MFlowAudio: Efficient Text-to-Audio Synthesis via Mamba-based Stateful Flow Matching

Hao Dai[1,2], Panyu Chen[1], Jagmohan Chauhan[1]*
[1] Department of Computer Science, University College London
[2] Department of Engineering, King's College London
{hao.dai, panyu.chen.25, jagmohan.chauhan}@ucl.ac.uk

Abstract

Recent advancements in audio generation have been largely driven by Transformer-based diffusion models. However, these models suffer from quadratic complexity of self-attention, severely bottlenecking the long-form audio synthesis. To overcome this limitation, we propose MFlowAudio, a novel latent audio generation framework that synergizes the continuous-time dynamics of Flow Matching with a custom-designed TCMamba backbone. TCMamba uses an innovative dual-scan mechanism: a TimeMamba module to capture long-range causal dependencies with linear complexity, and a ChannelMamba module to model cross-channel correlations such as harmonic structures. Exploiting this structural foundation, we formulate the Stateful Flow Matching (SFM) paradigm. This framework inherently enables chunk-wise training and streaming generation, maintaining an $\mathcal{O}(1)$ caching complexity without incurring extra computational overhead. To enable fine-grained controllable synthesis, we devise a novel guidance mechanism that neutralizes vector field collisions precipitated by on-the-fly semantic transitions of prompts. Comprehensive empirical evaluations confirm that MFlowAudio yields generative fidelity on par with state-of-the-art baselines while establishing significant computational efficiency, achieving above 14.5% acceleration in generation and facilitating perceptually seamless streaming synthesis with a constant latency of 1.8 seconds. Demo:https://huggingface.co/spaces/mflowaudio/MFlowAudio

1 Introduction

Recent years have witnessed unprecedented advancements in generative artificial intelligence, revolutionizing domains such as text, image, video, and speech synthesis. A key driver of this progress has been the rapid evolution of diffusion models [Ho et al., 2020, Lu et al., 2022, Lipman et al., 2023, Tong et al., 2024]. While these models obtain remarkable achievements in image generation, audio generation has concurrently emerged as a field of increasing research interest and substantial practical impact [Kong et al., 2021, Ghosal et al., 2023, Huang et al., 2023a, Liu et al., 2023].

In practice, many contemporary Text-to-Audio (T2A) models leverage generative architectures originally conceived for the Text-to-Image (T2I) domain, such as U-Net [Ronneberger et al., 2015] / Transformers [Vaswani et al., 2017, Dosovitskiy et al., 2021, Peebles and Xie, 2023]-based diffusion backbones. While early models (e.g., AudioLDM [Liu et al., 2023, 2024], Make-an-Audio [Huang et al., 2023b,a], and Tango [Ghosal et al., 2023]) achieved success at 16kHz, recent works like Stable-Audio-Open [Evans et al., 2024], TangoFlux [Hung et al., 2024], and SoundCTM [Saito et al., 2025] have shifted to full-band (44.1kHz) synthesis. As the synthesis of short-form 16kHz audio matures, the research frontier is naturally advancing toward a far more demanding objective: generating extended, continuous full-band audio, such as long-form narratives.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, modeling long-form, high-fidelity audio presents unique challenges. Continuous audio signals strictly adhere to temporal causality and exhibit notoriously low semantic density per sample. Therefore, preserving acoustic fidelity necessitates extremely high sampling rates, which causes the sequence length to explode as the audio duration scales. This inherent characteristic imposes a high computational burden when attempting to capture long-range dependencies. Therefore, the Transformer, which is dominant in current generation paradigms, encounters a severe bottleneck, facing two fundamental constraints: (1) it lacks a strong temporal inductive bias, relying instead on injected positional encodings [Dosovitskiy et al., 2021, Su et al., 2024, Kazemnejad et al., 2023] to discern sequence order; (2) the self-attention mechanism exhibits computational and memory complexity that scales quadratically ($\mathcal{O}(L^2)$) with the sequence length L.

Meanwhile, existing diffusion and flow-matching methods inherently operate on a monolithic, single-pass generation paradigm. This structure exacerbates the memory explosion bottleneck during both training and the generation of long audio sequences, while simultaneously restricting dynamic controllability during generation. Although autoregressive (AR) models theoretically can generate sequences of arbitrary length, their empirical performance and perceptual fidelity fall short of those of their continuous-time generative counterparts. This dilemma motivates a critical inquiry: *Is it possible to design an efficient and controllable architecture within the prevailing diffusion/flow-matching paradigm to fulfill the rapidly foreseeable demands of long-form generation?*

To surmount the challenges of computational efficiency and length extrapolation, we propose MFlowAudio, a novel T2A architecture. It integrates the efficient State Space Models (SSMs) [Gu et al., 2022] into a continuous-time generative framework for T2A. Specifically, we introduce the Mamba [Gu and Dao, 2024, Dao and Gu, 2024]-based TCMamba block, a core module designed to jointly model the temporal and channel domains, which is optimized via a flow-matching objective. We demonstrate that this architectural design facilitates chunk-wise training and generation, and explore its capability for real-time streaming generation and dynamic prompt switching, which potentially enhances the practical viability of long-form T2A applications, e.g., interactive agents, real-time game soundscapes and controlled VR audios. Our core contributions are as follows:

- We propose Time-Channel Mamba, which unifies temporal causality and cross-channel correlation modeling via a novel dual-scan mechanism. It achieves linear complexity with respect to sequence length, eradicating the quadratic memory bottlenecks inherent in Transformer-based continuous-time generative models.
- We introduce the Stateful Flow Matching (SFM) algorithm, which natively empowers efficient streaming generation and chunked training with a $\mathcal{O}(1)$ caching complexity and negligible computational overhead. Furthermore, to enable controllable synthesis, we propose a State-Decoupled Conditional Guidance (SDCG) mechanism that mitigates vector field interference during prompt switching.
- Extensive experiments demonstrate that MFlowAudio achieves generation fidelity highly competitive with state-of-the-art baselines. Furthermore, our framework exhibits system-level advantages, delivering above 14.5% generation speedup and achieving a perceptually seamless real-time streaming generation with a constant low latency of 1.8 seconds.

2 Related Work

Text-to-Audio Generation. Text-to-audio (T2A) generation has rapidly advanced with the adoption of diffusion-based latent models. Early works such as AudioLDM [Liu et al., 2023] and Make-An-Audio [Huang et al., 2023b] employed U-Net backbones at 16 kHz, demonstrating strong audio quality and text alignment. Subsequent models, including AudioGen [Kreuk et al., 2023], Tango [Ghosal et al., 2023], and Stable Audio Evans et al. [2024], extended this paradigm to 44.1 kHz (CD quality), significantly improving perceptual fidelity. To reduce sampling cost, alternatives such as consistency models [Song et al., 2023] (e.g., SoundCTM [Saito et al., 2025]) and flow matching approaches (e.g., FreeAudio [Jiang et al., 2025], TangoFlux [Hung et al., 2024]) have been introduced, enabling faster generation while maintaining quality. However, most models still suffer from a quadratic complexity issue in long contexts due to Transformer-based backbones.

Mamba for Generative Models. SSMs and their modern variant, Mamba [Gu and Dao, 2024, Dao and Gu, 2024], have recently demonstrated strong performance in discriminative and autoregressive tasks, as exemplified by VisionMamba [Zhu et al.] and Samba [Ren et al., 2025]. Building on this

momentum, several studies have begun exploring their potential in generative modeling. For instance, DiMSUM [Phung et al., 2024] integrates Mamba with Transformer in a hybrid architecture to leverage complementary strengths for diffusion-based image generation, while U-Shape Mamba [Ergasti et al., 2025] adapts SSMs into a U-Net style framework; both designs incorporate specific scanning strategies tailored for image data. In the audio domain, early efforts such as the SSM-based Sashimi [Goel et al., 2022] model operate directly on raw waveforms, although its fidelity and text alignment lag behind mainstream T2A models. Meanwhile, SpeechSSM [Park et al., 2025], designed for long-form speech synthesis, demonstrates the potential of SSMs in capturing long-range dependencies.

To the best of our knowledge, MFlowAudio is the first framework that applies standalone Mamba to diffusion-based T2A generation, which is specifically designed to natively model the temporal and cross-channel characteristics of audio through its novel dual-path architecture.

3 Preliminaries

Diffusion Model and Flow Matching. Diffusion models formulate generation as transforming a simple noise distribution into a complex data distribution through a continuous process, typically grounded in the theory of stochastic differential equations (SDEs) or their deterministic counterparts, ordinary differential equations (ODEs). Given a data sample $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$, the forward process gradually perturbs it into noise by progressively injecting Gaussian noise:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

The diffusion model learns to reverse the corruption process by means such as parameterizing the denoising process $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ Ho et al. [2020] or approximating the score function Song et al. [2020]. Flow matching methods [Lipman et al., 2023, Albergo et al., 2025] learn a deterministic transformation from noise to data by defining a probability path between a prior distribution $p(z) = \mathcal{N}(0, \mathbf{I})$ and the data distribution $q(\boldsymbol{x}_0)$ achieved via the ODE:

$$\frac{d\boldsymbol{x}_t}{dt} = v_\theta(\boldsymbol{x}_t, t) \tag{2}$$

where $v_\theta(\boldsymbol{x}_t, t)$ is a time-dependent vector field learned to regress the conditional flow $u_t(\boldsymbol{x}_t|\boldsymbol{z})$ from noise $\boldsymbol{z}$ to data sample $\boldsymbol{x}_0$. The training objective minimizes the difference between the model's velocity and the target vector field under the conditional probability path:

$$\mathcal{L}_{FM} = \mathbb{E}_{t,p(\boldsymbol{x}_t|\boldsymbol{z}),p(\boldsymbol{z})}[||v_\theta(\boldsymbol{x}_t, t) - u_t(\boldsymbol{x}_t|\boldsymbol{z})||^2] \tag{3}$$

Once trained, the model v_θ defines a deterministic mapping from the prior to the data distribution. To generate a sample, one first draws $\boldsymbol{z} \sim p(\boldsymbol{z})$ and then solves the probability flow ODE from $t = 0$ to $t = 1$ using any numerical ODE solver (e.g., Euler or Runge-Kutta methods).

Diffusion and flow matching models employ a parallel decoding paradigm, where the entire sequence is decoded over multiple steps. Let L denote the sequence length and NFE (Number of Function Evaluations) represent the total number of sampling iterations. The computational cost can then be approximated as $c(v_\theta(L)) \times \text{NFE}$, where $c(v_\theta(L))$ is the cost per iteration. Accordingly, efficiency can be improved along two primary directions: reducing NFE or optimizing $c(v_\theta(L))$.

Transformer v.s. Mamba. The transformer backbone is ubiquitous in generative modeling. A common pipeline is to patchify the input into patches, project each patch to a fixed-dimension embedding, add positional encodings, and model global interactions via self-attention over the resulting token sequence. While effective at capturing long-range spatial dependencies, self-attention requires pairwise interactions among all tokens and thus incurs $\mathcal{O}(L^2)$ computation and memory for a sequence of length L. Moreover, because self-attention is permutation-invariant, it depends on injected positional bias (e.g., RoPE) to encode order, which hampers extrapolation to longer contexts.

To enable more efficient audio generation, we investigate RNN-like inductive biases, with Mamba as a promising candidate. Mamba builds upon a continuous state space representation, which is discretized for practical implementation. Its core SSM follows a recurrent formulation:

$$\boldsymbol{h}_t = \boldsymbol{A}\boldsymbol{h}_{t-1} + \boldsymbol{B}\boldsymbol{x}_t; \quad \boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{h}_t + \boldsymbol{D}\boldsymbol{x}_t \tag{4}$$

while a novel selection mechanism makes the transition and input matrices ($\boldsymbol{B}$ and $\boldsymbol{C}$) input-dependent, enabling context-dependent representations. Furthermore, a hardware-aware parallel

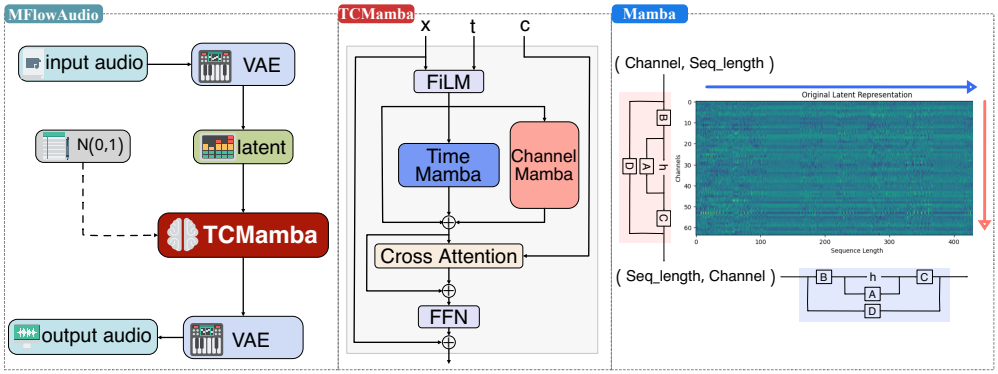

Figure 1: The key components of MFlowAudio: (left) the workflow of MFlowAudio, audio is encoded and decoded into latent using VAE and vocoder; (middle) the architecture of the TCMamba block; and (right) the dual-scanning mechanism of Mamba.

algorithm ensures efficient training despite the recurrent nature of the architecture. These properties make Mamba well-suited for long-form audio generation: it captures long-range dependencies with linear complexity $\mathcal{O}(L)$, while preserving temporal coherence across extended contexts.

A limitation of Mamba is its design for one-dimensional sequential data, while the dominant practice encodes audio as 2D time–frequency spectrograms, which are more compact and preserve essential acoustic structure. Mamba's native 1D recurrence, however, is not well matched to spectral patterns and cross-channel correlations. To bridge this gap and leverage Mamba's efficiency for long-range dependencies, we introduce a novel Mamba-based block tailored to operate on 2D spectrogram.

4 MFlowAudio

In this section, we will introduce the core design of MFlowAudio (Mamba-based Flow-Matching Text-to-Audio).

4.1 Time-Channel Mamba Block

State-of-the-art generative models for audio typically encode signals into a two-dimensional latent representation of shape (L, C), where L denotes length and C denotes channels. For instance, Stable Audio Open [Evans et al., 2024] uses a VAE to encode audio into a 64-channel 2D latent. Unlike image latents, the temporal axis of audio latents is separable: a latent of shape (L, C) can be partitioned into multiple contiguous chunks along the length dimension:

$$\begin{aligned}(L, C) &= \{(L^{(1)}, C), (L^{(2)}, C), \dots (L^{(k)}, C)\} \\ L &= L^{(1)} + L^{(2)} + \cdots + L^{(k)}\end{aligned} \tag{5}$$

each of which can be independently decoded into a coherent audio clip. This motivates us to design a time-latent oriented Mamba block that respects temporal separability while maintaining efficiency.

Time Mamba. Intuitively, audio can be treated as sequential data and modeled directly using Mamba. In our proposed TimeMamba block, the audio latent representation is processed as a sequence of shape $(1, C)$ along the temporal dimension. As illustrated in *Figure 1(right)*, the model scans the sequence from left to right, updating a hidden state $\boldsymbol{h}_t$ at each step to predict the subsequent representation. This design offers several key advantages:

- Inherent Temporal Modeling: The RNN-like recurrent state update naturally preserves chronological relationships and causal structure, which is fundamental to audio signals.
- Physically-Grounded Dynamics: SSMs are inherently designed to describe dynamical systems. Since audio is itself a physical phenomenon generated by a dynamical process, this formulation offers a more natural inductive bias for audio generation.
- Linear Computational Efficiency: By replacing the quadratic complexity of self-attention, TimeMamba achieves $\mathcal{O}(L)$ complexity with respect to sequence length.

Despite TimeMamba's strong capability in temporal modeling, its key shortcoming is the limited ability to capture coherent cross-channel structures, such as harmonic resonances across channels.

Channel Mamba. To overcome the limitation in cross-channel resonance modeling, we introduce ChannelMamba, a variant that reinterprets the latent as a sequence along the channel axis. ChannelMamba scans along the channel, which means the input shape is $(1, L)$, thereby capturing cross-channel correlations and harmonic structures. Since scanning across C requires a fixed length, which limits its ability to process audio of arbitrary length, we exploit the temporal separability of audio and instead scan shorter subsequences $(1, L^{(i)})$, enabling processing variable-length input while preserving local harmonic relationships.

The dual-scan mechanism is illustrated in *Figure 1(right)*: TimeMamba operates along the temporal dimension while ChannelMamba scans along the channel axis. By integrating FiLM and Cross-Attention for condition injection, we formulate the proposed TCMamba block, as depicted in *Figure 1(middle)*. Comprehensive architectural specifications are detailed in Appendix A.1.

Computational Complexity Analysis. For TimeMamba and ChannelMamba, both components build upon SSM, whose complexity scales linearly with sequence length. For an input latent $\boldsymbol{x} \in \mathbb{R}^{L \times C}$, the complexity of applying SSM updates is $\mathcal{O}(L \cdot C)$. For FiLM Modulation, it introduces minimal overhead, requiring only element-wise affine transformations. Therefore, the FiLM operation has complexity $\mathcal{O}(L \cdot C)$, maintaining the linear scaling characteristics of the overall architecture. For Cross-Attention: While cross-attention between the latent sequence (L, C) and conditioning embeddings (S, C) has complexity $\mathcal{O}(L \cdot S \cdot C)$, where S is the length of the conditioning sequence, in practice S is fixed and $\ll L$ for typical audio generation.

The overall complexity of TCMamba can be summarized as: $\mathcal{O}(L \cdot C + L \cdot S \cdot C) \approx \mathcal{O}(L)$ assuming C and S are constants or significantly smaller than L. This linear scaling enables efficient processing of long audio sequences while maintaining strong modeling capabilities.

Flow Matching Training. We adopt the Optimal Transport Flow Matching (OT-FM) framework [Tong et al., 2024] to train our generative model. OT-FM provides a simulation-free training objective, avoiding costly stochastic trajectory sampling, and instead relies on analytically defined optimal transport paths for efficient and stable learning. Formally, let $\boldsymbol{x}_0 \sim \mathcal{N}(0, \mathbf{I})$ denote the initial noise distribution and $\boldsymbol{x}_1 \sim q(\boldsymbol{x})$ be the target audio latent representation. For each timestep $t \in [0, 1]$, we define the linear interpolation path:

$$\boldsymbol{x}_t = t\,\boldsymbol{x}_1 + (1 - t)\,\boldsymbol{x}_0 \tag{6}$$

Our parameterized vector field $v_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$, conditioned on timestep t and semantic signal $\boldsymbol{c}$, is trained to approximate this velocity field $(\boldsymbol{x}_1 - \boldsymbol{x}_0)$. The Flow Matching loss is therefore:

$$\mathcal{L}_{FM} = \mathbb{E}_{\boldsymbol{x}_0 \sim \mathcal{N}(0,\mathbf{I}),\, \boldsymbol{x}_1 \sim q(\boldsymbol{x}),\, t \sim \mathcal{U}[0,1]} \Big[\big\| v_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}) - (\boldsymbol{x}_1 - \boldsymbol{x}_0) \big\|^2 \Big] \tag{7}$$

4.2 Stateful Flow-Matching

Given the recurrent design of Mamba, we design a novel Stateful Flow Matching (SFM) framework. Standard flow matching relies on a monolithic generation process, which imposes memory constraints when scaling to long sequences. Furthermore, extending it to generate consecutive temporal chunks typically necessitates intricate context-conditioning mechanisms to maintain temporal coherence. By exploiting the state-space representations, SFM circumvents these limitations by introducing an additional state cache. It achieves chunk-wise training and generation at near-zero additional computational cost with a constant cache, while maintaining the generation quality of the original one-time generation.

4.2.1 Chunked Training and Generation

To this end, we augment TCMamba block with an explicit state output, ensuring that each forward pass returns the terminal hidden state $\boldsymbol{h}_i$. Given that the Mamba is a sequential scanning structure, the state evolution along the temporal axis dictates that training or generation for any given chunk can resume from the final state of the preceding chunk. Consequently, by caching this terminal

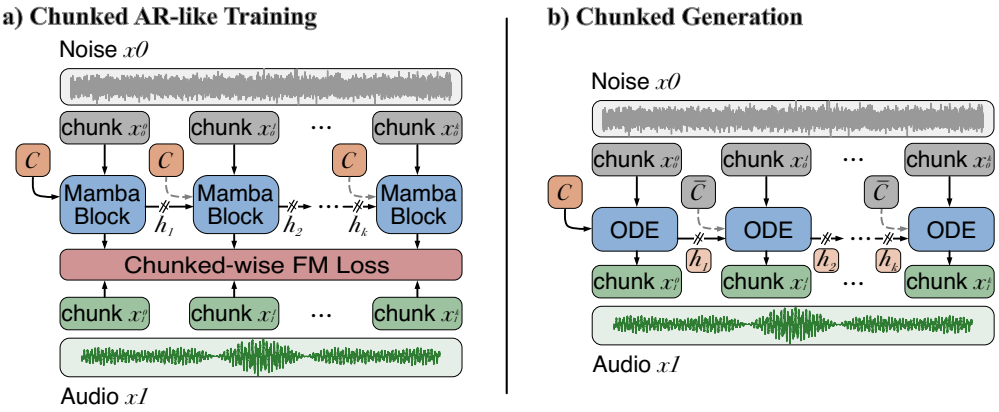

Figure 2: Beyond conventional monolithic training and generation, MFlowAudio also facilitates (a) chunked AR-style training and (b) chunked generation to support streaming.

state, TCMamba supports chunk-wise inference (as shown in Figure 2), remaining equivalent to the conventional single-pass paradigm (see Appendix A.4 for detailed theoretical analysis):

$$\begin{aligned} &\vec{v}_i, \boldsymbol{h}_{i+1} = v_\theta(\boldsymbol{c}; \boldsymbol{h}_i) \\ &\vec{v} = \{v_\theta(\boldsymbol{c}; \boldsymbol{h}_0), v_\theta(\boldsymbol{c}; \boldsymbol{h}_1), \cdots, v_\theta(\boldsymbol{c}; \boldsymbol{h}_i)\} \end{aligned} \tag{8}$$

$\mathcal{O}(1)$-Memory Temporal Propagation. The computational overhead introduced by this chunk-wise processing paradigm is negligible. Firstly, it incurs absolutely zero additional computational burden. Secondly, because the mechanism only requires caching the terminal state, the auxiliary memory cost for state propagation remains $\mathcal{O}(1)$ w.r.t. sequence length L. In contrast, the benefit in memory efficiency is substantial. By processing data strictly within a fixed chunk size of $L^{(c)}$ (where $L^{(c)} < L$ and is a constant), the overall spatial complexity is reduced to $\mathcal{O}(L^{(c)})$, thereby enabling the training and generation of long sequences under constrained memory budgets.

Streaming Generation and Time-to-First Latency. Beyond memory efficiency, the chunked formulation inherently enables real-time streaming generation. Because audio is consumed sequentially, playback commences immediately upon the synthesis of the initial chunk. Provided the per-chunk inference time remains strictly less than its acoustic duration (i.e., a Real-Time Factor < 1), MFlowAudio sustains a seamless pipeline capable of generating arbitrary-length audio with deterministic latency. This paradigm minimizes user-perceived delay, bounding it entirely by the Time-To-First-Chunk (TTFC). Furthermore, from a deployment perspective, the temporal amortization of computation profoundly alleviates serving bottlenecks, significantly maximizing server capacity and throughput for concurrent requests. See Appendix A.5 for details of implementation.

4.2.2 Dynamic and Controllable Conditional Generation

We further extend continuous generation by dynamically switching prompts at chunk boundaries. This mechanism brings benefits as it enables fine-grained temporal control (e.g., triggering vocal onset at 3.7s) and flexible compositional prompting. While some existing works attempt to achieve this via densely annotated datasets and bespoke architectures, our approach achieves temporal alignment through the intervention during chunked generation.

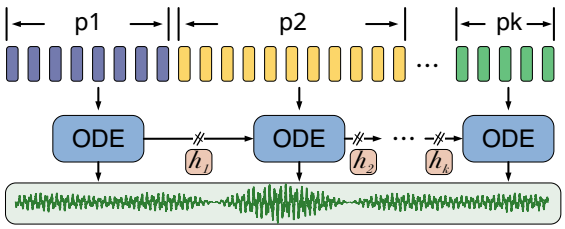

Figure 3: Switching prompts across variable-sized chunks to control the exact timing of events.

Dual-Driven Dynamics: Balancing State Inertia and Condition Injection. A critical issue of prompt switching is maintaining acoustic consistency across chunk boundaries, such as spatial reverberation, amplitude, and phase coherence. Due to the standard FM driving by prompt in a memoryless manner, generating consecutive chunks with distinct prompts inevitably introduces artifacts, e.g, audible clicks and phase discontinuities. In contrast, SFM is a dual-driven dynamic system, its vector field is governed by two distinct velocities: the current text condition and the historical state. When the prompt is null or transitions, SFM naturally relies on the "historical inertia" ($\vec{v}_{\text{state}}$) preserved within the states to perpetuate the prior semantic and acoustic properties:

$$\vec{v}_{\text{state}} = \boldsymbol{v}_\theta(\emptyset; \boldsymbol{h}_i); \quad \vec{v} = \vec{v}_{\text{uncond}} + \omega_{\text{cfg}} * (\vec{v}_{\text{state}} - \vec{v}_{\text{uncond}}) \tag{9}$$

This quasi-Markovian "inertia" guides the generation trajectory to glide smoothly along the data manifold. By functioning as a continuous regularizer across chunk boundaries, this state-driven momentum ensures acoustic coherence and spontaneously generates a natural transition zone.

State-Decoupled Conditional Guidance. When a semantically contradictory prompt is injected, the SFM confronts a severe control contention. In such circumstances, the model's vector field is simultaneously pulled by two antagonistic driving forces. This tension invokes vector field collision along the ODE integration trajectory. The resulting instability within the vector field dynamics induces destructive signal interference, leading to the representation collapse of either the historical state or the newly injected prompt, thereby degrading the generation quality.

To resolve this dilemma, we introduce a strategy that prioritizes the fidelity of the newly injected prompt while maximally retaining compatible historical state information. To achieve this, it is essential to extract the state-aware semantic vector, $\vec{p}$, from the composite vector field, which is distinct from conventional stateless $v_{\text{prompt}} = \boldsymbol{v}_\theta(\boldsymbol{c}, \emptyset)$. Specifically, we leverage the unconditional stateful vector field, $\vec{v}_{\text{state}}$, to explicitly decouple the prompt's pure contribution from the overall conditional vector field, $\vec{v}_{\text{cond}}$. Furthermore, by stripping away unconditional interference, we confine the guidance within the disentangled semantic subspace:

$$\vec{c} \triangleq \vec{v}_{\text{cond}} - \vec{v}_{\text{uncond}}, \quad \vec{s} \triangleq \vec{v}_{\text{state}} - \vec{v}_{\text{uncond}}, \quad \vec{p} = \vec{c} - \vec{s} \tag{10}$$

Ideally, the semantic direction $\vec{p}$ and the state direction $\vec{s}$ should remain orthogonal to prevent mutual interference. Therefore, to mitigate the vector field collision, we propose an Orthogonal Score Projection mechanism to extract and preserve the component of $\vec{p}^{\perp}$ that is orthogonal to $\vec{s}$, while truncating the component of $\vec{p}^{\parallel}$ that is parallel to and co-directional (reverse to preserve more history) with $\vec{s}$. This approach reinforces the impact of the new prompt within the pure semantic subspace while simultaneously preserving the underlying acoustic skeleton, thereby minimizing adverse state interference. The final rectified vector field is formulated as follows:

$$\vec{v}' = \vec{v}_{\text{uncond}} + \omega_{\text{cfg}} \Big[\vec{c} - \max\Big(0,\ \tfrac{\langle \vec{c}, \vec{s} \rangle}{\|\vec{s}\|^2} - \delta\Big)\, \vec{s} \Big]. \tag{11}$$

This State-Decoupled Conditional Guidance (SDCG) mechanism amplifies the guidance strength of the prompt with only an additional unconditional-stateful forward pass at the boundaries. It also introduces an adjustable scalar parameter $\delta \in [0, 1]$ to attenuate the historical state intensity. This δ serves as a versatile condition during generation, allowing for flexible, fine-grained control over the state-to-prompt influence ratio. See Appendix A.8 for more details.

5 Experiments

Following established practices in latent audio models, we adopt the VAE from Stable Audio Open as our audio encoder and utilize the T5 [Raffel et al., 2020] model as our text prompt encoder. The backbone of MFlowAudio is constructed by stacking multiple TCMamba blocks, with detailed architectural parameters provided in the Appendix A.1.

5.1 Experimental Setup

Datasets and Training Setup. We trained our model using the AudioCaps [Kim et al., 2019] and WavCaps [Mei et al., 2024] (AudioSet-SL subset only) training sets. All audio clips were resampled to 44.1 kHz. For testing, we evaluated our model on the AudioCaps full test set (4352 samples),

Table 1: Performance comparison across baselines on AudioCaps testset.

Duration	Model	Parameters	FD↓	KL↓	IS↑	FAD_v↓	CLAP↑	FAD_{openl3}↓
10s	AudioGen (16K)	1.5B	38.7259	2.5699	6.6716	6.6353	0.1950	192.4442
	AudioLDM2-48K	1.7B	36.2054	2.0261	5.442	5.4774	0.3827	148.1751
	Stable Audio Open	1B	32.0744	2.052	10.1462	3.7894	0.3442	109.0337
	TangoFlux-base	516M	16.9351	1.1724	**12.0486**	3.2514	**0.4722**	113.6228
	MFlowAudio-base	581M	**10.9178**	**1.1588**	10.8935	**2.9612**	0.4655	**95.6402**
	TangoFlux	516M	15.8429	1.1164	**12.5489**	**2.1963**	**0.4990**	102.8903
	MFlowAudio	581M	**11.8569**	**1.1158**	11.6865	2.5248	0.4957	**96.2363**
30s	AudioLDM2-48K	1.7B	52.5380	2.3984	5.1880	10.8777	0.2750	175.1954
	Stable Audio Open	1B	28.1238	1.9539	10.1849	4.3256	0.3619	108.4538
	TangoFlux-base	516M	24.6576	1.2224	9.0941	4.9399	0.4173	123.9447
	MFlowAudio-base	581M	**17.1580**	**1.1491**	**10.6577**	**3.6937**	**0.4333**	**103.0790**
	TangoFlux	516M	20.1872	**1.1189**	10.7179	2.9555	**0.4541**	102.2544
	MFlowAudio	581M	**17.5387**	1.1411	**11.3356**	**2.8981**	0.4448	**94.4700**

which was also resampled to 44.1 kHz. We trained our model using 8 NVIDIA H100 GPUs with a batch size of 64 per GPU for 100 epochs. See Appendix A.2 for the details of training and generation.

Baselines. We compared our model against three state-of-the-art full band (44.1 khz) text-to-audio generation models: AudioLDM2-48K [Liu et al., 2024], Stable Audio Open [Evans et al., 2024], and TangoFlux [Hung et al., 2024]. Stable Audio Open, TangoFlux, and MFlowAudio all share the same VAE architecture (Oobleck) and a T5-family text encoder. In addition, we also compared with an autoregressive 16khz model AudioGen [Kreuk et al., 2023]. For each baseline, we utilized the officially released pre-trained checkpoints and adhered to their recommended configurations to report their best performance.

Evaluation Metrics. To evaluate the generated audios, we employ six established metrics covering both intrinsic audio quality and text-audio alignment: CLAP [Wu* et al., 2023] Score (text alignment), PANNs [Kong et al., 2020]-based Inception Score (quality/diversity), VGGish [Hershey et al., 2017]-based and OpenL3 [Cramer et al., 2019]-based Frechet Audio Distance (FAD_v and FAD_{openl3}) [Kilgour et al., 2019] / Frechet Distance (FD) (quality/distributional match), and KL divergence (distributional match). To ensure a fair comparison and facilitate reproducibility, all evaluation metrics are computed using the publicly available audioldm-eval toolkit [Liu et al., 2023].

5.2 Results

Main Results. To validate the efficacy of our proposed architecture, we conduct a comprehensive comparison of generation quality. It is worth noting that TangoFlux, in addition to being trained on the audio dataset (TangoFlux-base), underwent fine-tuning using an auxiliary CRPO dataset. For comparison, we fine-tuned MFlowAudio utilizing a training set synthesised by TangoFlux. As presented in Table 1, under a near equivalent parameter budget, MFlowAudio demonstrates superior performance across primary acoustic quality metrics (FD, FAD_v, FAD_{openl3} and KL divergence). While slightly trailing TangoFlux in the CLAP and IS score, our model achieves an overall performance highly competitive with the current SOTA. These results suggest that TCMamba possesses audio processing capabilities comparable to Transformer-based architectures, underscoring its practical viability. We also conducted an evaluation on Clotho dataset, see Appendix A.6.

Generation Latency and Real-Time Streaming. To rigorously evaluate computational efficiency, we benchmarked the generation latency of all models under identical hardware conditions (a single RTX-5080 16GB GPU) using default configurations. For robustness, all reported latencies are averaged over 20 independent generation runs. As in Table 2, MFlowAudio demonstrates highest efficiency, achieving a speedup of 81.8% for 10-second and 14.5% for 30-second over baselines. Besides computational complexity, another reason affecting latency is that both StableAudio and TangoFlux generate a fixed-length latent (their maximum duration) and trim the output afterward, i.e., they always run the same amount of compute during denoising regardless of the target length, while MFlowAudio adapts its compute to the target length and generates audio.

Table 2: Generation latency comparison (s)

Duration	AudioLDM2-48K	Stable-Audio-Open	TangoFlux	MFlowAudio
10s	22.23±3.1 s	5.23±13 ms	3.51±3 ms	**1.93**±13 ms
20s	32.17±4.3 s	5.31±21 ms	3.54±12 ms	**2.17**±17 ms
30s	97.82±7.5 s	5.45±25 ms	3.55±5 ms	**3.10**±18 ms

Furthermore, we conducted a Time-To-First-Chunk (TTFC) evaluation to validate the model's streaming capabilities. As shown in Table 3, with a chunk size set to 64 (corresponding to approximately 3 seconds of audio), the initial decoding time is 1.8 seconds. Because this generation time is strictly shorter than the chunk's playback duration (i.e., Real-Time Factor < 1), perceptually seamless streaming generation is fully realizable. Regardless of the total target length, the user experiences a fixed wait time of 1.8 seconds before playback commences. Additionally, when increasing the batch size, decoding remains highly stable, yielding a maximum latency reduction of approximately 63.9%.

Table 3: One-time Latency v.s. Time-to-First-Chunk (s)

Duration	BatchSize	One-time	TTFC	Speedup	Reduction
10.0	1	1.9±19 ms	1.82±9 ms	1.1x	+5.8%
10.0	2	2.0±13 ms	1.85±21 ms	1.1x	+10.6%
20.0	1	2.1±26 ms	1.81±17 ms	1.2x	+16.8%
20.0	2	3.5±32 ms	1.85±31 ms	1.9x	+47.4%
30.0	1	3.1±18 ms	1.80±29 ms	1.7x	+41.7%
30.0	2	5.1±57 ms	1.85±37 ms	2.8x	+63.9%

Dynamic Injection and Boundary Smoothness Evaluation. To evaluate the model's capacity for dynamic prompt injection, we curated a subset of 1267 prompts from the standard benchmark that exhibit explicit temporal transitions (e.g., "followed by," "then," "after"). Each prompt was decomposed into a sequence of two to three distinct sub-prompts, which were uniformly anchored to predetermined onset timestamps at 0s, 3s, and 6s. Furthermore, to objectively quantify the acoustic continuity at the chunk boundaries (i.e., prompt switching points), we introduce two specialized signal processing metrics: Boundary Maximum Spectral Flux (BMaxSF) and Background Median Spectral Flux (BgMedSF). Building upon these, we formulate the Boundary-to-Background Ratio ($BBR = \frac{BMaxSF}{BgMedSF}$). A lower BBR indicates that the spectral variation at the transition boundary closely matches the natural variance of the background context, thereby serving as a robust indicator of seamless, artifact-free transitions (see Appendix A.7 for detailed definition and analysis.).

Table 4: Performance comparison of dynamic conditional injection

Model	FD↓	KL↓	IS↑	FAD_v↓	CLAP↑	FAD_{openl3}↓	BBR↓
TangoFlux (OneTime)	21.2885	1.1381	9.9338	2.1743	0.5208	110.7247	-
TangoFlux (Dyna-P)	22.6873	1.2788	6.2233	3.9077	0.4191	114.2842	11.6443
MFlowAudio (OneTime)	16.3070	1.1416	9.2517	2.2228	0.5186	96.8583	-
MFlowAudio (Dyna-P)	15.8575	1.2190	7.5717	2.8448	0.5087	101.2825	**4.3287**

The results are presented in Table 4. To establish an intuitive baseline, we included standard one-time generation as a reference. The data reveals that dynamic prompt injection incurs minimal degradation in overall audio fidelity; specifically, there is virtually no performance drop in CLAP and FD scores, confirming that the generated audio remains highly viable for practical applications. Furthermore, to benchmark transition smoothness, we evaluated the spectral flux metrics against TangoFlux (despite its lack of native support for dynamic prompting). As demonstrated, MFlowAudio exhibits a low BBR, indicating significantly smoother and more natural transitions across semantic boundaries.

Ablation Studies for SDCG. To validate the efficacy of proposed SDCG, we conducted a comprehensive set of ablation experiments. As detailed in Table 5, we evaluated SDCG against four alternative configurations: 1) w/o state: omitting state propagation entirely and relying instead on

Table 5: Ablation experiments of SDCG

Model	FD↓	KL↓	IS↑	FAD_v↓	CLAP↑	FAD_{openl3}↓	BBR↓
w/o state	19.3239	1.3015	6.8395	3.5030	0.4925	99.4748	7.4001
non-state prompt	16.9351	1.2767	7.2420	3.1169	0.4996	**99.2386**	5.4953
vanilla	16.0940	1.2536	**7.6370**	2.9659	0.5060	102.8763	**4.2633**
w/ naive sub	16.0056	1.2494	7.6201	2.8947	0.5075	102.3913	4.3578
w/ sdcg	**15.8575**	**1.2190**	7.5717	**2.8448**	**0.5087**	101.2825	4.3287

overlap-add techniques at boundaries; 2) non-state prompt: driving the ODE via a naive linear combination of stateless conditional field $\vec{v}_{\text{prompt}}$ and $\vec{v}_{\text{state}}$; 3) vanilla: utilizing the unadjusted conditional vector field $\vec{v}_{\text{cond}}$; and 4) naive subtraction: applying a direct difference $\vec{v}_{\text{cond}} - \vec{v}_{\text{state}}$.

The results demonstrate that SDCG is best on the fidelity/alignment metrics (FD, KL, FAD_v, CLAP) while trading off on IS, FAD_{openl3}, and BBR. This outcome is highly intuitive from a dynamic perspective: intensifying the prompt guidance inherently induces more decisive semantic shifts, whereas the vanilla model relies predominantly on state inertia, preserving acoustic smoothness at the cost of prompt adherence. This phenomenon highlights a trade-off between precise semantic control and boundary naturalness.

6 Conclusion and Discussion

We introduced MFlowAudio, a latent flow matching framework for text-to-audio generation that marries the efficiency of State Space Models with robust cross-channel modeling. By unifying TimeMamba and ChannelMamba via a dual-scan architecture, we realized a Stateful Flow Matching paradigm that circumvents both the quadratic complexity and the monolithic generation constraints of traditional Transformer-based diffusion models. Our approach delivers high-fidelity audio synthesis on standard benchmarks with high efficiency. In addition, its native support for chunk-wise streaming minimizes user-perceived latency, bringing the potential for real-time interactive systems and accommodating on-the-fly semantic interventions.

Currently, the dynamic control mechanism operates in a zero-shot manner, it relies purely on inference-time intervention rather than explicitly learning the underlying dynamics of acoustic transitions from data. The state guidance achieves transitions by mixing audio in different proportions, rather than by genuinely understanding how different elements should be composed. The model would benefit from further optimization on rich, sequentially annotated multi-prompt datasets. In addition, BBR mainly reflects changes at spectral boundaries and cannot capture other perceptual aspects, such as artifacts. And the subject evaluation should include more extensive A/B tests, providing a more comprehensive perceptual assessment of single-pass versus chunked generation, the SDCG ablations, and dynamic switching. Addressing these limitations will be the primary focus of our future research to unlock more complex controllable generation applications.

Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) grant, *MultiTasking and Continual Learning for Audio Sensing Tasks on Resource-Constrained Platforms [EP/X01200X/1]*, and NVIDIA Academic Grant, *CoVLA - Cloud-Edge Collaborative Data Generation Framework for VLA Learning*.

References

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Aurora Linh Cramer, Ho-Hsiang Wu, Justin Salamon, and Juan Pablo Bello. Look, listen, and learn more: Design choices for deep audio embeddings. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3852–3856, 2019. doi: 10.1109/ICASSP.2019.8682475.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Alex Ergasti, Filippo Botti, Tomaso Fontanini, Claudio Ferrari, Massimo Bertozzi, and Andrea Prati. U-shape mamba: State space model for faster diffusion. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, page 3242–3249. IEEE, June 2025. doi: 10.1109/cvprw67362.2025.00307. URL http://dx.doi.org/10.1109/CVPRW67362.2025.00307.

Zach Evans, Julian D. Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open, 2024. URL https://arxiv.org/abs/2407.14358.

Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria. Text-to-audio generation using instruction guided latent diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 3590–3598, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701085. doi: 10.1145/3581783.3612348. URL https://doi.org/10.1145/3581783.3612348.

Karan Goel, Albert Gu, Chris Donahue, and Christopher Re. It's raw! Audio generation with state-space models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 7616–7633. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/goel22a.html.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, 2017. doi: 10.1109/ICASSP.2017.7952132.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Jiawei Huang, Yi Ren, Rongjie Huang, Dongchao Yang, Zhenhui Ye, Chen Zhang, Jinglin Liu, Xiang Yin, Zejun Ma, and Zhou Zhao. Make-an-audio 2: Temporal-enhanced text-to-audio generation, 2023a.

Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: text-to-audio generation with prompt-enhanced diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023b.

Chia-Yu Hung, Navonil Majumder, Zhifeng Kong, Ambuj Mehrish, Amir Zadeh, Chuan Li, Rafael Valle, Bryan Catanzaro, and Soujanya Poria. Tangoflux: Super fast and faithful text to audio generation with flow matching and clap-ranked preference optimization, 2024. URL https://arxiv.org/abs/2412.21037.

Yuxuan Jiang, Zehua Chen, Zeqian Ju, Chang Li, Weibei Dou, and Jun Zhu. FreeAudio: Training-Free Timing Planning for Controllable Long-Form Text-to-Audio Generation, July 2025.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Drrl2gcjzl.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A metric for evaluating music enhancement algorithms, 2019. URL https://arxiv.org/abs/1812.08466.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. AudioCaps: Generating Captions for Audios in The Wild. In *NAACL-HLT*, 2019.

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 28:2880–2894, November 2020. ISSN 2329-9290. doi: 10.1109/TASLP.2020.3030497. URL https://doi.org/10.1109/TASLP.2020.3030497.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=a-xFK8Ymz5J.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CYK7RfcOzQ4.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. AudioLDM: Text-to-audio generation with latent diffusion models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 21450–21474. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/liu23f.html.

Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D. Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 32:2871–2883, May 2024. ISSN 2329-9290. doi: 10.1109/TASLP.2024.3399607. URL https://doi.org/10.1109/TASLP.2024.3399607.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=2uAaGwlP_V.

Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D. Plumbley, Yuexian Zou, and Wenwu Wang. WavCaps: A ChatGPT-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 1–15, 2024.

Se Jin Park, Julian Salazar, Aren Jansen, Keisuke Kinoshita, Yong Man Ro, and RJ Skerry-Ryan. Long-form speech generation with spoken language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=4AmFA0qNQ2.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182, 2023. doi: 10.1109/ICCV51070.2023.00387.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Hao Phung, Quan Dao, Trung Tuan Dao, Hoang Phan, Dimitris N. Metaxas, and Anh Tuan Tran. DiMSUM: Diffusion mamba - a scalable and unified spatial-frequency method for image generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=KqbLzSIXkm.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Liliang Ren, Yang Liu, Yadong Lu, yelong shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bIlnpVM4bc.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Koichi Saito, Dongjun Kim, Takashi Shibuya, Chieh-Hsin Lai, Zhi Zhong, Yuhta Takida, and Yuki Mitsufuji. SoundCTM: Unifying score-based and consistency models for full-band text-to-sound generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=KrK6zXbjfO.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), February 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Yusong Wu*, Ke Chen*, Tianyu Zhang*, Yuchen Hui*, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Forty-first International Conference on Machine Learning*.

A Appendix

A.1 Model Configurations

MFlowAudio follows a latent diffusion paradigm with the following components:

Audio VAE. We employ the VAE from Stable-Audio-Open as our latent audio encoder and vocoder. This VAE features a 64-channel bottleneck and achieves a compression factor of 2048. For a 44.1 kHz audio signal, the encoder produces a latent representation of shape $(L, 64)$, where the sequence length L is calculated as $L = \text{duration} \times 44100/2048$.

Text Encoder. We utilize the T5-base model as our text encoder. For each input prompt, the encoder produces a fixed-dimensional representation with a shape of $(128, 768)$, corresponding to 128 token embeddings each of dimensionality 768.

Time and Channel Mamba. Our TimeMamba module is implemented based on the official Mamba2 [Dao and Gu, 2024] source code (`https://github.com/state-spaces/mamba.git`), with a model dimension of 768 and a state dimension of 128. Similarly, ChannelMamba follows the same implementation but with a model dimension of 64 while maintaining the same state dimension of 128.

Conditional Generation. Conditioning signals guide the model toward semantically aligned outputs. Following the standard formulation of conditional diffusion models, the network input is represented as a triplet $(\boldsymbol{x}_t, t, \boldsymbol{c})$, where $\boldsymbol{x}_t$ is the noisy latent at timestep t, t is the diffusion timestep embedding, and $\boldsymbol{c}$ denotes the conditioning signal (e.g., text prompt embeddings).

For temporal conditioning, we adopt Feature-wise Linear Modulation (FiLM) [Perez et al., 2018], in which timestep t is projected via a learned embedding function.

FiLM. The timestep embedding is generated using a sinusoidal positional encoding, which is then projected through two linear layers with a GELU activation in between to produce the parameters (γ_t, β_t) with shape (1, 768), which are then applied within each TCMamba block to modulate activations:

$$\text{FiLM}(\boldsymbol{x}_t, t) = \gamma_t \odot \boldsymbol{x}_t + \beta_t \tag{12}$$

For semantic conditioning, we incorporate cross-attention to align the noisy latent representation with the conditioning signal.

Cross-Attention. The cross-attention mechanism employs 16 attention heads, and the total model dimension is 768. The key and value projections are derived from the T5 text embeddings, while the query projection is obtained from the latent representation. It is defined as:

$$\text{CrossAttn}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V \tag{13}$$

where $Q = W_Q \cdot \boldsymbol{x}_t, K = W_K \cdot \boldsymbol{c}, V = W_V \cdot \boldsymbol{c}$, and d is the feature dimension for scaling. This design enables the model to dynamically attend to the most semantically relevant components of c, ensuring faithful alignment between conditioning inputs and generated audio.

Overall Architecture. The overall architecture begins with an initial linear projection layer that maps the 64-channel latent input to a 768-dimensional feature space. This is followed by a stack of 48 identical TCMamba modules. Each TCMamba module contains a TimeMamba block and a ChannelMamba block, augmented with FiLM and cross-attention. The network concludes with a final linear projection layer that maps the 768-dimensional features back to the original 64-channel latent space.

A.2 Training and Sampling Details

The training procedure, outlined in *Algorithm 1*, minimizes the loss function depicted in *Figure 6*. The AdamW optimizer was employed with an initial learning rate of 1×10^{-4}, decayed using a

cosine annealing scheduler over two decay cycles. During generation, as detailed in *Algorithm 2*, we employ an Euler solver with a fixed step size of 0.04, and the CFG scale ω_{cfg} was set to 2.5.

Algorithm 1: Stateful Flow Matching Training

Input : datasets $(q(\boldsymbol{x}), \boldsymbol{c}), v_\theta, t_{\text{scheduler}}, \epsilon = 0.1$

for *epoch in epochs* **do**
  `// Sample data`
  Sample $\boldsymbol{x}_1 \sim q(\boldsymbol{x}), \boldsymbol{x}_0 \sim \mathcal{N}(0, \mathbf{I}), t \sim t_{\text{scheduler}}$
  if $random() < \epsilon$ **then**
    $\boldsymbol{c} \leftarrow \emptyset$
  end if
  else
    $\boldsymbol{c} \leftarrow \text{Encoder}(\boldsymbol{c})$
  end if
  `// Interpolate`
  $\boldsymbol{x}_t \leftarrow t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_0$
  `// full sequence training`
  `// Predict velocity`
  $\hat{v_\theta} \leftarrow v_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$
  `// Compute loss`
  $\mathcal{L}_{FM} \leftarrow ||\hat{v_\theta} - (\boldsymbol{x}_1 - \boldsymbol{x}_0)||^2$
  `// chunked-wise training`
  for *each chunk* i **do**
    $\boldsymbol{x}_0^i \leftarrow \boldsymbol{x}_0[i]$, $\boldsymbol{x}_1^i \leftarrow \boldsymbol{x}_1[i]$
    $\boldsymbol{x}_t^i \leftarrow t\boldsymbol{x}_1^i + (1-t)\boldsymbol{x}_0^i$
    `// Predict velocity`
    $\hat{v}_\theta^i, h^i \leftarrow v_\theta(\boldsymbol{x}_t^i, t, \boldsymbol{c}, h^{i-1})$
    `// Compute loss`
    $\mathcal{L}_{FM}^i \leftarrow ||\hat{v}_\theta^i - (\boldsymbol{x}_1^i - \boldsymbol{x}_0^i)||^2$
  end for
  `// Chunked-wise FM loss`
  $\mathcal{L}_{FM} \leftarrow \sum_i \mathcal{L}_{FM}^i$
  Update v_θ using AdamW optimizer
end for

A.3 Subjective Evaluation

We conducted a subjective evaluation with 25 participants using a blind listening test to assess randomly selected audio samples. Participants rated the samples across two dimensions: Text Relevance (REL) and Overall Quality (OVL), both on a 5-point Likert scale. As shown in Table 6, our approach slightly outperforms TangoFlux regarding overall quality, highlighting the effectiveness of the MFlowAudio framework in enhancing generative fidelity.

We further conducted a subjective evaluation of the audio generated at prompt switches, focusing on artifacts, clicks/pops, and transition naturalness. Three metrics were used: Boundary Artifact Score (BAS): Perceived severity of audible discontinuities at chunk boundaries; Click/Pop Occurrence (CPO): Presence of transient clicks or pops in the generated audio; and Transition Naturalness (TN): Smoothness and plausibility of transitions when the text prompt is switched. The results (Table 7) show that our method produces more natural transitions with fewer clicks and pops.

A.4 Equivalence of Chunked and One-Time Integration

In Flow Matching models, the generation trajectory is solved by integrating the Ordinary Differential Equation (ODE). With the assumption of a constant semantic condition (i.e., the text prompt c remains

Algorithm 2: MFlowAudio Generation

Input : $v_\theta, \omega, \delta, \boldsymbol{c}$, step-size=0.04

$\boldsymbol{x}_t \sim \mathcal{N}(0, \mathbf{I})$

`// One-time Generation`

for *t in [0, 1] with step-size* **do**

  $\vec{v}_{cond} \leftarrow v_\theta(\boldsymbol{x}_t, t, c)$

  $\vec{v}_{uncond} \leftarrow v_\theta(\boldsymbol{x}_t, t, \emptyset)$

  $\vec{v} \leftarrow \vec{v}_{uncond} + \omega(\vec{v}_{cond} - \vec{v}_{uncond})$

  $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t + \text{step-size} \cdot \vec{v}$

end for

`// Chunked generation with SDCG`

for *each chunk* L_c **do**

  $\boldsymbol{x}_t^c \leftarrow \boldsymbol{x}_t[c]$

  for *t in [0, 1] with step-size* **do**

    $\vec{v}_{cond}^c, h^c \leftarrow v_\theta(\boldsymbol{x}_t^c, t, c, h^{c-1})$

    $\vec{v}_{uncond}^c, h_u^c \leftarrow v_\theta(\boldsymbol{x}_t^c, t, \emptyset, h_u^{c-1})$

    if $\boldsymbol{c}^c \neq \boldsymbol{c}^{c-1}$ **then**

      $\vec{v}_{state}^c \leftarrow \vec{v}_{uncond}^c$

      $\vec{v}_{uncond}^c, h_u^c \leftarrow v_\theta(\boldsymbol{x}_t^c, t, \emptyset, h_u^0)$

      $\vec{v}^c \leftarrow \text{SDCG}(\omega, \vec{v}_{cond}^c, \vec{v}_{state}^c, \vec{v}_{uncond}^c)$

    else

      $\vec{v}^c \leftarrow \vec{v}_{uncond}^c + \omega(\vec{v}_{cond}^c - \vec{v}_{uncond}^c)$

    end if

    $\boldsymbol{x}_t^c \leftarrow \boldsymbol{x}_t^c + \text{step-size} \cdot \vec{v}^c$

  end for

end for

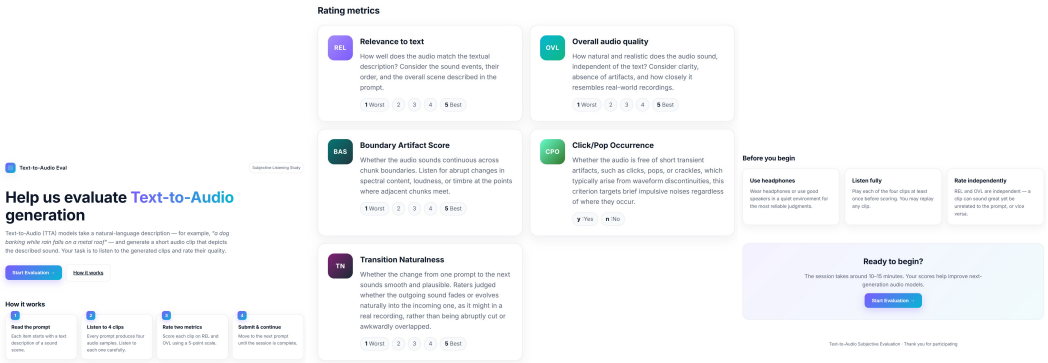

Figure 4: evaluation platform for subjective evaluations.

invariant), we establish that the state-passing chunked integration is mathematically strictly equivalent to the monolithic one-time global integration. This equivalence provides a theoretical guarantee for our model's capacity to generate continuous, seamless audio sequences.

Definition A.1 (Global and Streaming ODE Systems)**.** Let the global target audio sequence be denoted as $\boldsymbol{x} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(K)}]$, where $\boldsymbol{x}^{(i)}$ represents the i-th chunk. Assuming the existence of an ideal causal global vector field $\vec{v}_{global}$, its corresponding global ODE system is formulated as:

$$\frac{d\boldsymbol{x}_t}{dt} = \vec{v}_{global}(\boldsymbol{x}_t, t), \quad X_0 \sim \mathcal{N}(0, I)$$

In our proposed Stateful Flow Matching (SFM) architecture, historical context is continuously propagated through the hidden state $h_{i-1}(t)$. Consequently, the piecewise streaming ODE system is

Table 6: Subjective evaluations

Model	REL↑	OVL↑
AudioLDM2-48K	2.61	2.57
Stable-Audio-Open	2.82	3.28
TangoFlux	4.15	3.52
MFlowAudio	4.09	3.59

Table 7: Naturalness evaluations with dynamic prompting

Model	BAS↑	CPO↓	TN↑
TangoFlux (Dyna-P)	4.08	0.76	2.04
MFlowAudio (Dyna-P)	4.12	0.02	4.18

defined as:

$$\frac{d\boldsymbol{x}_t^{(i)}}{dt} = \boldsymbol{v}_\theta(\boldsymbol{x}_t^{(i)}, t, h_{i-1}(t)), \quad \text{where } h_i(t) = f(h_{i-1}(t), \boldsymbol{x}_t^{(i)})$$

where $t \in [0, 1]$ denotes the integration time mapping from the noise distribution to the data distribution.

Building upon the definition of the streaming ODE system, we can establish the explicit relationship between the global vector field and its piecewise counterpart.

Proposition A.2 (Vector Field Isomorphism)**.** *Assume the model architecture satisfies strict temporal causality, and the hidden state mapping $\mathcal{H}$ serves as a sufficient summary of the historical information, namely $h_{i-1}(t) \equiv \mathcal{H}(\boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(i-1)})$. With these conditions, the spatial concatenation of the streaming local vector fields is strictly isomorphic to the global causal vector field.*

Proof. Given that the global vector field $\vec{v}_{global}$ is strictly causal, its component corresponding to the i-th chunk depends exclusively on the current chunk and its preceding history:

$$\vec{v}_{global}^{(i)}(\boldsymbol{x}_t, t) = \hat{v}(\boldsymbol{x}_t^{(i)}, t, \boldsymbol{x}_t^{(1:i-1)})$$

In our proposed SFM architecture, the hidden state $h_{i-1}(t)$ recursively encodes the entire causal history $\boldsymbol{x}_t^{(1:i-1)}$. Consequently, there exists a parameterized local vector field $\boldsymbol{v}_\theta$ such that:

$$\boldsymbol{v}_\theta(\boldsymbol{x}_t^{(i)}, t, h_{i-1}(t)) \equiv \vec{v}_{global}^{(i)}(\boldsymbol{x}_t, t) \quad \forall i \in \{1, \ldots, K\}, \forall t \in [0, 1]$$

This implies that at any given integration timestep t, the spatial concatenation of the piecewise local dynamics driven by the state-passing mechanism is mathematically equivalent to the holistic global dynamics. □

Given this proposition, it can be deduced that, with the same initial noise, the integration trajectories of one-time and chunked generation are equivalent.

Theorem A.3 (Trajectory Equivalence)**.** *Given an identical initial Gaussian noise distribution $\boldsymbol{x}_0 \sim \mathcal{N}(0, I)$ and assuming constant semantic conditioning, the concatenated solution trajectory $\tilde{\boldsymbol{x}}_t = [\tilde{\boldsymbol{x}}_t^{(1)}, \ldots, \tilde{\boldsymbol{x}}_t^{(K)}]$ obtained via piecewise integration in SFM is strictly equivalent to the global trajectory $\boldsymbol{v}_t$ solved by the monolithic ODE system. That is, $\tilde{\boldsymbol{x}}_t = \boldsymbol{x}_t, \forall t \in [0, 1]$.*

Proof. According to the Picard-Lindelöf theorem from fundamental ODE theory (which establishes the existence and uniqueness of solutions to initial value problems):

Assuming the vector field $\vec{v}_{global}(\boldsymbol{x}, t)$ is locally Lipschitz continuous with respect to $\boldsymbol{x}$ across the state space, then for any given initial condition $\boldsymbol{x}_0$, there exists a unique global solution trajectory $\boldsymbol{x}_t$.

Now, consider the piecewise integration process of SFM. As established in Proposition A.2, each sub-ODE $\frac{d\boldsymbol{x}_t^{(i)}}{dt} = \boldsymbol{v}_\theta(\boldsymbol{x}_t^{(i)}, t, h_{i-1}(t))$ fundamentally constitutes an exact subsystem of the global ODE framework $\frac{d\boldsymbol{x}_t}{dt} = \vec{v}_{global}(\boldsymbol{x}_t, t)$.

Because the initial condition $\tilde{\boldsymbol{x}}_0 = \boldsymbol{x}_0$ is strictly identical and the driving vector fields are equivalent everywhere, the concatenated trajectory $\tilde{\boldsymbol{x}}_t$ derived from piecewise integration is necessarily a valid solution to the aforementioned global initial value problem.

Furthermore, since the Picard-Lindelöf theorem guarantees the absolute uniqueness of this global solution, it inevitably follows that:

$$\tilde{\boldsymbol{x}}_t \equiv \boldsymbol{x}_t, \quad \forall t \in [0, 1]$$

□

A.5 Implementation of State Caching and Chunked Generation

Per Step State Caching. Building upon the established Theorem A.3, we can conclude that as long as the underlying model maintains strict causality, caching the hidden state h_t is sufficient to achieve chunk-wise integration that is mathematically identical to monolithic global integration. However, because the numerical ODE solver in Flow Matching operates over multiple discrete timesteps, strict equivalence must be meticulously preserved at every single intermediate step.

To ensure this, our framework explicitly caches and propagates h_t at each discrete integration step along the trajectory. The total number of integration steps is typically defined by a fixed constant, commonly referred to as the Number of Function Evaluations (NFE). In our default generation configuration, the NFE is set to 25, which corresponds to a uniform integration step size of 0.04.

Causal Caching in TCMamba. Our overall framework effectively satisfies the causality prerequisite outlined in Proposition A.2 along the critical temporal dimension. Specifically, our model employs a dual-scan mechanism comprising two functionally distinct modules: a TimeMamba and a ChannelMamba. While the ChannelMamba scans orthogonally along the channel axis and does not adhere to temporal causal constraints, the TimeMamba strictly enforces unidirectional causality along the time axis. Since the theoretical requirement for seamless chunk-wise generation relies exclusively on temporal continuity, we streamline our state-passing mechanism by only caching and propagating the hidden states of the TimeMamba, bypassing the ChannelMamba.

The cache holds the per-layer TimeMamba SSM states across the NFE integration steps, and the two CFG branches, i.e., its size is $(2 \times \text{NFE} \times \text{per-layer states})$, a fixed constant in absolute size. It does not grow with audio duration (unlike a Transformer KV-cache, which grows $\mathcal{O}(L)$).

The state passed between chunks is the per-block Mamba2 SSM state, shape $(B, 24, 64, 128)$, it has no length dimension. A matched-parameter Transformer backbone (self-attention, temporal width 800) has no such compressed state and must retain every chunk's keys/values.

Table 8: Cache size comparison between MFlowAudio and Transformer-based diffusion models

L **(dur.)**	**MFlowAudio SSM state / fwd**	**Transformer KV cache / fwd**
128 (6 s)	**18.0 MB**	18.8 MB
256 (12 s)	**18.0 MB**	37.5 MB
512 (24 s)	**18.0 MB**	75.0 MB
scaling	**constant,** $\mathcal{O}(1)$	$4\times$, $\mathcal{O}(L)$

As shown in Table 8, the SSM cache is identical at 6 s, 12 s and 24 s (18 MB fp16) and adds zero extra FLOPs; the full streaming-generation cache is $18\,\text{MB} \times \text{NFE} \times 2\,(\text{CFG}) \approx 900$ MB, still independent of duration. The Transformer KV cache instead grows linearly ($4\times$ over $6 \to 24$ s), and being bidirectional it cannot stream causally at all.

Implementation of Streaming Generation. For the backbone, the cached SSM state is a sufficient statistic only for the SSM recurrence. Mamba2's causal conv1d retains an additional finite state (its preceding frames) that the cache omits. Because this second assumption is only approximately met, we obtain approximate equivalence, which we mitigate with a 3-frame warmup and quantify as a bounded 2–6% residual.

As per the VAE, since the Stable Audio Open Oobleck decoder is a convolutional autoencoder (2048× upsampling, 21.5 Hz latent) with a receptive field of ≈ 8 latent frames each side. We measure the

boundary error of decoding chunk-by-chunk with ‘W‘ latent frames of context per side, versus the full one-pass decode (reference):

Table 9: Lookahead Analysis in Stable Audio Open Oobleck VAE

overlap W	**lookahead**	**rel-L1 vs full**	**peak error**
0 (naive per-chunk)	0 ms	4.54%	0.62
4	182 ms	0.39%	0.15
8	**364 ms**	**0.049%**	**0.003**
16	727 ms	0.047%	0.002

As shown in Table 9, at $W \geq 8$ the streamed decode reproduces the full decode to the fp16 floor (peak 0.003, inaudible), because a convolution's output within its receptive field is identical once the full context is supplied; below that, naive per-chunk decode produces audible clicks at every seam (peak 0.62).

Therefore, our practical implementation introduces a W=8 lookahead buffer to accommodate this non-causality without breaking the overall streaming pipeline.

About the equivalence. The only operator that mixes information across time in TCMamba is TimeMamba, whose SSM hidden state is a sufficient summary of the past by construction of the linear recurrence. The other components are pointwise in time and hence invariant to chunking: cross-attention attends only to the (fixed) text tokens, and FiLM/FFN act per frame; ChannelMamba operates within local windows of length sub_l with no cross-window state, so it is unaffected as long as chunk boundaries align with window boundaries; and CFG is a linear combination of two such equivalent passes.

However, exact chunk-wise equivalence requires several implementation-level alignments: window alignment, causal conv1d warmup at chunk boundaries, and consistent global time positions. We acknowledge that these cannot make the chunked and one-time paths bit-exact in practice; our implementation handles every one of these alignment points as carefully as possible, which yields an empirically very small discrepancy. We therefore only achieve ”approximate equivalence with empirically negligible discrepancy”.

Table 10: Chunked generation v.s. one-time generation

Model	**Chunk-Size**	**FD**$\downarrow$	**KL**$\downarrow$	**IS**$\uparrow$	**FAD**$_v\downarrow$	**CLAP**$\uparrow$	**FAD**$_{openl3}\downarrow$
MFlowAudio	-	11.8569	1.1158	11.6865	2.5248	0.4957	96.2363
TangoFlux-chunked	$\sim$64	21.0276	1.6026	9.6641	3.3030	0.3446	109.1011
MFlowAudio	32	12.0262	1.1277	11.3561	2.7999	0.4808	98.0833
MFlowAudio	64	11.5988	1.1545	11.8024	2.5961	0.4900	93.6472
MFlowAudio	96	11.3551	1.1447	11.6086	2.5946	0.4890	94.2503
MFlowAudio	128	11.2576	1.1262	11.5822	2.4817	0.4910	93.0274

We conducted a quantitative analysis to validate the soundness of chunked generation. As shown in Table 10, the performance discrepancy between chunked generation and monolithic one-time generation is negligible.

A.6 Out of Domain Generalization

To assess generalization, we evaluate on the Clotho evaluation set. This is a genuine out-of-domain test for our model: Clotho is sourced from FreeSound, which we do not train on, so there is no domain overlap or leakage from our side.

As shown in Table 11, MFlowAudio generalizes well to this different-domain distribution, outperforms other baselines.

Table 11: Performance comparison across baselines on Clotho testset.

Duration	Model	Parameters	FD↓	KL↓	IS↑	FAD_v↓	CLAP↑	FAD_{openl3}↓
10s	AudioLDM2-48K	1.7B	28.4461	2.4003	4.5949	5.5341	0.4886	191.7932
	Stable Audio Open	1B	29.7914	2.3376	8.4862	2.3854	0.4895	86.9946
	TangoFlux-base	516M	20.9078	2.210	8.7841	2.8864	0.4796	85.5309
	MFlowAudio-base	581M	**16.7074**	**2.1671**	8.7764	**2.1386**	0.5031	**73.8142**
	TangoFlux	516M	21.0051	2.2747	**9.0996**	2.4150	0.5014	78.1923
	MFlowAudio	581M	19.6883	2.2069	8.7865	3.1331	**0.5058**	82.2531

A.7 Quantitative Evaluation of Transition Smoothness

Audio is inherently a time-continuous signal; consequently, temporal discontinuities are highly perceptible to the human auditory system. Naive concatenation of two independently generated audio chunks inevitably introduces severe acoustic artifacts, such as:

1) Phase Discontinuities (Phase Jumps): The temporal boundary of two independently sampled waveforms rarely aligns precisely at a zero-crossing. An abrupt, non-differentiable amplitude jump at the splicing point physically induces a sudden broadband energy burst. Perceptually, this translates into highly obtrusive ”click” or ”pop” artifacts.

2) Noise Floor Shifts: Discrepancies in ambient background noise or acoustic environments between adjacent chunks (e.g., transitioning from a chunk with a subtle hiss to a purely clean one) result in a sudden drop-off or unnatural introduction of noise. This severely breaks acoustic continuity, creating a jarring sensation akin to power being abruptly cut off.

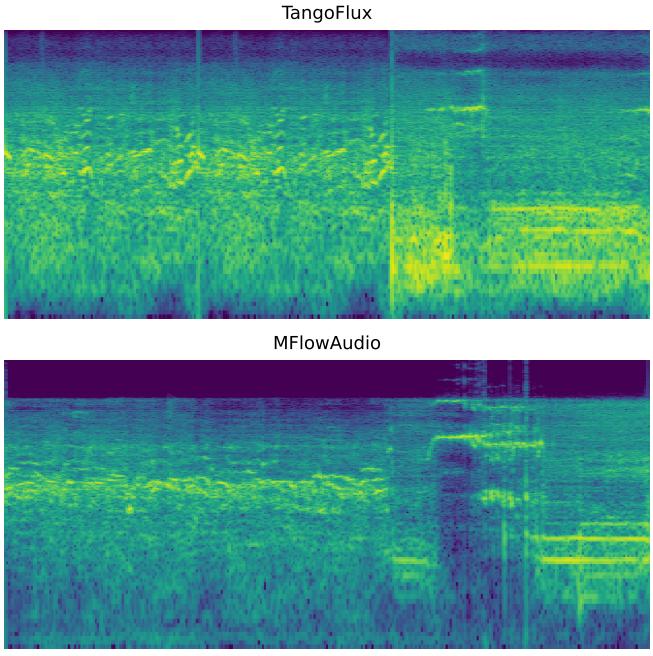

Figure 5: Spectrum generated by TangoFlux (top) and MFlowAudio (bottom) via dynamic prompting. Prompts: ”0, people are talking” ”3, a crowd of people talking” ”6, a horse neighing as a lawnmower engine runs in the background”. There is a noticeable spectral jump in TangoFlux at the third and sixth seconds (a bright vertical line).

As illustrated in Figure 5, the top spectrogram presents pronounced bright vertical striations at the temporal boundaries. These striations indicate severe structural fragmentation and abrupt spectral discontinuities across the transition.

To quantify this unnaturalness during dynamic prompt switching, we formulate three targeted evaluation metrics based on the concept of spectral flux. Collectively defining this framework as Boundary Spectral Flux, the formulation is detailed as follows:

Definition A.4 (Boundary Spectral Flux Metrics)**.** Let $\mathrm{SF}^{(i)}[m] = \sum_{\omega} \max\big(X^{(i)}_{m+1}[\omega] - X^{(i)}_{m}[\omega], 0\big)$ be the half-wave-rectified spectral flux at frame m of sample i (STFT with Hann window, N=2048, hop H=512), and $\tau_m = (m+1)H/\mathrm{sr}$. Given prompt-switch times $\{t^{(i)}_k\}_{k=1}^{K_i}$ and window half-width Δ, the three reported quantities are defined as follows:

$$
\begin{aligned}
\mathrm{BMaxSF} &= \tfrac{1}{N}\sum_{i=1}^{N} \tfrac{1}{K_i}\sum_{k=1}^{K_i} \max_{m:\,|\tau_m - t^{(i)}_k| \le \Delta} \mathrm{SF}^{(i)}[m], && \text{(boundary peak)},\\
\mathrm{BgMedSF} &= \tfrac{1}{N}\sum_{i=1}^{N} \mathrm{median}\big\{\mathrm{SF}^{(i)}[m] : m \in \big\{m :\, |\tau_m - t^{(i)}_k| > \Delta,\ \forall k\big\}\big\}. && \text{(background frames)}\\
\mathrm{Ratio} &= \mathrm{BMaxSF} \,/\, \mathrm{BgMedSF}
\end{aligned}
\tag{14}
$$

A.8 Parameter Analysis of State Decoupled Conditional Guidance

As articulated in Proposition A.2, guaranteeing acoustic continuity in chunk-wise generation inherently necessitates strict consistency of the vector field at each integration step. However, when the semantic condition (i.e., the text prompt) transitions dynamically between adjacent chunks, this fundamental continuity is inevitably disrupted. The cached state exerts historical inertia, forcefully constraining the generation trajectory to converge toward a subset of the manifold that is acoustically coherent with the preceding chunk, while the newly injected prompt attempts to redirect the trajectory's endpoint toward a different semantic target space. Consequently, vector field collision precipitates at every single step of the ODE integration process.

To circumvent this antagonistic interference and successfully steer the trajectory toward a highly specific, unified manifold that optimally satisfies both temporal continuity and the newly introduced semantic constraints, we employ SDCG to dynamically calibrate and orthogonalize the target vector field at each integration step. Since the historical state might go against the guidance of the new prompt in scenarios of semantic mismatch, we introduce a scaling factor, δ (as formulated in Equation 11), to explicitly modulate the flux of the historical state.

Table 12: Performance varies with δ

δ	**FD**↓	**KL**↓	**IS**↑	**FAD**$_v$↓	**CLAP**↑	**FAD**$_{openl3}$↓	**BMaxSF**	**BgMedSF**	**BBR**↓
1.0	16.0462	1.2453	7.6155	2.9257	0.5071	102.2526	852.1209	199.0192	4.2816
0.9	16.0299	1.2422	7.6105	2.9177	0.5072	102.1618	852.9687	199.1520	4.2830
0.8	16.0364	1.2387	7.6059	2.9077	0.5074	102.0876	854.3832	199.2957	4.2870
0.5	15.9700	1.2310	7.6002	2.8672	0.5078	101.7815	857.0271	199.6675	4.2923
0.1	15.8575	1.2190	7.5717	2.8448	0.5087	101.2825	865.8578	200.0253	4.3287

The empirical parameter analysis is presented in Table 12. Given that our evaluation prompts were not explicitly engineered for sequential coherence, inherent semantic conflicts across boundaries are pronounced. Consequently, decreasing δ (thereby attenuating the state's inductive bias) significantly enhances semantic alignment and distributional fidelity. However, this relaxation entails a corresponding decline in the Boundary-to-Background Ratio (BBR). This degradation confirms that overly penalizing the state flux compromises temporal continuity, precipitating unnatural acoustic transitions. To achieve an optimal trade-off between semantic responsiveness and boundary smoothness, thoughtfully engineered prompt trajectories are necessary.

A.9 The Efficacy of ChannelMamba

To evaluate the contribution of the ChannelMamba module, we conducted an ablation study by removing it entirely from the TCMamba architecture (denoted as the No-Channel baseline). Notably, this specific module is lightweight, introducing a marginal parameter overhead of merely 2M (e.g., increasing the total parameter count from 581M to 579M).

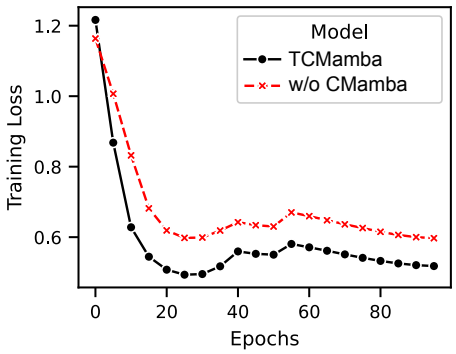

Figure 6: The loss curve during training.

Despite its minimal parameter footprint, its impact on model performance is significant. As illustrated by the training loss trajectories in Figure 6, the No-Channel configuration exhibits substantially slower convergence and ultimately plateaus at a higher loss floor compared to the full model equipped with ChannelMamba.

Table 13: Ablation for ChannelMamba

Duration	**Model**	**Parameters↓**	**CLAP↑**	**IS↑**	**FAD↓**	**FD↓**	**KL↓**
10s	w/o ChannelMamba	579M	0.3758	10.0166	6.9879	18.1077	1.1983
	MFlowAudio	581M	0.4655	11.6865	2.5248	11.8569	1.1158
30s	w/o ChannelMamba	579M	0.2596	5.5623	9.0628	45.6336	1.3096
	MFlowAudio	581M	0.4448	11.3356	2.8981	17.5387	1.1411

Empirical evaluations on the test set further corroborate this observation. As shown in Table 13, the No-Channel model suffers a severe degradation in both semantic alignment and acoustic generation fidelity. Crucially, this performance gap widens drastically during sequence length extrapolation (e.g., 30 seconds of audio), revealing that spectral modeling is indispensable for maintaining long-term structural coherence. In sum, these findings underscore that the lightweight ChannelMamba module yields massive gains in both training stability and generative capacity.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main contributions and scope of the paper are clearly stated in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the conclusion section, we explicitly discuss the limitations of the paper.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the complete assumptions and proofs for the theoretical analysis in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide sufficient details to enable reproducibility of the experiments and commit to releasing the checkpoints and resulting data in the future.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: All datasets used are publicly available. We include the core pseudo-codes in the paper and provide a demo page to showcase a subset of the generated results. In addition, we commit to releasing the code and checkpoints in the future.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: In the appendix, we present detailed information on the datasets, hyperparameters, optimizers, and other settings used in both the training and generation processes.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper reports the variance across multiple repeated experimental runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details of the GPU platforms used for all experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that all aspects of the paper comply with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper presents a standard machine learning study and does not involve societal impact considerations.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper is a standard machine learning study based on publicly available datasets and does not involve additional safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: This work uses publicly available datasets and models, and all of them are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All released data, code, and models will be properly documented upon open-sourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The appendix includes screenshots of the interface used to guide the subjective evaluation.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The evaluation experiments in this paper do not require IRB.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: This paper does not involve the use of LLMs in any part of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.