is the ground truth? reliability of multi-annotator data for audio tagging. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 76–80. IEEE, 2021.

Mei, X., Meng, C., Liu, H., Kong, Q., Ko, T., Zhao, C., Plumbley, M. D., Zou, Y., and Wang, W. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3339–3354, 2024. doi: 10.1109/TASLP.2024.3419446.

Mesaros, A., Heittola, T., and Virtanen, T. Tut database for acoustic scene classification and sound event detection. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 1128–1132. IEEE, 2016.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Piczak, K. J. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1015–1018, 2015.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

Salamon, J., Jacoby, C., and Bello, J. P. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 1041–1044, 2014.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Sun, L., Xu, X., Wu, M., and Xie, W. Auto-acd: A large-scale dataset for audio-language representation learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 5025–5034, 2024.

Wu, Y., Chen, K., Zhang, T., Hui, Y., Berg-Kirkpatrick, T., and Dubnov, S. 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)*, pp. 1–5. IEEE, 2023.

Xu, Y., Zhang, S.-X., Yu, J., Wu, Z., and Yu, D. Comparing discrete and continuous space llms for speech recognition. In *Interspeech 2024*, pp. 2509–2513, 2024. doi: 10.21437/Interspeech.2024-1533.

Yuan, Z., Liu, Y., Liu, S., and Zhao, S. Continuous speech tokens makes llms robust multi-modality learners. *arXiv preprint arXiv:2412.04917*, 2024.

Ziv, A., Gat, I., Lan, G. L., Remez, T., Kreuk, F., Copet, J., Défossez, A., Synnaeve, G., and Adi, Y. Masked audio generation using a single non-autoregressive transformer. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Ny8NiVfi95.

A. Classifier Free Guidance (cfg)

Cfg is achieved by pushing away the noise predictions ϵ_{cond} generated with the text condition from the ones ϵ_{uncond} generated without the text condition at each diffusion sampling step, as shown in Eq. (4).

$$\epsilon = \epsilon_{\text{uncond}} + \beta_{\text{cfg}} \cdot (\epsilon_{\text{cond}} - \epsilon_{\text{uncond}}) \tag{4}$$

The cfg scaler $\beta_{\text{cfg}}^{(t)}$ at decoding iteration t controls how far to push away the generated latents from the unconditional ones. This scaler can be controlled by a cfg scheduler $\zeta^{(t)}$ during the iterative decoding process shown in Eq. (5).

$$\beta_{\text{cfg}}^{(t)} = 1 + \zeta^{(t)} \cdot (\beta_{\text{cfg_max}} - 1) \tag{5}$$

The cfg scheduler used in this work is $\zeta^{(t)} = \cos\left(\frac{\pi}{2} \cdot \frac{t}{T}\right)$. The max cfg scaler $\beta_{\text{cfg_max}}$ is set to 5.0 by default unless specified. The relationship between the cfg scaler $\beta_{\text{cfg}}^{(t)}$ and decoding iteration t is visualized in Figure 5. Using a higher max cfg scaler indicates that the generated latents are more forced to follow the text condition during generation.

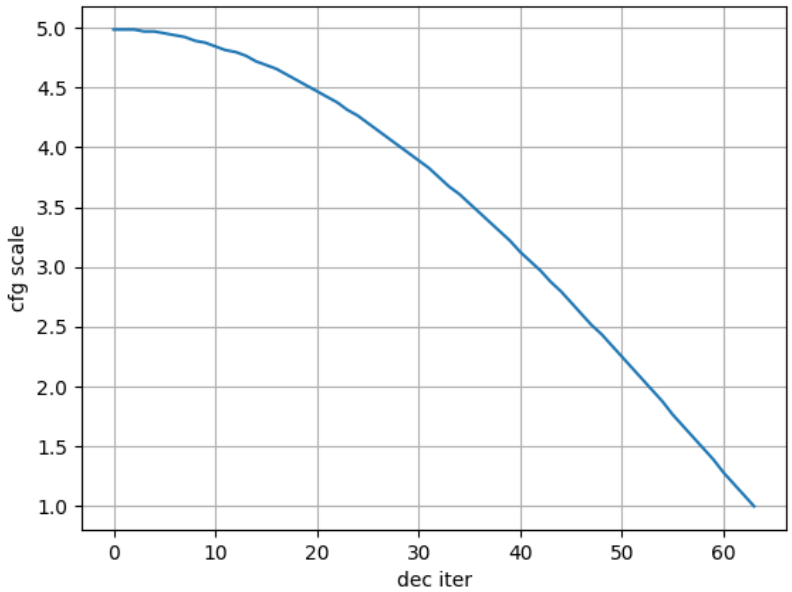

Figure 5. The cfg scaler $\beta_{\text{cfg}}^{(t)}$ at each decoding iteration during a 64-step iterative mask-based parallel decoding process.

Table 7 reports the performance of using different max cfg scalers for IMPACT model (c) trained with AS (5500 h) for unconditional pre-training and AC+WC (1200 h) for text-conditional training. The results demonstrate that there is a high correlation between the max cfg scaler and the CLAP score.

Table 7. Performance of IMPACT with different max cfg scalers $\beta_{\text{cfg_max}}$.

$\beta_{\text{cfg_max}}$	**FAD** ↓	**KL** ↓	**IS** ↑	**CLAP** ↑
1.0	3.36	1.36	6.74	0.261
2.0	1.51	1.08	9.07	0.328
3.0	1.39	**1.05**	9.85	0.346
4.0	1.25	1.07	**10.19**	**0.355**
5.0	**1.13**	1.09	10.03	0.353

B. System Level Latency

Table 8. Latency values of baseline models and IMPACT base with different batch sizes measured in seconds on a single Tesla V100 GPU with 32 GB VRAM. "diff steps" indicates the number of diffusion sampling steps $\hat{t}$ used for diffusion-based models. "coom" is the abbreviation of cuda-out-of-memory, indicating that the GPU is unable to process the forwarding of the model with the corresponding batch size.

Batch Size	diff steps $\hat{t}$	**1**	**2**	**4**	**8**	**16**	**32**	**64**
AudioGen v2	-	36.9	37.0	37.1	37.2	37.6	46.8	77.4
Tango 2	200	36.0	68.2	107.8	182.3	coom	coom	coom
EzAudio-L	50	12.6	14.2	15.8	29.1	55.3	108.9	coom
EzAudio-XL	50	14.4	14.5	21.4	40.2	78.4	155.4	coom
MAGNET-S	-	1.6	2.1	3.7	6.9	13.3	26.2	52.0
MAGNET-L	-	3.9	7.0	13.0	24.8	49.3	97.4	195.9
Make-an-Audio 2	100	3.5	11.1	12.6	15.9	34.1	41.3	81.7
AudioLDM2-full	200	46.7	48.8	57.8	96.1	148.2	275.4	coom
AudioLDM2-full-large	200	77.9	78.2	153.7	195.7	328.1	643.7	coom
IMPACT base, dec iter 64	100	19.8	19.9	20.6	22.2	24.0	27.9	37.3
IMPACT base, dec iter 32	100	10.1	10.4	10.5	11.2	12.6	15.3	20.5
IMPACT base, dec iter 16	100	5.0	5.2	5.4	5.7	6.6	8.5	13.1
IMPACT base, dec iter 8	100	2.5	2.6	2.8	3.2	4.1	5.7	9.5
IMPACT base, dec iter 4	100	**1.3**	**1.4**	**1.5**	**1.8**	**2.7**	**4.4**	**7.8**

Table 8 compares the latency values of baseline models and IMPACT base across various batch sizes (1, 2, 4, 8, 16, 32, 64) on a Tesla V100 GPU with 32 GB VRAM. Baseline models are configured using hyperparameters that yield their optimal performance on objective metrics. AudioGen v2[1] is the public version of AudioGen but is slightly different from the original AudioGen model published in (Kreuk et al., 2023). AudioGen v2 generates audio in 10 seconds, adopts discrete representations from a retrained EnCodec model, and no audio mixing augmentations are used during training.

The results highlight the efficiency and scalability of IMPACT compared to existing baseline models. Notably, IMPACT demonstrates significantly lower latency across all batch sizes when using fewer decoding iterations.

C. Combinations of Decoding Iterations and Diffusion Steps of IMPACT base

C.1. FAD

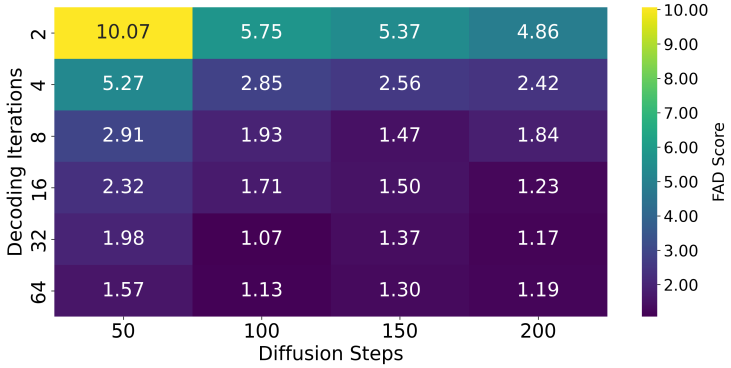

Figure 6. Heatmap visualizing FAD scores for IMPACT model (b) under varying decoding iterations and diffusion steps at a batch size of 8. FAD score is depicted by color intensity, with brighter colors indicating higher values.

[1] https://github.com/facebookresearch/audiocraft/blob/main/model_cards/AUDIOGEN_MODEL_CARD.md

C.2. KL

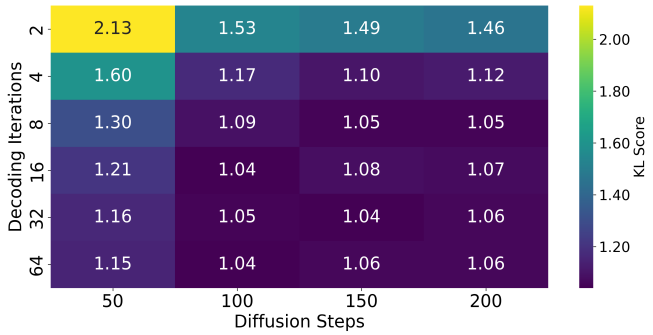

Figure 7. Heatmap visualizing KL scores for IMPACT model (b) under varying decoding iterations and diffusion steps at a batch size of 8. KL score is depicted by color intensity, with brighter colors indicating higher values.

C.3. IS

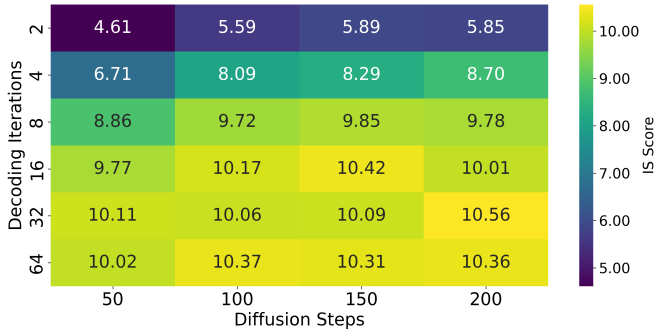

Figure 8. Heatmap visualizing IS scores for IMPACT model (b) under varying decoding iterations and diffusion steps at a batch size of 8. IS score is depicted by color intensity, with brighter colors indicating higher values.

C.4. CLAP

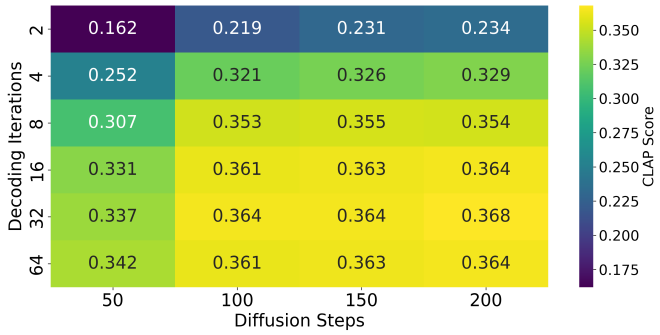

Figure 9. Heatmap visualizing CLAP scores for IMPACT model (b) under varying decoding iterations and diffusion steps at a batch size of 8. CLAP score is depicted by color intensity, with brighter colors indicating higher values.

D. Implementation Details

D.1. Random Position Selection

Throughout the mask-based decoding process, the mask $M^{(t)}$ at each decoding iteration t follows a subset property $M^{(t+1)} \subset M^{(t)}$ where $\subset$ indicates a strict subset in element-wise terms:

$$\forall\, i \in \{1, \ldots, N\}, \quad M^{(t+1)}[i] \leq M^{(t)}[i]. \tag{6}$$

This means a position i that is unmasked ($M[i] = 0$) at decoding iteration t must also have been unmasked at the rest of the decoding iterations. At each decoding iteration, the model predicts the positions that are different between $M^{(t+1)}$ and $M^{(t)}$, i.e., the positions where $M^{(t)}[i] = 1 \quad$ and $M^{(t+1)}[i] = 0$. We denote the set of indices of the positions to be predicted at each decoding iteration as $M^{(t)}_{\text{pred}}$. The behavior of this design at each decoding iteration $t+1$ will result in randomly selecting a subset of unpredicted positions of $\mathbf{z}^{(t)}$ to be generated. This is different from previous mask-based generative models Soundstorm, VampNet, and MAGNET that operate on discrete audio tokens, since they are able to select positions that have low confidence scores to predict at each decoding iteration.

D.2. Mask Scheduling

Another purpose of applying the cosine function is that it is a concave function within $[0, \frac{\pi}{2}]$. This causes the number of positions to be predicted at decoding iteration t, denoted as $|M^{(t)}_{\text{pred}}|$, increasing throughout the iterative decoding process. This ensures that more latents are generated when more previous content is available.

D.3. Transformer-based Latent Encoder

The Transformer-based latent encoder Enc_ϕ consists of two subencoders, Enc_{ϕ_1} and Enc_{ϕ_2}, which follows the scheme of a masked autoencoder (MAE) (He et al., 2022)[2]. As shown in Eq. (7), the masked positions of the audio latents $\mathbf{z}$ are dropped before forwarding it to the first encoder Enc_{ϕ_1} along with the concatenated text condition vector sequence $\mathbf{c}$. We denote the audio latent sequence with masked positions dropped as $\mathbf{z}_{\text{drop}}$. The reason for dropping the masked audio latents at this stage is to reduce both computational and memory requirements since a large portion of positions in $\mathbf{z}$ are masked.

$$\mathbf{g} = \text{Enc}_{\phi_1}(\text{concat}(\mathbf{c}, \mathbf{z}_{\text{drop}})), \mathbf{g} \in \mathbb{R}^{(L+(1-q)\cdot N)\times D} \tag{7}$$

At this point, we reinsert mask latent embeddings (placeholder) to the masked positions in $\mathbf{g}$, resulting in $\mathbf{g}' \in \mathbb{R}^{(L+N)\times D}$. $\mathbf{g}'$ is then forwarded to Enc_{ϕ_2} to produce the final hidden representation $\mathbf{h}$ mentioned in Eq. (1). The forwarding process of Enc_{ϕ_2} is shown in Eq. (8).

$$\mathbf{h} = \text{Enc}_{\phi_2}(\mathbf{g}'), \mathbf{h} \in \mathbb{R}^{(L+N)\times D} \tag{8}$$

The Transformer blocks of the Transformer-based latent encoder are initialized using pre-trained MAR checkpoints[3], as failing to do so results in poor performance.

D.4. Diffusion Head

In our work, we directly adopted the diffusion head of MAR's (Li et al., 2024). The diffusion head processes the input latents via a linear projection and then infuses time-dependent information through a timestep embedder. Condition vectors h_i are likewise projected before being added to the time embedding, forming a combined representation that is passed through a series of ResBlock layers (He et al., 2016). Each ResBlock leverages adaptive layer normalization (AdaLN) (Peebles & Xie, 2023), where the conditioning vectors modulate the normalized hidden states via learned shifts and scales. Following these residual transformations, a final linear layer produces the output (e.g., mean and variance for diffusion). Unlike the Transformer-based latent encoder, this module is able to be trained from scratch.

[2]The original MAE paper treats this as an encoder-decoder framework. However, in this work, we refer to both the encoder and the decoder as encoders and name the whole module as "transformer-based latent encoder" since the role of this module is to encode the audio latents into a hidden representation h to serve as the condition for the diffusion head.

[3]Checkpoints can be accessed from the official third-party MAR github repository https://github.com/LTH14/mar?tab=readme-ov-file#preparation

E. Dataset Information

Table 9 lists the pre-training data and text-conditional training data for each model. The pre-training data for IMPACT refers to the training data used for unconditional pre-training.

AS:AudioSet (Gemmeke et al., 2017) , AC:AudioCaps (Kim et al., 2019), WC:WavCaps (Mei et al., 2024), BBC:BBC sound effects, Cv2:Clotho v2 (Drossos et al., 2020), VGG:VGG-Sound, FSD50K:Freesound Dataset 50k[4], FS:Freesound Dataset, FTUS:Free To Use Sounds, SGE:Sonniss Game Effects, WSE:WeSoundEffects, PM:Paramount Motion, US:Urban Sound (Salamon et al., 2014), MI:Musical Instrument, MC:MusicCaps, GMG:Gtzan Music Genre, ESC50:Environmental Sound Classification (Piczak, 2015), AA:Audio-alpaca[5], AFAS:AF-AudioSet, AACD:Auto-ACD (Sun et al., 2024), ASQC:AS-Qwen-Caps, ASSLGC:AS-SL-GPT4-Caps, AASE:Adobe Audition Sound Effects[6], ASTK:Audiostock[7], MACS (Martín-Morató & Mesaros, 2021), ES:Epidemic Sound[8], WT:WavText5Ks (Deshmukh et al., 2022), TUT:TUT acoustic scene (Mesaros et al., 2016), FMA:Free Music Archive (Defferrard et al., 2016), MSD: Million Song Dataset (Bertin-Mahieux et al., 2011), LJS:LJSpeech[9], GGS:GigaSpeech (Chen et al., 2021).

Table 9. Training data configurations for each model.

		pre-train data	fine-tune data
	AudioGen	-	AS+BBC+AC+Cv2+VGG+FSD50K+FTUS+SGE+WSE+PM
	Tango	-	AC
	Tango-full-ft	AS+AC+FS+BBC+US+MI+MC+GMG+ESC50	AC
	Tango-AF&AC-FT-AC	AFAS, AC	AC
	Tango 2	AS+AC+FS+BBC+US+MI+MC+GMG+ESC50	AA
	EzAudio-L (24kHz)	AS+AACD+ASQC+ASSLGC	AC
	EzAudio-XL (24kHz)	AS+AACD+ASQC+ASSLGC	AC
	MAGNET-S	-	AS+BBC+AC+Cv2+VGG+FSD50K+FTUS+SGE+WSE+PM
	MAGNET-L	-	AS+BBC+AC+Cv2+VGG+FSD50K+FTUS+SGE+WSE+PM
	Make-an-Audio 2	-	AS+AC+WC+AASE+ASTK+ESC50+FSD50K+MACS+ES+US+WT+TUT
	AudioLDM2-AC-large*	-	AC
	AudioLDM2-full	-	AS+AC+WC+VGG+FMA+MSD+LJS+GGS
(a)	IMPACT base	-	AC+WC
(b)	IMPACT base	AS	AC
(c)	IMPACT base	AS	AC+WC
(c')	IMPACT base	AS	AC+WC & AC
(d)	IMPACT base	AC+WC	AC+WC
(d')	IMPACT base	AC+WC	AC+WC
(e)	IMPACT large	AS	AC
(f)	IMPACT large	AS	AC+WC

[4]https://annotator.freesound.org/fsd
[5]https://huggingface.co/datasets/declare-lab/audio-alpaca
[6]https://www.adobe.com/products/audition/offers/adobeauditiondlcsfx.html
[7]https://audiostock.net/
[8]https://www.epidemicsound.com/
[9]https://keithito.com/LJ-Speech-Dataset/

F. Objective Evaluation

F.1. FD and FAD

FD and FAD are metrics specifically designed to assess the fidelity of generated audio by measuring the distance between the distributions of embeddings from real and generated audio samples. A lower FD or FAD score indicates that the generated audio closely resembles real audio in terms of these perceptual features, reflecting higher fidelity and realism.

F.2. KL

KL divergence quantifies how the probability distribution of sound events in the generated audio differs from that of the real audio, with smaller values indicating that the generative model effectively captures the underlying distribution of the real audio data.

F.3. IS

IS measures both the quality and diversity of generated audio samples by computing the KL divergence between the conditional class distribution and the marginal class distribution over all samples using a pre-trained classifier. A higher IS suggests that the generated audio is high-quality and diverse.

For metrics FD, FAD, KL, and IS, we follow the implementation of the commonly used audioldm_eval[10] package.

F.4. CLAP

CLAP evaluates the semantic consistency between the input text and the generated audio by computing the cosine similarity between embeddings of input text prompts and generated audio in a shared embedding space learned by models trained to align textual descriptions with corresponding audio representations. A higher CLAP score signifies better alignment and that the generated audio accurately reflects the intended textual content. The CLAP model used for evaluation is clap-htsat-fused[11], which is different from the one[12] used for the text condition to avoid gaining advantage on the CLAP metric.

[10] https://github.com/haoheliu/audioldm_eval
[11] https://huggingface.co/laion/clap-htsat-fused
[12] https://huggingface.co/lukewys/laion_clap/blob/main/630k-audioset-fusion-best.pt

G. Subjective Evaluation Platform

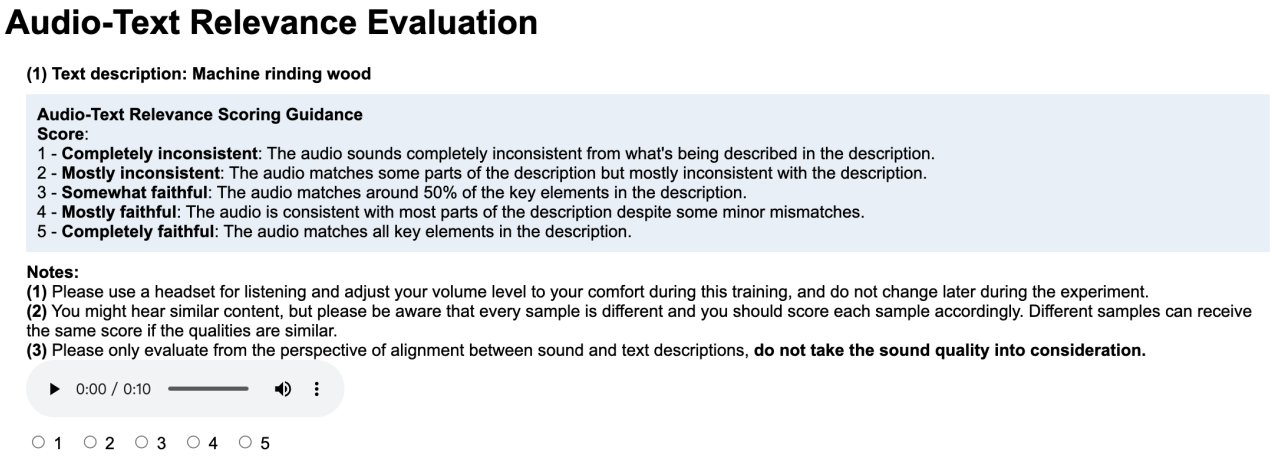

Figure 10. REL rating platform.

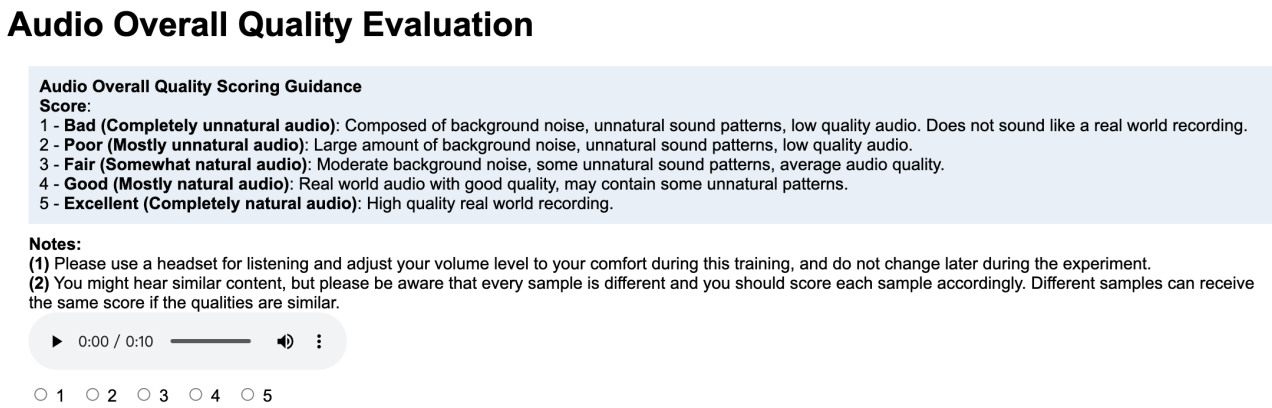

Figure 11. OVL rating platform.