

Live Avatar: Streaming Real-time Audio-Driven Avatar Generation with Infinite Length

Yubo Huang^{1,2}, Hailong Guo^{2,3}, Fangtai Wu^{2,4}, Weiqiang Wang⁵, Shijie Huang², Qijun Gan⁴, Shifeng Zhang², Lin Liu¹, Sirui Zhao^{1*}, Enhong Chen^{1*}, Jiaming Liu^{2*†}, and Steven Hoi²

¹ University of Science and Technology of China, China

² Qwen Applications Business Group of Alibaba, China

³ Beijing University of Posts and Telecommunications, China

⁴ Zhejiang University, China

⁵ Monash University, Australia

siruit@ustc.edu.cn, cheneh@ustc.edu.cn, jmliu1217@gmail.com

*Corresponding authors. †Project leader.

Abstract. Audio-driven avatar interaction demands real-time, streaming, and infinite-length generation—capabilities fundamentally at odds with the sequential denoising and long-horizon drift of current diffusion models. We present Live Avatar, an algorithm-system co-designed framework that addresses both challenges for a 14-billion-parameter diffusion model. **On the algorithm side**, a two-stage pipeline distills a pre-trained bidirectional model into a causal, few-step streaming one whose KV cache stores *noisy* rather than clean history. This lossy conditioning, counter-intuitively, suppresses error accumulation and enables models trained on short clips to generalize stably beyond 10,000 seconds. **On the system side**, this *noisy* conditioning also breaks the sequential sampling bottleneck, allowing Timestep-forcing Pipeline Parallelism (TPP) to assign each GPU a fixed denoising timestep, converting the sequential diffusion chain into an asynchronous spatial pipeline that simultaneously boosts throughput and improves temporal consistency. Live Avatar achieves 45 FPS with a TTFF of 1.21 s on 5 H100 GPUs, and to our knowledge is the first to enable practical real-time streaming of a 14B diffusion model for infinite-length avatar generation. Our project page is at <https://liveavatar.github.io/>.

Keywords: Interactive Video Generation · Audio-driven Avatar Generation · Long Video Generation

1 Introduction

Audio-driven avatar generation, the synthesis of photorealistic human face video whose motion is driven by an input audio stream, is a foundational technology for interactive digital communication. Its applications are expansive, ranging from virtual reality and live streaming to digital assistants. The demand for systems capable of producing high-fidelity, expressive, and real-time avatars has driven

Table 1: Comparison of state-of-the-art audio-driven avatar generation methods. Live Avatar simultaneously achieves streaming, real-time, and infinite-length generation with a large-scale (14B) diffusion model.

Method	streaming	real-time	infinite-length	scale
Hallo3 [6]	✗	✗	✗	5B
StableAvatar [40]	✗	✗	✓	1.3B
Wan-S2V [13]	✗	✗	✗	14B
Ditto [23]	✓	✓	✓	0.2B
InfiniteTalk [47]	✗	✗	✓	14B
OmniAvatar [12]	✗	✗	✗	14B
Live Avatar (Ours)	✓	✓	✓	14B

significant recent advancements, particularly with the rise of diffusion models for high-fidelity video synthesis [2, 16, 42]. Despite their success in setting new benchmarks for visual quality, deploying these powerful generative models in real-time⁶ (≥ 24 FPS), streaming environments over unbounded durations faces fundamental and conflicting challenges.

The first challenge is long-horizon consistency. Existing methods suffer from compounding errors that cause identity drift and color artifacts within minutes [40]. Recent progress such as Self-Forcing [18] mitigates the train–test gap but still degrades rapidly under minute-level rollout; LongLive [48] sustains longer generation via native long-duration training but is designed for text-to-video and too costly to scale to 14B models.

The second challenge is the real-time–fidelity trade-off. Large-scale diffusion models [13] yield superior visual quality but are inherently slow due to sequential multi-step denoising, while existing real-time approaches [23] sidestep this sequential bottleneck entirely by adopting non-iterative generation methods or very small models, at the cost of fidelity. Reconciling these two desiderata at scale remains an open problem.

To address these critical challenges, we propose **Live Avatar**, an algorithm–system co-designed framework that enables large diffusion models (up to 14 billion parameters) for real-time, streaming, and infinite-length audio-driven avatar generation without compromising visual fidelity.

For infinite-length stability, we leverage the static-scene prior of avatar interaction to anchor identity in a persistent sink frame and store only *noisy* representations in the KV cache—the noise acts as a low-pass filter that suppresses accumulated artifacts while preserving motion dynamics, and keeps the latent distribution compact to prevent out-of-distribution drift. Additional mechanisms further address distribution drift and positional extrapolation, together enabling stable generation beyond 10,000 seconds.

⁶ Throughout this paper, “real-time” refers to generation throughput exceeding playback speed, not end-to-end conversational latency.

For real-time streaming, we distill the model into a few-step causal one via Self-Forcing and propose Timestep-forcing Pipeline Parallelism (TPP), which exploits the *noisy* KV cache to pipeline denoising steps across GPUs, breaking the sequential sampling bottleneck. Combined with system-level optimizations, this achieves high throughput and low latency at 14B scale.

The core contributions of Live Avatar are as follows:

- **Causal, Streamable Adaptation Framework.** We propose a two-stage framework that adapts a pretrained bidirectional diffusion model into a causal, few-step streaming model, with a novel motion-frame-as-scaffold mechanism that bridges Stage 1 and Stage 2 by providing functionally analogous training signals, yielding a $5\times$ distillation convergence speedup.
- **Long-Horizon Stability.** We show that storing only *noisy* history in the KV cache enables train-short-infer-long: training on ~ 3 s clips yet extrapolating stably beyond 10,000 s. Additional mechanisms further address distribution drift and positional extrapolation.
- **Timestep-forcing Pipeline Parallelism (TPP).** We propose TPP, which exploits the *noisy* KV cache to assign each GPU a fixed denoising timestep, breaking the sequential sampling bottleneck and boosting throughput. Combined with system-level optimizations, TPP achieves 45 FPS with a TTFF of 1.21 s on $5\times$ H100 GPUs, to our knowledge the first practical real-time streaming of a 14B diffusion model.
- **Standardized Benchmark.** We introduce GenBench (Short/Long), including long-video test cases exceeding five minutes.

2 Related Work

Streaming and Long Video Generation. Streaming and long video generation require efficient management of both computation and memory resources [7, 20, 25, 51]. Diffusion Forcing [3] introduces varied noise levels to sequential targets, enabling flexible-length streaming generation. Self-Forcing [18] addresses train-inference mismatch by conditioning on previously generated frames, yet still suffers from exposure bias and fails to produce minute-level long videos. LongLive [48] applies sliding window distillation for streaming long video generation, but is natively designed for text-to-video and unsuitable for image-conditioned tasks, and its training inefficiency prevents scaling to large-scale models such as 14B-parameter architectures. While these approaches improve video quality and temporal consistency, none achieve real-time, streaming long video generation with large-scale diffusion models.

Audio-driven Avatar Video Generation. Audio-driven avatar video generation requires subject consistency and effective motion control. Early works [34, 53] leverage GANs and 3D motion models for lip-syncing and facial animation. With the success of diffusion models, several studies [39, 52] adapt diffusion frameworks and ReferenceNet architectures for avatar generation. DiT-based video generation models [2, 16, 21, 42] have demonstrated remarkable visual quality, inspiring

a surge of DiT-based avatar methods [8, 15, 23, 26, 31, 43, 54, 56]. Meanwhile, autoregressive approaches [1, 4, 22, 46] offer an alternative by combining autoregressive and diffusion strategies. However, none achieve real-time, streaming, infinite-length avatar generation with large-scale diffusion models.

Diffusion Distillation. Diffusion model distillation accelerates video generation through various paradigms, including trajectory distillation [11], Consistency Models [27, 36, 55], and Distribution Matching Distillation (DMD) [29, 30, 50]. Among these, DMD has proven particularly effective for streaming video generation: CausVid [51], Self-Forcing [7, 18], and LongLive [48] all employ DMD-based few-step distillation on temporally segmented long videos, achieving both significant sampling speedup and improved long-video quality. Recent analysis further suggests that DMD functions analogously to reinforcement learning, with the pretrained diffusion model serving as a reward signal [28], potentially explaining its effectiveness in streaming settings.

3 Preliminaries

3.1 Video Diffusion Models

Video diffusion models generate high-fidelity video sequences by progressively denoising from a Gaussian prior $x_T \sim N(0, I)$, following the reverse process of a forward diffusion. In this work, we adopt the flow matching [24], where noisy latents at time t are constructed as

$$x_t = (1 - s_t) \cdot x_0 + s_t \cdot x_T, \quad (1)$$

where $s_t \in [0, 1]$ is a scheduling function that controls the interpolation between clean and noise. The model is trained to predict the target velocity

$$v = x_T - x_0 \quad (2)$$

leading to the standard mean squared error objective:

$$\mathcal{L} = \mathbb{E}_{x_0, x_T, t} [\|v_\theta(x_t, t, c) - (x_T - x_0)\|_2^2] \quad (3)$$

where c denotes conditional inputs such as text embeddings.

To reduce computational cost, most modern video diffusion models operate in a compressed latent space. We use a causal 3D VAE to encode input videos into temporal-latent representations, ensuring that future frames do not leak during training. Text conditioning is achieved through a pre-trained language model that produces contextual embeddings fed into the diffusion backbone.

3.2 Distribution Matching Distillation

Distribution Matching Distillation (DMD) [50] aims to distill a pre-trained teacher diffusion model into a student model that operates with fewer sampling steps. Let $p_{\theta, t}(\mathbf{x}_t)$ denote the distribution induced by the few-step student model

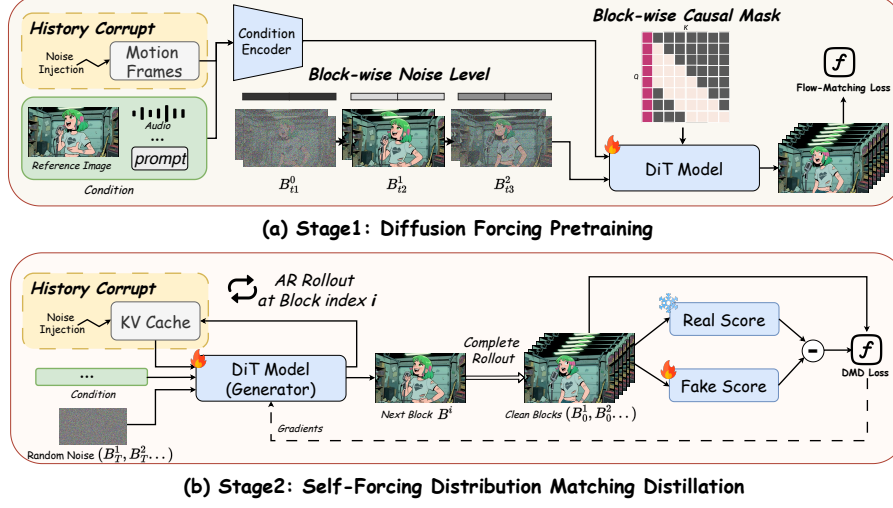


Fig. 1: The Live Avatar Training Framework. (a) Stage 1 Diffusion Forcing Pretraining with motion-frame scaffolding: noisy motion frames provide auxiliary temporal context alongside block-wise independent noise and causal attention masks. (b) Stage 2 Self-Forcing Distillation with History Corrupt: the motion frames are removed and replaced by the rolling KV cache, where noise level consistency between the KV cache and the noisy latents is enforced.

$\mathbf{x} = G_\theta(\mathbf{z})$, and let $p_{\text{data},t}(\mathbf{x}_t)$ represent the corresponding ground-truth distribution produced by the teacher diffusion model at time step t . The primary objective of DMD is to minimize the distribution, i.e., reverse Kullback-Leibler (KL) divergence, between these two distributions at each time step t : $\mathbb{E}_t [D_{\text{KL}}(p_{\theta,t} \parallel p_{\text{data},t})]$. The gradient of DMD loss is given by:

$$\nabla_\theta \mathcal{L}_{\text{DMD}} = -\mathbb{E}_{t,\mathbf{z}} \left[(s_{\text{real}}(\mathbf{x}_t, t) - s_{\text{fake},\phi}(\mathbf{x}_t, t))^\top \frac{\partial G_\theta(\mathbf{z})}{\partial \theta} \right] \quad (4)$$

where $\mathbf{x}_t = \Psi(\hat{\mathbf{x}}, t)$ is the noise scheduler, $\hat{\mathbf{x}} = G_\theta(\mathbf{z})$ is the data prediction of the distilled model, s_{real} and $s_{\text{fake},\phi}$ denote the score functions corresponding to the pre-trained teacher diffusion model and the student generator, respectively.

s_{real} , $s_{\text{fake},\phi}$, and $G_\theta(\mathbf{z})$ are all initialized from the pre-trained teacher model induced by v_θ . The training proceeds by alternately updating $s_{\text{fake},\phi}$ and $G_\theta(\mathbf{z})$, in which $s_{\text{fake},\phi}$ is trained on samples generated by the current student generator $G_\theta(\mathbf{z})$, and $G_\theta(\mathbf{z})$ is trained using the DMD loss defined above. For multi-step distillation, DMD first performs a multi-step sampling trajectory using the student generator: $\mathbf{z} \xrightarrow{G_\theta} \hat{\mathbf{x}}_{t_1} \xrightarrow{\Psi} \mathbf{x}_{t_1} \xrightarrow{G_\theta} \hat{\mathbf{x}}_{t_2} \xrightarrow{\Psi} \mathbf{x}_{t_2} \rightarrow \dots \rightarrow \mathbf{x}_{t_N}$. Then, at each training iteration, a random intermediate state $\mathbf{x}_{t_i}$ from this trajectory is selected and used in place of pure noise $\mathbf{z}$ as the starting point for the DMD training procedure [49].

4 The Live Avatar Framework

In this section, we present the technical details of Live Avatar. We first detail the model architecture in Sec. 4.1, followed by the overall training framework in Sec. 4.2 and Fig. 1. We then investigate long video generation in Sec. 4.3, proposing three strategies to address visual quality degradation, identity drift, and color artifacts in long-term generation. Finally, the inference framework and Timestep-forcing Pipeline Parallelism are demonstrated in Sec. 4.4.

4.1 Model Architecture

In order to enable streaming video generation, the Live Avatar adopts autoregressive generation by factorizing the joint distribution

$$B_{t-1}^i = D_\theta(B_t^i, \underbrace{B_t^{(i-w):(i-1)}}_{\text{kv cache}}, I, a^i, t^i) \quad (5)$$

combining diffusion-based frame synthesis with causal dependencies across chunks. B in Eq. 5 are blocks of consecutive noisy frame latents. In our work, we set the number of frame latents to 3. w is the rolling KV-cache window size. I denotes the sink frame, which provides the appearance; a^i and t^i are the audio embedding and prompt embedding for the i -th block respectively. The underscore t denotes the denoising step, and the superscript i denotes the block index. Note that in the model, the kv cache $B_t^{(i-w):(i-1)}$ and the noisy block B_t^i share the same noise level, this is a crucial design that improves generation quality and maximize the inference speed, which will be illustrated in Sec. 4.4.

4.2 Model Training

Our overall training framework is illustrated in Figure 1, which consists of two stages: 1) Stage 1, **Diffusion Forcing Pretraining** with motion-frame scaffolding, and 2) Stage 2, **Self-Forcing Distribution Matching Distillation** with History Corrupt.

Stage 1: Diffusion Forcing Pretraining. Following prior practice [3, 51], we apply block-wise independent noise scheduling and causal attention masks—frames within a block share full attention while inter-block attention is strictly causal—and train the model with the standard flow-matching loss. We additionally introduce motion frames [6] as auxiliary context, whose role is detailed below.

Motion-Frame as Scaffold. Employing a naive two-stage training converges slowly on long-horizon capabilities, yet each self-forcing rollout step is far more expensive than supervised training. Inspired by [14], we repurpose the motion frames—context frames preceding the current clip, originally for clip continuation—as scaffolding in Stage 1 by injecting noise into them. Since both noisy motion frames and the Stage 2 noisy KV cache serve a functionally analogous role as temporal context for continuing generation, this cheaply teaches dynamics-identity

decoupling without costly self-forcing rollouts. The motion frames are then entirely replaced by the KV cache in Stage 2, yielding a $5\times$ convergence speedup. **Stage 2: Self-Forcing Distillation.** In Stage 2 we distill the bidirectional teacher into a causal, few-step streaming model following Self-Forcing [18]. The causal student denoises one block at a time while conditioning on a rolling KV cache of previously generated blocks. Crucially, we omit the extra clean-cache refresh forward pass used in prior work [18], so that the KV cache always contains *noisy* representations—a strategy we call **History Corrupt**, whose motivation is detailed in Sec. 4.3. During the denoising of each block, the model attends to the KV cache from previous blocks at the *same* timestep; the noise-level implications of this design are also discussed in Sec. 4.3.

4.3 Long Video Generation

Existing talking-avatar systems exhibit pronounced degradation over long, autoregressive generation—manifesting as identity drift, color shifts, and temporal instability [40]. In practice, we perform inference in a rolling KV cache fashion [18], which extends the temporal horizon but, on its own, does not prevent collapse. We attribute these long-horizon failures to three internal phenomena:

- (i) **Test-time conditioning drift.** The conditioning pattern at test time (e.g., the RoPE-relative positioning between the sink frame and current target blocks) gradually diverges from the training-time setup, weakening identity cues.
- (ii) **Distribution drift.** The distribution of generated frames progressively deviates from realistic video distributions, likely driven by persistent factors (e.g., a real-data sink frame whose distribution subtly differs from the model’s generation manifold) that continuously push the rolling generation toward unrealistic outputs.
- (iii) **Error accumulation.** Subtle imperfections in each generated block are inherited and compounded frame-by-frame through the clean KV cache, as the model attends to fine-grained details—including artifacts—from previous blocks. This compounding causes rapid quality deterioration over time.

To address these challenges, we propose three complementary strategies—History Corrupt, Adaptive Attention Sink, and Rolling RoPE—that together enable stable infinite-length generation.

History Corrupt. For avatar interaction, the subject typically resides in a relatively static scene: apart from facial expressions, body gestures, and mild background dynamics, the visual content does not undergo rapid change. This prior allows a simplifying design assumption—the sink frame can provide sufficiently useful appearance and identity information for *every* generated frame throughout the entire sequence. Under this assumption, the role of the rolling KV cache reduces to conveying *motion dynamics* alone, while fine-grained identity and appearance details should be sourced exclusively from the persistent sink frame.

This motivates a simple yet effective design: we store only *noisy* representations in the KV cache, rather than the conventional clean cache. Intuitively,

Gaussian noise acts as a low-pass filter [35] on the cached representations, suppressing high-frequency details (including accumulated artifacts) while preserving low-frequency motion structure. This forces the model to extract dynamic cues from the noisy history while relying on the clean sink frame for identity and appearance, achieving an effective *dynamics–identity decoupling*. As a result, generation errors in previous blocks are no longer faithfully propagated, directly addressing error accumulation (iii).

Furthermore, noisy KV caches also mitigate distribution drift (ii). At higher noise levels, the marginal distribution $p_t(\mathbf{x}_t)$ converges toward the Gaussian prior and occupies a progressively more compact region of the latent space [37]. Consequently, the noisy cache is far less likely to drift into out-of-distribution regions compared to a clean cache, which must remain on the narrow data manifold where small perturbations can compound into distributional shift.

We further observe empirically that enforcing the noisy block and its attended KV cache to share the *same* noise level—a strategy we call **timestep-forcing**—significantly reduces inter-frame flickering, consistent with observations in TalkingMachines [26]. This timestep-forcing constraint also naturally lends itself to efficient multi-GPU inference, as discussed in Sec. 4.4.

Adaptive Attention Sink (AAS)⁷. By default, the user-provided reference image (i.e., the image-to-video conditioning input) serves as the sink frame. However, this real-data image resides on a slightly different distribution from the model’s own generation manifold, introducing a persistent bias that accumulates into color, exposure, or style deviations over long runs. To counteract this form of distribution drift (ii), AAS replaces the sink frame with the model’s own first generated latent immediately after the first block is produced, and uses it as the persistent sink for all subsequent conditioning. By keeping the sink frame within the model’s learned generation manifold, AAS eliminates the distributional mismatch between conditioning and generated content.

Rolling RoPE⁷. To mitigate test-time conditioning drift (i), we introduce a dynamic position-alignment mechanism for the sink frame. The sink frame is permanently cached in KV memory and its temporal offset is adjusted via a controllable RoPE shift so that its relative position to the current noisy states remains consistent with training. This dynamic RoPE alignment lets the model continuously reference identity features from the sink frame without rigidly constraining local motion, thereby stabilizing long-range identity and structural fidelity.

4.4 Timestep-forcing Pipeline Parallelism

Deploying large video generation models in real-time settings remains challenging due to the inherent sequential structure of diffusion-based sampling. Our distilled 14B model, for example, reaches only 5 FPS on a single GPU and 6 FPS under conventional 4-GPU sequential parallelization. In contrast, existing real-time streaming methods, such as CausVid [51] and LongLive [48], achieve

⁷ Details of AAS and Rolling RoPE are provided in the supplementary materials.

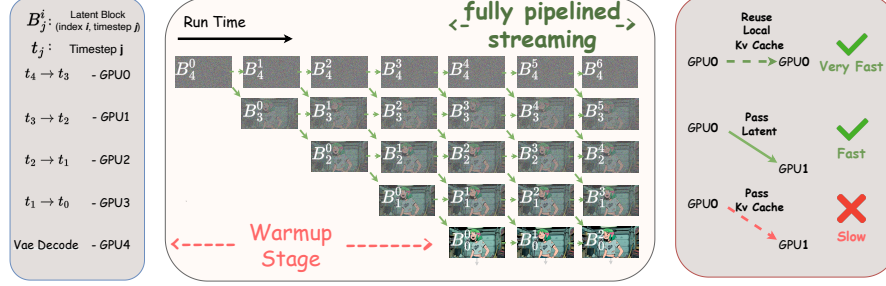


Fig. 2: A visual illustration of Timestep-forcing Pipeline Parallelism (TPP). After warm-up fills the pipeline, all GPUs denoise **simultaneously** in the fully pipelined stage, turning the sequential diffusion chain into an asynchronous spatial pipeline. For example, GPU2 always performs the $t_2 \rightarrow t_1$ step: it reuses its local KV cache (**very fast**) and sends only the latent to GPU3 (**fast**).

higher frame rates by substantially reducing model capacity—often relying on lightweight 1.3B models and aggressive quantization—at the cost of generation fidelity. This establishes a long-standing tradeoff between model quality and real-time performance that has yet to be resolved.

To overcome the sequential bottleneck of diffusion sampling, we introduce **Timestep-forcing Pipeline Parallelism (TPP)** (illustrated in Figure 2), which assigns each GPU a fixed timestep t_i and partitions the T denoising steps across T devices. Each GPU repeatedly performs its designated transformation $t_i \rightarrow t_{i-1}$, converting the sequential diffusion chain into an asynchronous spatial pipeline. Through this reparameterization, throughput is determined by a single denoise forward rather than the sum of all diffusion steps, yielding an ideal speedup proportional to the total number of denoising steps. Importantly, TPP is both *model-agnostic* and *hardware-agnostic*: it applies to any causal diffusion model—not only distilled ones—and only requires that each device can execute a single denoising step with minimal inter-device bandwidth, since TPP communicates only the compact latent between stages.

TPP operates in two stages. During warm-up, the first block is propagated through all timesteps to fill the pipeline, which completes quickly given the small number of sampling steps. Once filled, the system enters the fully pipelined streaming phase: each GPU repeatedly performs its assigned denoising step, passes the latent to the next device, and immediately processes the next block, achieving maximal parallel throughput. Since TPP assigns one timestep per GPU, the timestep-forcing strategy (Sec. 4.3) maps directly onto the pipeline: each GPU_i maintains its own local rolling KV cache, requiring no inter-GPU communication and thus remaining extremely fast and efficient. After finishing its step, GPU_i passes only the latent to GPU_{i+1} , keeping the communication cost negligible. To prevent pipeline bottlenecks, the VAE decoding stage is offloaded to an additional dedicated GPU, which consumes the clean latent and outputs synchronized video chunks.

5 Experiments

5.1 Experimental Settings

Implementation Details. The overall model architecture is borrowed from WanS2V [13]. In Stage 1, we initialize from its weights and implement motion frames with the frozen FramePack encoder from WanS2V; each motion frame is independently noised via the flow-matching forward process [24] with $t \sim \mathcal{U}[0, 1000]$. In Stage 2, the motion-frame encoder is removed, the teacher score and fake score branches are initialized from WanS2V, and the student is initialized from Stage 1. All training and inference are performed at a fixed resolution of 720×400 and 84 frames. Experiments are conducted on 128 NVIDIA H100 GPUs, with 25 K steps for Stage 1 and 500 steps for Stage 2. The per-GPU batch size is 1. To handle the high memory demand of Self-Forcing training, we adopt FSDP with gradient accumulation to reduce memory consumption. The learning rate is $1e-5$ for the student and $2e-6$ for the fake score. We group every 3 latents into a block, set the KV-cache window size $w=4$ blocks (Eq. 5), and use a single sink frame. We train models with a LoRA, whose rank and alpha are set to 128 and 64, respectively. At inference time, we apply a series of kernel-level optimizations (collectively referred to as Kernel Opt. in Table 3): FP8 quantization, FlashAttention-3, cuDNN fused kernels, torch.compile, LoRA weight merging, and VAE feature caching which caches intermediate features to enable streaming decoding. All reported metrics reflect this optimized configuration.

Datasets. We train on AVSpeech [10], a large-scale audio-visual dataset of talking-head clips. We adopt the preprocessing of OmniAvatar [12] and keep only clips longer than 10s, yielding 400K training samples. To evaluate our model’s out-of-domain (OOD) generalization, we created a synthetic benchmark named GenBench. This test set was generated using Gemini-2.5 Pro, Qwen-Image [44], and CosyVoice [9]. It is composed of two subsets: GenBench-ShortVideo, comprising 100 test samples with an approximate duration of 10 seconds, and GenBench-LongVideo, which contains 15 test videos, each exceeding 5 minutes in duration. The benchmark is designed to be challenging, featuring a wide diversity of character styles (photorealistic humans, animated characters, and anthropomorphic non-humans) and visual compositions, including frontal and profile views, as well as half-body and full-body shots. This variety allows for a robust assessment of the model’s performance on unseen data.

Evaluation Metrics. We employ Q-Align [45] to evaluate perceptual quality (IQA) and aesthetic appeal (ASE). Audio-visual synchronization is measured via Sync-C and Sync-D [5]. Identity consistency is assessed by DINOv2 [33] cosine similarity (Dino-S) between generated frames and the reference image. For the in-domain AVSpeech evaluation (reported in the supplementary), we additionally include FID [17] and FVD [41] since ground-truth videos are available.

5.2 Comparison with Existing Methods

We compare Live Avatar against current state-of-the-art open-sourced audio-driven avatar generation approaches, including Ditto [23], Echomimic-V2 [32],

Table 2: Quantitative comparisons of our methods with state-of-the-art methods.

Dataset	Model	Metrics					
		ASE $\uparrow$	IQA $\uparrow$	Sync-C $\uparrow$	Sync-D $\downarrow$	Dino-S $\uparrow$	FPS $\uparrow$
GenBench-Short Video	Ditto [23]	3.31	4.24	4.09	10.76	0.99	21.80
	Echomimic-V2 [32]	2.82	3.61	5.57	9.13	0.79	0.53
	Hallo3 [6]	3.12	3.97	4.74	10.19	0.94	0.26
	StableAvatar [40]	3.52	4.47	3.42	11.33	0.93	0.64
	OmniAvatar [12]	3.53	4.49	6.77	8.22	0.95	0.16
	WanS2V [13]	3.36	4.29	5.89	9.08	0.95	0.25
	Ours	3.44	4.51	7.03	8.30	0.96	45.2
GenBench-Long Video	Ditto [23]	2.90	4.48	3.98	10.57	0.97	21.80
	Hallo3 [6]	2.65	4.04	6.18	9.29	0.83	0.26
	StableAvatar [40]	3.00	4.66	1.97	13.57	0.94	0.64
	OmniAvatar [12]	2.36	2.86	8.00	7.59	0.66	0.16
	WanS2V [13]	2.63	3.99	6.04	9.12	0.80	0.25
	Ours	3.42	4.76	7.16	8.31	0.97	45.2

Hallo3 [6], Stable-Avatar [40], OmniAvatar [12], and WanS2V [13]. All methods are benchmarked at 720×400 on a single H100 node, with sequence-parallel methods using their maximum supported parallelism and all other settings kept at official defaults. Quantitative results on GenBench are presented in Table 2.

GenBench-Short Video. On short-form evaluation, our method achieves the best IQA and competitive ASE with OmniAvatar and Stable-Avatar, outperforming the rest. Although we build upon WanS2V and use step distillation, we surpass the teacher on visual quality; this aligns with the tendency of DMD-distilled models to concentrate the output distribution and attain slightly higher perceptual scores [38], and the DMD loss has been interpreted as a specialized reinforcement learning objective that effectively optimizes both aesthetic appeal and foundational visual fidelity [28, 30]. On Dino-S we obtain 0.96; Ditto’s 0.99 is expected as it is a face-editing model that retains the reference face more than a full video-generation pipeline. Audio-visual synchronization is comparable to OmniAvatar (Sync-C and Sync-D are both strong for both methods). The most pronounced gap is inference speed: we reach 45.2 FPS. Ditto is the only other method with near real-time throughput (21.8 FPS); we use roughly $70\times$ its parameters yet reach 45.2 FPS, which underscores the effectiveness of DMD, TPP, and our kernel-level optimizations.

GenBench-Long Video. Long-duration generation reveals a clear separation among methods. Echomimic-V2 is excluded as it does not support long-video inference. Comparing Short to Long results in Table 2, most generative baselines degrade noticeably: OmniAvatar drops from 3.53/4.49 to 2.36/2.86 (ASE/IQA), WanS2V from 3.36/4.29 to 2.63/3.99, and Hallo3 from 3.12/3.97 to 2.65/4.04. Ditto, as a face-editing model, maintains relatively stable quality but lags in audio-visual synchronization (Sync-C 3.98). Our method, by contrast, preserves near-identical quality across durations (ASE $3.44 \rightarrow 3.42$, IQA $4.51 \rightarrow 4.76$, Dino-

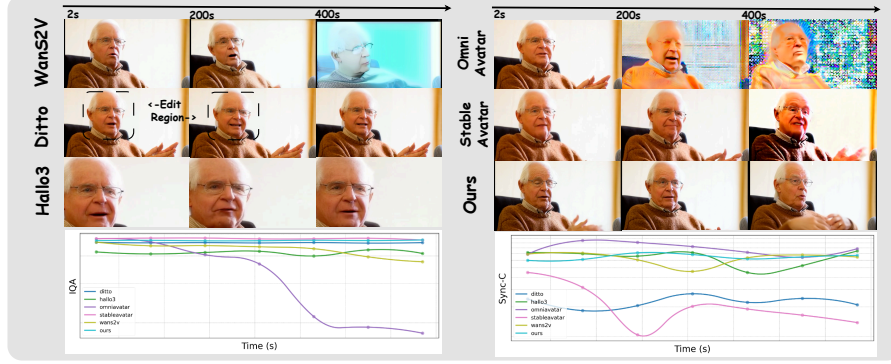


Fig. 3: Qualitative comparisons with state-of-the-art methods.

S 0.96 $\rightarrow$ 0.97). OmniAvatar achieves the highest Sync-C/Sync-D on Long, but its severe visual collapse (Dino-S 0.66) suggests that loss of identity constraints artificially inflates sync scores. We achieve competitive synchronization (Sync-C 7.16) while maintaining the best overall visual and identity quality, yielding a more balanced profile across all metrics. Figure 3 corroborates this: most base-lines, especially OmniAvatar and WanS2V, exhibit visible quality degradation over time, whereas ours maintains stable fidelity throughout.

Table 3: Ablation on Inference Efficiency. Table 4: Ablation on Long Video Generation (GenBench-LongVideo). SP_4 : sequence parallelism across 4 GPUs.

DMD	SP_4	TPP	VAE	Para.	Kern.	Opt.	#GPU	NFE	FPS $\uparrow$	TTFF $\downarrow$	Methods	Metrics			
												ASE $\uparrow$	IQA $\uparrow$	Sync-C $\uparrow$	Dino-S $\uparrow$
$\times$	$\times$	$\times$	$\times$	$\times$	$\times$		1	80	0.29	45.50	w/o AAS	3.13	4.68	7.25	0.96
$\checkmark$	$\times$	$\times$	$\times$	$\times$	$\times$		1	5	3.66	4.56	w/o Rolling RoPE	3.38	4.82	7.29	0.86
$\checkmark$	$\checkmark$	$\times$	$\times$	$\times$	$\times$		4	5	4.50	3.94	w/o History Corrupt	2.90	3.88	7.14	0.81
$\checkmark$	$\times$	$\checkmark$	$\times$	$\times$	$\times$		4	4	10.16	4.73	Ours	3.42	4.76	7.16	0.97
$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\times$	$\times$		5	4	20.88	2.89					
$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$		5	4	45.2	1.21					

5.3 Ablation Study

Study of inference efficiency. Table 3 incrementally enables each component. DMD removes CFG and cuts NFE from 80 to 5 (4 denoising steps + 1 clean-cache pass needed by Self-Forcing [51]). SP_4 provides only marginal speedup since per-block sequences are short (3 latents). TPP more than doubles throughput by pipelining steps across GPUs and eliminating the clean-cache pass (NFE 5 $\rightarrow$ 4). VAE parallelization and kernel-level optimizations yield further gains, bringing the full system to 45.2 FPS / 1.21 s TTFF on 5 H100 GPUs.

Study on long-horizon stability strategies. As shown in Table 4 and Figure 4, each component targets a distinct failure mode. Removing History Corrupt causes the most severe degradation (ASE/IQA drop to 2.90/3.88, Dino-S

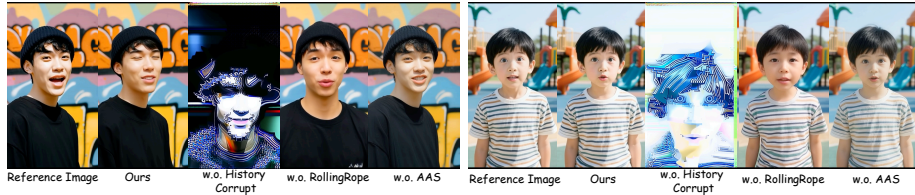


Fig. 4: Visual ablation of long-horizon stability components on long video generation.

to 0.81). Without AAS, progressive color desaturation emerges, with noticeably grayed-out frames (ASE drops to 3.13; Figure 4). Removing Rolling RoPE triggers identity drift (Dino-S $0.97 \rightarrow 0.86$), with visible changes in hair details and facial features (Figure 4).

Study on denoising step count. As shown in Table 5, increasing steps from 2 to 4 brings modest visual gains but markedly improves audio-visual synchronization (Sync-C $6.41 \rightarrow 7.03$), since fewer steps leave insufficient budget for early, motion-critical denoising stages. Thanks to TPP, FPS stays at 45.2 regardless of step count; only TTFF grows ($0.68\text{ s} \rightarrow 1.21\text{ s}$), which remains practical for interactive streaming where end-to-end latency is dominated by network transport. We therefore default to 4 steps.

Table 5: Effect of denoising step count on GenBench-ShortVideo. TPP keeps FPS constant regardless of step count; TTFF scales with the number of steps.

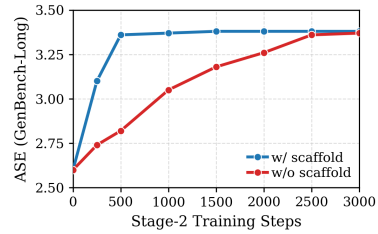
Method	ASE $\uparrow$	IQA $\uparrow$	Sync-C $\uparrow$	Sync-D $\downarrow$	Dino-S $\uparrow$	TTFF (s) $\downarrow$	FPS $\uparrow$
Ours (2 steps)	3.37	4.25	6.41	9.02	0.94	0.68	45.2
Ours (3 steps)	3.41	4.36	6.58	8.85	0.95	0.95	45.2
Ours (4 steps)	3.44	4.51	7.03	8.30	0.96	1.21	45.2

Study on noise level of KV cache. Table 6 compares three KV-cache noise schedules on GenBench-Long: clean-KV-cache, fix-noisy-KV-cache (noise fixed at $t=557$), and our timestep-forcing. Both baselines follow standard Self-Forcing inference [51]. We additionally report T.Flicker [19], which measures temporal consistency. Clean-KV-cache degrades rapidly over long horizons due to error accumulation. Fixed noise improves quality by decoupling identity and motion dynamics in the cache [35], but both baselines suffer from severe flickering (T.Flicker 0.876/0.891). Only timestep-forcing resolves this (T.Flicker 0.971) by maintaining step-matched caches. Furthermore, timestep-forcing is the only schedule compatible with TPP.

Table 6: Effect of KV-cache noise-level scheduling on GenBench-Long.

Setting	Metrics					
	ASE $\uparrow$	IQA $\uparrow$	Sync-C $\uparrow$	Sync-D $\downarrow$	Dino-S $\uparrow$	T.Flicker $\uparrow$
clean-kv-cache	3.05	4.44	6.11	9.10	0.90	0.876
fix-noisy-kv-cache	3.34	4.63	7.00	8.38	0.93	0.891
timestep-forcing	3.42	4.76	7.16	8.31	0.97	0.971

Study on motion-frame scaffolding. Starting from the same 25 K-step Stage 1 with all other hyperparameters aligned, we compare Stage 2 convergence with and without motion-frame scaffolding (Figure 5). The scaffolded variant saturates at ~ 500 steps, while the baseline requires ~ 2500 steps—a $5\times$ speedup with no loss in final quality.

**Fig. 5:** Study of motion-frame scaffolding.

6 Conclusion

We present **Live Avatar**, an algorithm-system co-designed framework that, to our knowledge, is the first to enable practical real-time streaming of a 14-billion-parameter diffusion model for infinite-length audio-driven avatar generation. On the algorithm side, a two-stage training pipeline adapts a pretrained bidirectional model into a causal, few-step streaming one. Three complementary long-horizon strategies—History Corrupt, Adaptive Attention Sink, and Rolling RoPE—enable stable generation beyond 10,000 seconds⁸. On the system side, Timestep-forcing Pipeline Parallelism (TPP) assigns each GPU a fixed denoising timestep, converting the sequential diffusion chain into an asynchronous spatial pipeline that achieves 45 FPS with a TTFF of 1.21 s on 5 H100 GPUs. We also introduce GenBench, a long-form benchmark exceeding five minutes, to support reproducible evaluation.

Limitation. Our long-horizon stability strategies are grounded in the static-scene prior inherent to avatar interaction, where the subject and background remain largely unchanged; as native long-duration training becomes feasible, such priors may become unnecessary. Additionally, the current TTFF of 1.21 s, combined with network transport overhead, yields an end-to-end latency of approximately 3 s, which does not yet meet the stringent requirements of seamless bidirectional interaction.

⁸ We provide experimental results on 10,000-second generation in the supplementary material.

Acknowledgements

This work was supported in part by the grants from National Natural Science Foundation of China (No. 62406264, 62222213, 62072423).

References

1. Ao, T.: Body of her: A preliminary study on end-to-end humanoid agent. arXiv preprint arXiv:2408.02879 (2024)
2. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024), <https://openai.com/research/video-generation-models-as-world-simulators>
3. Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems* **37**, 24081–24125 (2024)
4. Chen, M., Cui, L., Zhang, W., Zhang, H., Zhou, Y., Li, X., Tang, S., Liu, J., Liao, B., Chen, H., et al.: Midas: Multimodal interactive digital-human synthesis via real-time autoregressive video generation. arXiv preprint arXiv:2508.19320 (2025)
5. Chung, J.S., Zisserman, A.: Out of time: automated lip sync in the wild. In: *Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20–24, 2016, Revised Selected Papers, Part II* 13. pp. 251–263. Springer (2017)
6. Cui, J., Li, H., Zhan, Y., Shang, H., Cheng, K., Ma, Y., Mu, S., Zhou, H., Wang, J., Zhu, S.: Hallo3: Highly dynamic and realistic portrait image animation with video diffusion transformer. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 21086–21095 (2025)
7. Cui, J., Wu, J., Li, M., Yang, T., Li, X., Wang, R., Bai, A., Ban, Y., Hsieh, C.J.: Self-forcing++: Towards minute-scale high-quality video generation. arXiv preprint arXiv:2510.02283 (2025)
8. Du, F., Li, T., Zhang, Z., Qiao, Q., Yu, T., Zhen, D., Jia, X., Yang, Y., Yin, S., Liu, S.: Rap: Real-time audio-driven portrait animation with video diffusion transformer. arXiv preprint arXiv:2508.05115 (2025)
9. Du, Z., Wang, Y., Chen, Q., Shi, X., Lv, X., Zhao, T., Gao, Z., Yang, Y., Gao, C., Wang, H., Yu, F., Liu, H., Sheng, Z., Gu, Y., Deng, C., Wang, W., Zhang, S., Yan, Z., Zhou, J.: Cosyvoice 2: Scalable streaming speech synthesis with large language models (2024), <https://arxiv.org/abs/2412.10117>
10. Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., Freeman, W.T., Rubinstein, M.: Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. arXiv preprint arXiv:1804.03619 (2018)
11. Frans, K., Hafner, D., Levine, S., Abbeel, P.: One step diffusion via shortcut models. arXiv preprint arXiv:2410.12557 (2024)
12. Gan, Q., Yang, R., Zhu, J., Xue, S., Hoi, S.: Omniaavatar: Efficient audio-driven avatar video generation with adaptive body animation. arXiv preprint arXiv:2506.18866 (2025)
13. Gao, X., Hu, L., Hu, S., Huang, M., Ji, C., Meng, D., Qi, J., Qiao, P., Shen, Z., Song, Y., Sun, K., Tian, L., Wang, G., Wang, Q., Wang, Z., Xiao, J., Xu, S., Zhang, B., Zhang, P., Zhang, X., Zhang, Z., Zhou, J., Zhuo, L.: Wan-s2v: Audio-driven cinematic video generation (2025), <https://arxiv.org/abs/2508.18621>

14. Gelberg, Y., Eguchi, K., Akiba, T., Cetin, E.: Extending the context of pretrained llms by dropping their positional embeddings (2025), <https://arxiv.org/abs/2512.12167>
15. Guo, Y., Liu, X., Zhen, C., Yan, P., Wei, X.: Arig: Autoregressive interactive head generation for real-time conversations. arXiv preprint arXiv:2507.00472 (2025)
16. HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., Richardson, E., Levin, E., Shiran, G., Zabari, N., Gordon, O., et al.: Ltx-video: Realtime video latent diffusion. arXiv preprint arXiv:2501.00103 (2024)
17. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* **30** (2017)
18. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self forcing: Bridging the train-test gap in autoregressive video diffusion. arXiv preprint arXiv:2506.08009 (2025)
19. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., Wang, Y., Chen, X., Wang, L., Lin, D., Qiao, Y., Liu, Z.: Vbench: Comprehensive benchmark suite for video generative models (2023), <https://arxiv.org/abs/2311.17982>
20. Kodaira, A., Hou, T., Hou, J., Tomizuka, M., Zhao, Y.: Streamdit: Real-time streaming text-to-video generation. arXiv preprint arXiv:2507.03745 (2025)
21. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
22. Li, T., Tian, Y., Li, H., Deng, M., He, K.: Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems* **37**, 56424–56445 (2024)
23. Li, T., Zheng, R., Yang, M., Chen, J., Yang, M.: Ditto: Motion-space diffusion for controllable realtime talking head synthesis. arXiv preprint arXiv:2411.19509 (2024)
24. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
25. Liu, K., Hu, W., Xu, J., Shan, Y., Lu, S.: Rolling forcing: Autoregressive long video diffusion in real time. arXiv preprint arXiv:2509.25161 (2025)
26. Low, C., Wang, W.: Talkingmachines: Real-time audio-driven facetime-style video via autoregressive diffusion models. arXiv preprint arXiv:2506.03099 (2025)
27. Lu, C., Song, Y.: Simplifying, stabilizing and scaling continuous-time consistency models. arXiv preprint arXiv:2410.11081 (2024)
28. Luo, W.: Diff-instruct++: Training one-step text-to-image generator model to align with human preferences. arXiv preprint arXiv:2410.18881 (2024)
29. Luo, W., Hu, T., Zhang, S., Sun, J., Li, Z., Zhang, Z.: Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems* **36**, 76525–76546 (2023)
30. Luo, Y., Hu, T., Sun, J., Cai, Y., Tang, J.: Learning few-step diffusion models by trajectory distribution matching. arXiv preprint arXiv:2503.06674 (2025)
31. Meng, D., Xiao, S., Zhang, X., Wang, G., Zhang, P., Wang, Q., Zhang, B., Bo, L.: Mirrorme: Towards realtime and high fidelity audio-driven halfbody animation. arXiv preprint arXiv:2506.22065 (2025)
32. Meng, R., Zhang, X., Li, Y., Ma, C.: Echomimicv2: Towards striking, simplified, and semi-body human animation (2025), <https://arxiv.org/abs/2411.10061>

33. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jégou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research* (2024), <https://arxiv.org/abs/2304.07193>
34. Prajwal, K., Mukhopadhyay, R., Namboodiri, V.P., Jawahar, C.: A lip sync expert is all you need for speech to lip generation in the wild. In: *Proceedings of the 28th ACM international conference on multimedia*. pp. 484–492 (2020)
35. Song, K., Chen, B., Simchowitz, M., Du, Y., Tedrake, R., Sitzmann, V.: History-guided video diffusion (2025), <https://arxiv.org/abs/2502.06764>
36. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models (2023)
37. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. *CoRR* **abs/1907.05600** (2019), <http://arxiv.org/abs/1907.05600>
38. Team, I., Cai, H., Cao, S., Du, R., Gao, P., Hoi, S., Hou, Z., Huang, S., Jiang, D., Jin, X., Li, L., Li, Z., Li, Z.Y., Liu, D., Liu, D., Shi, J., Wu, Q., Yu, F., Zhang, C., Zhang, S., Zhou, S.: Z-image: An efficient image generation foundation model with single-stream diffusion transformer (2025), <https://arxiv.org/abs/2511.22699>
39. Tian, L., Wang, Q., Zhang, B., Bo, L.: Emo: Emote portrait alive generating expressive portrait videos with audio2video diffusion model under weak conditions. In: *European Conference on Computer Vision*. pp. 244–260. Springer (2024)
40. Tu, S., Pan, Y., Huang, Y., Han, X., Xing, Z., Dai, Q., Luo, C., Wu, Z., Jiang, Y.G.: Stableavatar: Infinite-length audio-driven avatar video generation. *arXiv preprint arXiv:2508.08248* (2025)
41. Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717* (2018)
42. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314* (2025)
43. Wang, Z., Zhang, P., Qi, J., Xu, G.W.S., Zhang, B., Bo, L.: Omnitalter: Real-time text-driven talking head generation with in-context audio-visual style replication. *arXiv e-prints* pp. arXiv–2504 (2025)
44. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-image technical report (2025), <https://arxiv.org/abs/2508.02324>
45. Wu, H., Zhang, Z., Zhang, W., Chen, C., Liao, L., Li, C., Gao, Y., Wang, A., Zhang, E., Sun, W., et al.: Q-align: Teaching llms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090* (2023)
46. Xie, Y., Gu, T., Li, Z., Zhang, C., Song, G., Zhao, X., Liang, C., Jiang, J., Xu, H., Luo, L.: X-streamer: Unified human world modeling with audiovisual interaction. *arXiv preprint arXiv:2509.21574* (2025)
47. Yang, S., Kong, Z., Gao, F., Cheng, M., Liu, X., Zhang, Y., Kang, Z., Luo, W., Cai, X., He, R., et al.: Infinitetalk: Audio-driven video generation for sparse-frame video dubbing. *arXiv preprint arXiv:2508.14033* (2025)
48. Yang, S., Huang, W., Chu, R., Xiao, Y., Zhao, Y., Wang, X., Li, M., Xie, E., Chen, Y., Lu, Y., et al.: Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622* (2025)

49. Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., Freeman, B.: Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems* **37**, 47455–47487 (2024)
50. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 6613–6623 (2024)
51. Yin, T., Zhang, Q., Zhang, R., Freeman, W.T., Durand, F., Shechtman, E., Huang, X.: From slow bidirectional to fast autoregressive video diffusion models. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 22963–22974 (2025)
52. Yu, H., Wang, Z., Pan, Y., Cheng, M., Yang, H., Wang, C., Xie, T., Xu, X., Wei, X., Cai, X.: Llia—enabling low-latency interactive avatars: Real-time audio-driven portrait video generation with diffusion models. *arXiv preprint arXiv:2506.05806* (2025)
53. Zhang, W., Cun, X., Wang, X., Zhang, Y., Shen, X., Guo, Y., Shan, Y., Wang, F.: Sadtalker: Learning realistic 3d motion coefficients for stylized audio-driven single image talking face animation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 8652–8661 (2023)
54. Zhen, D., Yin, S., Qin, S., Yi, H., Zhang, Z., Liu, S., Qi, G., Tao, M.: Teller: Real-time streaming audio-driven portrait animation with autoregressive motion generation. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 21075–21085 (2025)
55. Zheng, K., Wang, Y., Ma, Q., Chen, H., Zhang, J., Balaji, Y., Chen, J., Liu, M.Y., Zhu, J., Zhang, Q.: Large scale diffusion distillation via score-regularized continuous-time consistency. *arXiv preprint arXiv:2510.08431* (2025)
56. Zhu, Y., Zhang, L., Rong, Z., Hu, T., Liang, S., Ge, Z.: Infp: Audio-driven interactive head generation in dyadic conversations. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 10667–10677 (2025)