Model Configurations

We list the model hyper-parameters of MaskAudioFlow in Table 4.

Vocoder

We train a BigVGAN (Lee et al. 2022) vocoder from scratch for the spectrogram to waveform generation. The synthesizer includes the generator and multi-resolution discriminator (MRD). The generator is built from a set of look-up tables (LUT) that embed the discrete representation and a series of blocks composed of transposed convolution and a residual block with dilated layers. The transposed convolutions upsample the encoded representation to match the input sample rate.

Text-to-Music generation

In this section, we perform a comparative analysis of audio samples generated by FlashAudio against several established music generation systems: 1) GT, the ground-truth audio; 2) MusicGen (Copet et al. 2023); 3) MusicLDM (Chen et al. 2024); 4) AudioLDM 2 (Liu et al.). The results are presented in the Table 5, and we have the following observations:

Model	**FAD** (↓)	**KL** (↓)	**CLAP** (↑)
GT	/	/	0.46
AudioLDM 2	3.81	1.22	**0.43**
MusicGen	4.50	1.41	0.42
MusicLDM	5.20	1.47	0.40
MaskAudioFlow	**3.35**	**1.20**	**0.43**

Table 5: Text-to-music generation comparison.

In terms of audio quality, MaskAudioFlow achieves the highest perceptual quality with FAD of 3.35 and KL of 1.20, which demonstrates the improved performance of the masked autoregressive flow-matching model compared to previous conditional flow matching or diffusion baselines. This highlights MaskAudioFlow's effectiveness in producing high-quality music samples and its generalization to audio-related domains.

Preliminaries: Flow-based Generative Models

Here, we give preliminaries on flow generative models. Denote data distribution as p_1, and tractable prior distribution as p_0. Most generative models work by mapping samples $\mathbf{x}_0 \sim p_0(\mathbf{x}_0)$ to data $\mathbf{x}_1$.

A flexible class of generative models (Karras et al. 2022; Song et al. 2020) (i.e., score-based diffusion models) based on turning noise $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$ into data $\mathbf{x}_0$ have been introduced. These models use the time-dependent process $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \varepsilon$, where α_t is a decreasing function of t and σ_t is an increasing function, and set both α_t and σ_t indirectly through different formulations of a stochastic differential equation (SDE).

Common to score-based diffusion models that the process $\mathbf{x}_t$ can be sampled dynamically using SDE, we consider the probability flow ordinary differential equation (ODE) with a velocity field:

$$d\mathbf{x}_t = \mathbf{v}_\theta(\mathbf{x}_t, t)dt, \tag{6}$$

where the velocity $\mathbf{v}$ is parameterized by a neural network θ, and $t \in [0, 1]$. By solving the probability flow ODE backwards in time from $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$, we can generate samples and approximates the ground-truth data distribution $p(x)$. We refer to Eq. 6 as a flow-based generative model.

However, this process is computationally expensive, especially for large network architectures parameterizing $\mathbf{v}_\theta(\mathbf{x}_t, t)$. It is proven that estimating a vector field $\mathbf{u}_t$ that generates a probability path between p_0 and p_1 is equivalent. To construct $\mathbf{u}_t$, we define a forward process, corresponding to a probability path $p_t(\mathbf{x} \mid \mathbf{x}_1)$ with a data sample $\mathbf{x}_t$ between p_0 and $p_1 = \mathcal{N}(0, \mathbf{I})$, with boundary condition $p_{t=0}(\mathbf{x} \mid \mathbf{x}_1) = p_0$ and $p_{t=1}(\mathbf{x} \mid \mathbf{x}_1) = \mathcal{N}(\mathbf{x} \mid \mathbf{x}_1, \sigma^2 \mathbf{I})$ for sufficiently small σ. While regressing $\mathbf{u}_t$ with the *Flow Matching* objective $\mathcal{L}_{FM}$ to learn the velocity field $\mathbf{v}(\mathbf{x}, t)$:

$$\mathcal{L}_{FM} = \min_\theta \mathbb{E}_{t, p_t(\mathbf{x})} \|\mathbf{v}_\theta(\mathbf{x}, t) - \mathbf{u}_t(\mathbf{x})\|^2, \tag{7}$$

For *Conditional Flow Matching*, we have:

$$\mathcal{L}_{CFM} = \min_\theta \mathbb{E}_{t, p_t(\mathbf{x}|\mathbf{x}_1)} \|\mathbf{v}_\theta(\mathbf{x}, t) - \mathbf{u}_t(\mathbf{x} \mid \mathbf{x}_1)\|^2, \tag{8}$$

Rectified Flows (RFs) (Albergo and Vanden-Eijnden 2022; Liu, Gong, and Liu 2022) define the forward process as straight paths between the data distribution and a standard normal distribution, where we have $\mathbf{x}^{\mathbf{OT}} = (1 - (1 - \sigma_{\min})t)\mathbf{x}_0 + t\mathbf{x}_1$, as the flow from $\mathbf{x}_0$ to $\mathbf{x}_1$.

The objective loss function can be written as:

$$\mathcal{L}_{OT-CFM} = \min_\theta \mathbb{E}_{t, p_t(\mathbf{x}|\mathbf{x}_1)} \|\mathbf{v}_\theta(\mathbf{x}^{\mathbf{OT}}, t) - \mathbf{u}_\mathbf{t}^{\mathbf{OT}}(\mathbf{x}^{\mathbf{OT}} \mid \mathbf{x}_1)\|^2, \tag{9}$$

These properties minimize the trajectory curvature and connect data and noise on a straight line. It has been demonstrated that flow probabilistic models (Ma et al. 2024a; Liu, Gong, and Liu 2022; Esser et al. 2024) can learn diverse data distribution in multiple domains, such as images and time series. In this work, we compare the flow formulation to existing DDPM in video-guided audio generative modeling and demonstrate its benefits.

Evaluation

To probe audio quality, we conduct the MOS-Q (mean opinion score) tests and explicitly instruct the raters to *"focus on examining the audio quality and naturalness."*. The testers present and rate the samples, and each tester is asked to evaluate the subjective naturalness on a 20-100 Likert scale.

To probe video-audio alignment, human raters are shown an audio and a video and asked *"Does the audio align with text faithfully?"*. They must respond with "completely", "mostly", or "somewhat" on a 20-100 Likert scale to score MOS-F.

Our subjective evaluation tests are crowd-sourced and conducted via Amazon Mechanical Turk. These ratings are obtained independently for model samples and reference audio. The screenshots of instructions for testers have been

Table 4: Hyperparameters of MaskAudioFlow. We use T and F to denote the time and frequency moe layers respectively.

Hyperparameter		
MaskAudioFlow	Transformer Layer	16
	Diffusion Head Layer	4
	Transformer Embed Dim	768
	Transformer Attention Headers	12
	Number of Parameters	160 M
BigVGAN Vocoder	Upsample Rates	[5, 4, 2, 2, 2, 2]
	Hop Size	320
	Upsample Kernel Sizes	[9, 8, 4, 4, 4, 4]
	Number of Parameters	121.6M

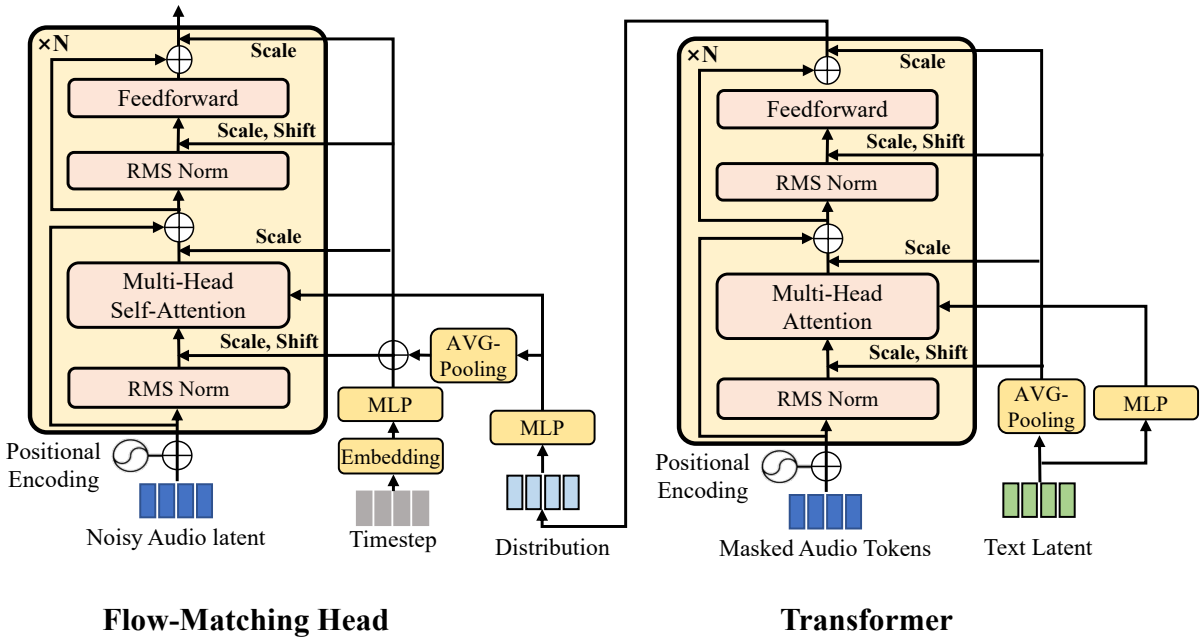
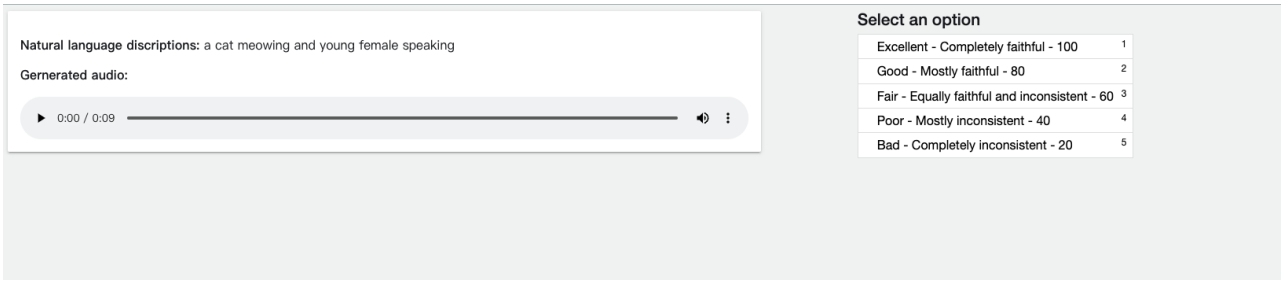


Figure 4: MaskAudioFlow transformer detailed architecture.

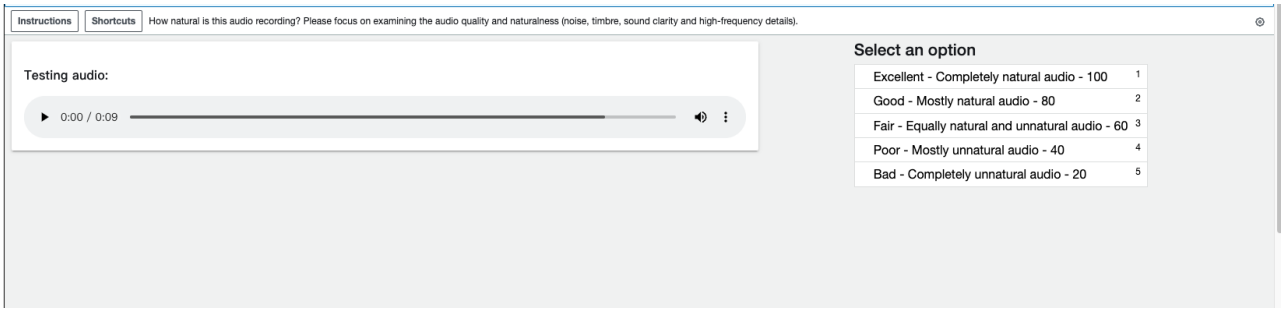
shown in Figure. We paid $10 to participants hourly and totally spent about $400 on participant compensation. A small subset of audio samples used in the test is available at https://MaskAudio.github.io/.

Visualization

In this section, we attach the visualization of iterative decoding.



(a) Screenshot of MOS-F testing.



(b) Screenshot of MOS-Q testing.

Figure 5: Screenshots of subjective evaluations.

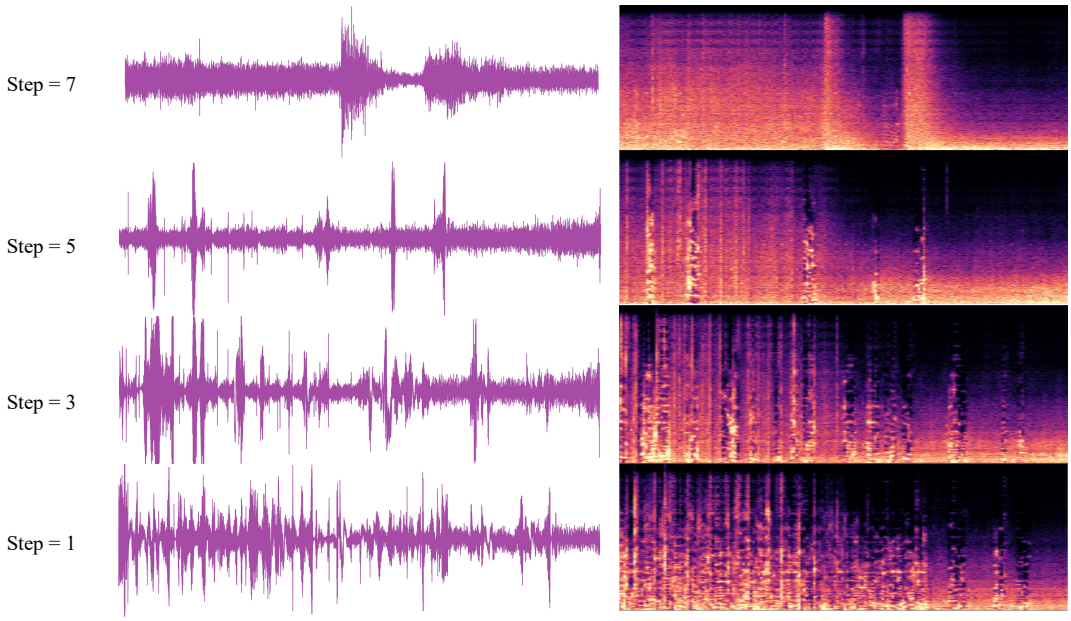


Figure 6: Visualization of iterative decoding. We use “People speaking with loud bangs followed by a slow motion rumble” as a prompt. Left: waveforms. Right: mel-spectrograms.