

Vidu S1: A Real-Time Interactive Video Generation Model

Jintao Zhang^{*†}, Kai Jiang^{*}, Jintao Chen^{*}, Xu Wang^{*}, Yang Luo^{*}, Yuji Wang^{*}, Dechuang Chen^{*}, Jungang Li^{*}, Chengyang Ye, Tianyu Chen, Hongzhou Zhu, Min Zhao, Yuxuan Jiang, Zhengkun Huang, Chendong Xiang, Kaiwen Zheng, Haoxu Wang, Xiaohang Wang, Qi Jia, Xin Chen, Yimin Chen, Youhe Jiang, Fangcheng Fu, Zhijie Deng[‡], Fan Bao[‡], Jianfei Chen[‡], Jun Zhu[‡]

Tsinghua University, Shengshu Technology

<https://vidu.com/vidu-stream>

Abstract

We introduce Vidu S1, a real-time interactive video generation model supporting voice control of digital characters. Users can control video generation content at any moment through voice instructions. Vidu S1 supports infinite-length real-time video generation without blurring, drift, or visual distortion. Built with TurboDiffusion and TurboServe, Vidu S1 outputs 540p real-time videos at up to 42 FPS on regular consumer GPUs. Users can upload custom images of real people, anime, and pets, and choose different voice tones for personalized experiences. Experiments show that Vidu S1 achieves the best performance across all test metrics while fully meeting real-time inference requirements. A playable online demo is available at <https://vidu.com/vidu-stream>.

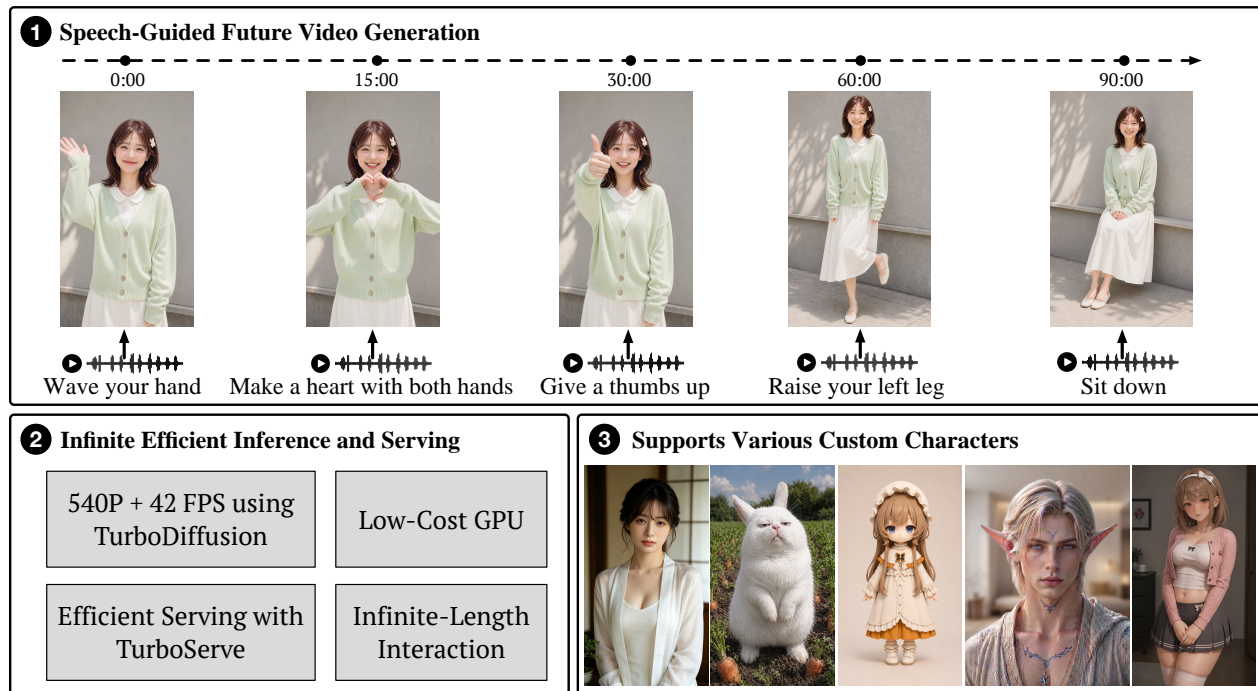


Figure 1 Overview of Vidu S1. Vidu S1 supports speech-guided future control, efficient infinite-duration inference, and diverse customized characters for real-time interactive video generation.

^{*}Core contributors, co-first authorship. [†]Project lead. [‡]Advisors.

1 Introduction

Background and Demand. Recent video generation models, such as Sora, Veo, Wan, and Seedance [1–4], have shown strong ability in generating high-quality videos. However, most of them still follow an offline, one-shot generation paradigm: a user enters a prompt, waits for minutes or even tens of minutes, and receives the complete video only after generation finishes. During this process, the user can only passively wait to receive information and cannot actively initiate any interaction. This limitation comes from the offline diffusion paradigm, where the model denoises the entire video synchronously over many steps and only produces an entire clean video at the end. Such a paradigm works well for offline content creation, but human visual entertainment is not limited to pre-generated videos. People also enjoy face-to-face communication, live streaming, games, talking with someone, and other interactive visual experiences, where content must respond immediately to the user. From a demand perspective, suppose that each user has an average demand of $\alpha \in [0, 1]$ for real-time interactive visual content, e.g., $\alpha = 0.5$. Then the total demand scales with $\alpha \times N$, where N is the number of users. In contrast, suppose that each user has an average demand of $\beta \in [0, 1]$ for offline-generated visual content, e.g., $\beta = 0.5$. Since offline-generated videos can be replayed and shared, their generation demand scales more like $\beta \times N/m$, where m is the average number of views per generated video. If we assume $\alpha \approx \beta$ and $m > 100$, then the demand for real-time interactive video generation is much greater than that for offline-generated videos.

Real-time interactive video generation is one of the most important future directions for industry.

Limitation. Despite the importance of this direction, existing autoregressive video generation models still have several key limitations. (1) Many methods [5–8] only change the generation process from offline generation to autoregressive generation, but they still do not support real-time user interaction during generation. (2) Most methods [9–12] do not use speech as a direct and explicit control signal for future video content, making it difficult for users to control the generation process through natural spoken instructions. (3) Long-horizon video generation remains challenging because small errors can accumulate over time, causing drift, instability, and eventually visual collapse [13, 14]. As a result, many models [9, 13–21] can only generate videos within a limited duration and cannot maintain stable generation over an open-ended stream. (4) Real-time video generation is not only a modeling problem. It also requires strong inference kernels, serving systems, scheduling strategies, and large-scale deployment infrastructure. Without such infrastructure, even a capable model may be too slow or too expensive for practical use.

Vidu S1. To address these limitations, we introduce Vidu S1, a real-time interactive video generation model designed for continuous user interaction. (1) Vidu S1 allows users to actively interact with the generation process at any moment, intervening in real time instead of merely specifying all controls before generation begins. (2) Vidu S1 treats user speech as a direct and explicit control signal, so spoken instructions can continuously guide what the video should generate next. (3) Vidu S1 supports stable long-horizon real-time generation by mitigating error accumulation during streaming generation, allowing the video to continue indefinitely without rapid drift or collapse. (4) It is also built together with an efficient inference and serving stack, including TurboDiffusion [22] and TurboServe [23], which enables low-cost real-time inference on commodity GPUs at 540p resolution and 42 FPS.

Contribution. We summarize our contributions as follows.

1. We introduce Vidu S1, a real-time interactive video generation model that supports continuous user interaction and enables explicit speech control over future video content during generation.
2. Vidu S1 supports indefinitely long video generation with high quality and consistency. By mitigating error accumulation during streaming, it allows video streams to run without drift or collapse.
3. Vidu S1 achieves real-time video generation at 540p resolution and 42 FPS. With an efficient inference and serving stack, Vidu S1 makes high-quality real-time generation practical on low-cost GPUs.
4. Experiments on Vidu-StreamBench and HDTF demonstrate that Vidu S1 combines leading quality and real-time interactivity, achieving the best CSIM (0.9192), Sync-D (7.847), and DOVER (0.5660), strong human preference while supporting speech-guided control and stable long-horizon streaming.

2 Method

2.1 Data Preparation

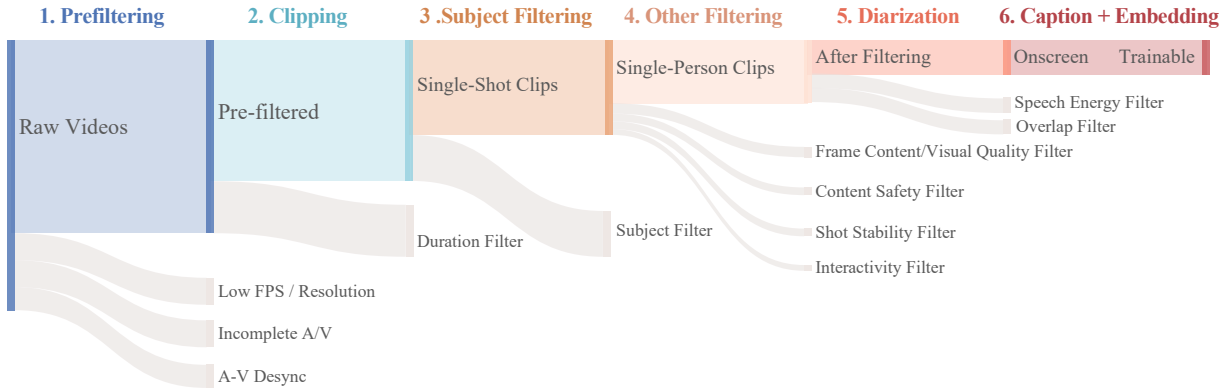


Figure 2 Data filtering pipeline for raw videos, showing progressive refinement through prefiltering, single-shot clipping, subject filtering, quality and safety checks, diarization, and caption generation. The final selected clips and their captions are embedded to form training data.

As illustrated in Figure 2, we build a progressive data processing pipeline that converts heterogeneous raw videos into high-quality, temporally coherent, and semantically annotated training clips. The pipeline first applies a prefiltering stage to discard technically unreliable videos, then segments the remaining data into single-shot clips, followed by subject-level filtering, quality and safety screening, speech diarization, and caption generation. This staged design jointly improves visual clarity, temporal stability, audio-visual consistency, and cross-modal interpretability, thereby yielding reliable training data for controllable character video generation. A detailed description of each stage is provided below.

Data Collection and Preprocessing. The quality of the training data substantially affects both the training performance and the model’s generalization capability. For this purpose, we collect and process a corpus of high-quality, diverse, and highly interactive single-person, single-shot video. The raw videos are drawn from two categories: livestream or talking-head videos, and high-quality footage from films and television dramas. The former is used mainly to learn fine-grained characteristics such as facial expressions, body movements, and lip synchronization, whereas the latter serves to improve the model’s generalization and consistency across diverse shot angles, scenes, and visual styles. Before formally entering the pipeline, all data are first deduplicated and then pre-filtered according to metrics such as frame rate, resolution, audio-visual integrity, and audio-visual synchronization.

Data Clipping and Filtering. Raw videos are segmented into single-shot clips along shot boundaries, ensuring that the model learns continuous and coherent visual content. Long shots are further segmented, with cut points constrained so as not to fall in the middle of speech, ultimately yielding single-shot clips ranging from 3 to 60 seconds in duration. We evaluate the data comprehensively across multiple perspectives and levels, and apply strict filtering rules for cleaning, in order to guarantee the quality of the training data. In practice, we found that using expert models only exhibits clear limitations when assessing video data. For instance, face-detection models struggle to generalize to exaggerated or highly stylized 2D-animated subjects; some expert models are image-based and can only make judgments on sampled frames, rendering them prone to erroneous assessments owing to insufficient global information. We therefore introduce an omni model [24, 25] to perform global semantic understanding over the complete video as a supplement. The omni model is used to generate semantic labels along multiple quality dimensions, including editing, subject, action, emotion, face, speech, scene, shot, and tone. The expert models and the Omni model together form a joint filtering system that possesses both global context awareness and local detail sensitivity. Specifically, we focus

on the following aspects: (1) *subject detection*, we ensure that each clip contains exactly one subject occupying a reasonable proportion of the frame; (2) *frame cleanliness*, we filter out clips that contain information irrelevant to the visual content, such as watermarks, subtitles, and overlaid advertisements; (3) *visual quality*, we use aesthetic and technical scoring to select clips that are clear, complete, and visually pleasing, while avoiding artifacts such as blur, jitter, and flicker; (4) *content safety*, we filter out NSFW and other inappropriate content so as to prevent the model from learning harmful information; (5) *shot stability*, we retain static shots or shots with slow motion in order to reduce the risk of shot drift during long-term generation; and (6) *interactivity*, we require the subject in the frame to exhibit clear actions or behaviors, thereby enabling the model to learn meaningful motion information.

Speech Diarization. High-fidelity lip synchronization can only be learned if the model observes visual performances that are consistent with the corresponding speech during training. To this end, we extract the speech component from the raw audio. We then apply voice activity detection and active speaker detection to annotate the timestamps of each speech segment together with its associated speaker. Based on the correspondence between the speaker and the on-screen subject, every segment is assigned to one of three categories: *onscreen* (the speaker matches the person shown in the video), *offscreen* (the speaker does not match any visible subject), and *overlap* (multiple voices coincide within the segment). Clips containing overlapping segments are directly filtered out. During processing, we observed that the diarization model behaves unstably in scenarios involving singing or strong background music, frequently misassigning vocals to the music stem or introducing noticeable artifacts such as synthetic timbre and distortion. To mitigate this, we further introduce a heuristic rule that discards segments in which the speaker is vocalizing, but the speech energy proportion is too low. In this way, we preserve the cleanliness and reliability of the speech signal.

Data Captioning. We annotate captions at two levels of granularity: full-clip captions and speech-aware chunk-level captions, which serve complementary roles. Clip-level captions provide a coherent global semantic anchor for the entire video, whereas speech-aware chunk-level captions align descriptions with their corresponding temporal intervals, offering fine-grained and temporally localized conditioning signals for controllable and interactive streaming generation. The annotation model produces structured natural-language descriptions covering a broad range of visual, auditory, and dialogue-related attributes, including subject appearance, actions, motion, emotion, scene context, camera language, cinematic properties, lighting, on-screen text, dialogue, sound effects, and background music. To improve annotation fidelity and reduce cross-modal hallucination, we adopt a dual-path strategy that decouples visual and auditory modalities. Visual attributes are inferred exclusively from video frames, while acoustic attributes are inferred exclusively from the audio track. Overall, the proposed structured annotation scheme improves the quality and consistency of multimodal representations while enabling more reliable annotation and more controllable generation.

2.2 Training

We aim to train a streaming video-audio generation model capable of producing temporally coherent video and synchronized audio under multimodal conditions. For clarity, we first introduce the notation used throughout the training framework.

Notation. Let v_0^i denote the clean video representation, and let a_0^i denote the clean audio representation associated with the i -th frame. To formulate video-audio joint generation, we define the clean joint state as:

$$\mathbf{x}_0^i = [v_0^i; a_0^i], \quad (1)$$

where $[\cdot; \cdot]$ denotes feature concatenation along the modality dimension. For a video-audio sequence of length N , we denote the corresponding clean joint-state sequence as $\mathbf{x}_0^{1:N} = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^N\}$.

The unified conditioning interface is denoted by \mathbf{c} , which includes speech, text prompts, and reference images. For a given sequence, we represent the conditioning sequence as $\mathbf{c}^{1:\infty} = \{\mathbf{c}^1, \mathbf{c}^2, \dots\}$, where \mathbf{c}^i denotes the conditioning representation associated with the i -th video-audio state. The subscript $t_j \in \{t_0, \dots, t_T\}$ denotes the diffusion timestep of a noisy state, while $t_0 = 0$ corresponds to the clean state. We use the hat notation

$\hat{\cdot}$ to denote model-predicted quantities. In particular, $\hat{\mathbf{x}}_0^i$ denotes the predicted clean estimate of the i -th video-audio state. We use f_θ^{bi} and f_θ to denote the bidirectional and causal denoising models, respectively.

Our training pipeline consists of three stages. In the first stage, we train a bidirectional video-audio generation model that serves as the foundation of our framework. In the second stage, we initialize the autoregressive model from the pretrained bidirectional model and further adapt it to a causal generation setting using a hybrid training strategy that unifies Teacher Forcing [26, 27] and Diffusion Forcing [5]. This stage equips the model with stable multi-step autoregressive generation capabilities. Finally, we employ Distribution Matching Distillation (DMD) [28] with Phased Consistency Models (PCM) [29] regularization to distill the autoregressive model into a few-step generator, significantly improving inference efficiency while maintaining generation quality.

Stage 1: Bidirectional Teacher Training. We first train a bidirectional teacher on full video-audio sequences. For a finite sequence of N frames, the bidirectional teacher is conditioned on the complete sequence $\mathbf{c}^{1:N}$ and is trained to denoise the corresponding joint latent states $\mathbf{x}_0^{1:N}$. Its input and output at timestep t_j can be written as:

$$\hat{\mathbf{x}}_0^{1:N} = f_\theta^{\text{bi}}(\mathbf{x}_{t_j}^{1:N}, t_j, \mathbf{c}^{1:N}), \quad (2)$$

Here $\mathbf{x}_{t_j}^{1:N}$ denotes the noised version of joint video-audio states 1 to N at diffusion timestep t_j , $\mathbf{c}^{1:N}$ denotes the complete condition sequence, and $\hat{\mathbf{x}}_0^{1:N}$ denotes the clean-state estimate predicted from this noisy state. The training objective is defined as:

$$\mathcal{L}_{\text{bi}} = \mathbb{E}_{(\mathbf{x}_0^{1:N}, \mathbf{c}^{1:N}), t_j} \left[\left\| f_\theta^{\text{bi}}(\mathbf{x}_{t_j}^{1:N}, t_j, \mathbf{c}^{1:N}) - \mathbf{x}_0^{1:N} \right\|_2^2 \right], \quad (3)$$

This stage establishes a high-quality generative prior for the subsequent causal adaptation and distillation.

Stage 2: Causal Teacher Training. We next initialize a causal teacher from the trained bidirectional teacher and adapt it to the streaming autoregressive setting. Specifically, we impose a causal attention mask on video-audio tokens, so that each target state can only access the causal condition context and the valid historical video-audio prefix.

For each frame $i > 1$, only the current joint state is denoised, conditioned on the available condition sequence $\mathbf{c}^{\leq i}$ and the historical video-audio prefix. The autoregressive denoising formulation can be written as:

$$\hat{\mathbf{x}}_0^i = f_\theta(\mathbf{x}_{t_j}^i, t_j, \mathbf{c}^{\leq i}, \mathbf{x}_{\tau_j}^{\leq i}, \tau_j), \quad (4)$$

where $\mathbf{x}_{t_j}^i$ is the noisy state of the current video-audio target at timestep t_j , $\mathbf{c}^{\leq i}$ denotes all conditions available up to frame i , $\mathbf{x}_{\tau_j}^{\leq i}$ denotes the historical video-audio prefix represented at noise level τ_j , and $\hat{\mathbf{x}}_0^i$ is the clean-state estimate predicted from this noisy state.

To reduce the gap between training and inference, we adopt a hybrid training strategy that combines Teacher Forcing and Diffusion Forcing. For each training sample, we draw a sample from $\text{Bernoulli}(p)$ to choose between Teacher Forcing and Diffusion Forcing.

Under Teacher Forcing, the historical prefix is given by clean ground-truth video-audio states $\mathbf{x}_0^{\leq i}$, which is equivalent to setting $\tau_j = 0$. This provides stable supervision for condition synchronization and motion consistency. Under Diffusion Forcing, the historical prefix is instantiated as noisy historical states $\mathbf{x}_{\tau_j}^{\leq i}$ with $\tau_j > 0$, improving the model’s robustness to imperfect prefixes during autoregressive rollout.

$$\mathcal{L}_{\text{causal}} = \mathbb{E}_{(\mathbf{x}_0^{1:N}, \mathbf{c}^{1:N}), i, t_j, \tau_j} \left[\left\| f_\theta(\mathbf{x}_{t_j}^i, t_j, \mathbf{c}^{\leq i}, \mathbf{x}_{\tau_j}^{\leq i}, \tau_j) - \mathbf{x}_0^i \right\|_2^2 \right]. \quad (5)$$

Through this adaptation, the causal teacher retains the generation quality of the bidirectional teacher while acquiring stable streaming generation capability.

Stage 3: DMD with PCM Regularization. Following the autoregressive warm-up stage, the model acquires a preliminary capability for multi-step autoregressive generation. However, generating videos still requires a large number of denoising steps for each autoregressive frame, resulting in substantial computational overhead and slow inference. To improve generation efficiency, we further apply DMD to compress the autoregressive generation process into only a few sampling steps. In our autoregressive setting, the causal generator predicts future states conditioned on the previously generated noisy prefix $\hat{\mathbf{x}}_{\tau_j}^{<i}$ and external conditions $\mathbf{c}^{1:\infty}$. By recursively applying the autoregressive generation process described in Equation. (4), we obtain a generated video-audio sequence $\hat{\mathbf{x}}_0^{1:N}$. DMD optimizes the generator by following the gradient that minimizes the discrepancy between the generated and data distributions. Following [28], the gradient of the DMD objective is given by

$$\nabla_{\theta} \mathcal{L}_{\text{DMD}} = \mathbb{E}_{z,t} \left[w(t) \left(\nabla_{\mathbf{x}_t^{1:N}} \log p_{\text{data}}^t(\mathbf{x}_t^{1:N}) - \nabla_{\mathbf{x}_t^{1:N}} \log p_{\theta}^t(\mathbf{x}_t^{1:N}) \right)^{\top} \frac{d\mathbf{x}_t^{1:N}}{d\theta} \right], \quad (6)$$

where $\nabla_{\mathbf{x}_t} \log p_{\text{data}}^t(\mathbf{x}_t)$ and $\nabla_{\mathbf{x}_t} \log p_{\theta}^t(\mathbf{x}_t)$ denote the score functions of the data and generated distributions, respectively, and $w(t)$ is the timestep-dependent weighting function.

Nevertheless, we observe that optimizing solely with the DMD objective frequently suffers from mode collapse, resulting in unstable generation behaviors such as camera drift, content degeneration, and temporal inconsistency. To alleviate these issues, we introduce the distillation objective in PCM [29] as an additional regularization mechanism during training. Specifically, we instantiate the distance function in the PCM objective using a perceptual feature distance, yielding

$$\mathcal{L}_{\text{PCM}} = \mathbb{E} \left[\lambda(t_n) d_{\text{PERC}} \left(f_{\theta}(\mathbf{x}_{t_{n+1}}^{1:N}, t_{n+1}), f_{\theta^-}(\hat{\mathbf{x}}_{t_n}^{1:N}, t_n) \right) \right], \quad (7)$$

where $d_{\text{PERC}}(\cdot, \cdot)$ denotes the perceptual feature distance, f_{θ^-} is the EMA student network, and $\lambda(t_n)$ is the timestep-dependent weighting function following PCM.

During training, we optimize the generator using a weighted combination of the DMD objective and the PCM consistency objective.

2.3 Inference

2.3.1 Streaming Inference

During inference, we employ sliding-window decoding to enable online autoregressive generation of arbitrarily long video-audio sequences under limited memory and computational resources. At the i -th autoregressive decoding step, attention is restricted to a fixed-length sliding window consisting of three components: (1) a persistent reference context, composed of the latent tokens extracted from the user-provided first frame together with the first generated video-audio state; (2) cached historical video-audio states retained within the sliding window; and (3) the current video-audio state being denoised.

The persistent reference context plays a role analogous to the sink token in large language models [30] and the sink frame adopted by recent streaming video generation methods [17, 31], providing stable global conditioning throughout the generation process. Restricting attention to this fixed-length window satisfies the causal autoregressive constraint while keeping the per-step inference latency constant regardless of the generated sequence length. To further improve the efficiency and temporal stability of streaming inference, we employ Rotary Position Embedding [32] (RoPE) Repositioning and TwinCache.

RoPE Repositioning To improve inference efficiency, we cache historical key-value features before applying RoPE [31, 33–36]. As the sliding window advances, RoPE is applied to the cached features according to their updated relative positions in the current window, avoiding redundant recomputation of historical features. This strategy preserves cross-window positional consistency while ensuring that all visible positions remain within the positional range encountered during training.

TwinCache We introduce **TwinCache**, a stage-aware caching strategy that maintains complementary noisy and clean caches for historical video-audio states. The persistent reference context, consisting of the latent tokens extracted from the user-provided first frame together with the first generated video-audio state, is constructed only once and remains fixed throughout inference. In contrast, each subsequently generated video-audio state maintains two cached representations: a noisy cache extracted from a predefined denoising step and a clean cache obtained after the final denoising step.

During denoising, the model attends to the persistent reference context together with the cached historical representations. During intermediate denoising steps, the historical context is represented by the noisy cache, i.e., $\hat{x}_{\tau_j}^{<i}$, whereas at the final denoising step it is replaced by the clean cache, i.e., $\hat{x}_0^{<i}$.

Unlike clean representations, the residual noise retained in $\hat{x}_{\tau_j}^{<i}$ preserves coarse temporal dynamics while suppressing the accumulation of high-frequency artifacts over long autoregressive sequences, acting as an implicit low-pass prior for temporal propagation [37, 38]. Consequently, the noisy history cache provides stable temporal guidance without imposing strong appearance constraints. At the final denoising step, the clean history cache $\hat{x}_0^{<i}$ is attended together with the persistent reference context to restore fine-grained visual details while maintaining temporal coherence and identity consistency [39]. By decoupling temporal propagation from appearance refinement via stage-aware cache scheduling, TwinCache effectively balances long-term temporal consistency and visual fidelity in streaming video-audio generation.

2.3.2 Inference Infrastructure

Building on the inference configuration above, we deploy a hardware–software co-designed acceleration stack aligned with the technical route of TurboDiffusion [22], incorporating its key worker-side techniques, with TurboServe [23] as a reference for cluster-level streaming serving. By combining the above with our hardware–software co-design, we save per-step compute and memory in diffusion inference while preserving generation quality, enabling real-time inference on multiple GPUs. The main components are as follows.

Attention Acceleration. Attention is a major contributor to both latency and computational cost in the diffusion model [40]. We accelerate attention with SageAttention [41–45], SpargeAttention [46, 47], and Sparse-Linear Attention (SLA) [48, 49]. They significantly reduce attention latency while maintaining high generation quality.

Linear Layer Quantization. The choice of quantization granularity is critical to accuracy. We find that per-tensor and per-channel quantization are both vulnerable to outliers. A few extreme values stretch the quantization range, and the remaining values are therefore quantized with poor precision, which leads to noticeable quality loss. To address this, we implement custom per-block W8A8 quantized GEMM operators in CUDA. These operators reduce the inference-time memory footprint and significantly accelerate linear-layer computation, while maintaining high precision and generation quality.

Kernel Fusion. We fuse sequences of operators into custom Triton/CUDA kernels. Merging fine-grained kernels cuts host launch and synchronization overhead and avoids spilling intermediate tensors to global memory, reducing memory bandwidth pressure. Representative fusions include RMSNorm combined with selected elementwise operations.

CUDA Graph. Streaming video generation naturally exhibits two characteristics: individual operators have short computation times, and the inference process follows a relatively stable, repetitive execution pattern. Under these conditions, host-side kernel launch overhead accumulates and can become a bottleneck, leaving the GPU underutilized. We therefore adopt CUDA Graph: a fixed-structure subgraph is captured once into a CUDA graph and then replayed one or more times at each inference step. Each replay merges many independent kernel launches into a single graph launch, reducing redundant launch overhead and improving GPU utilization.

Table 1 Quantitative comparison of different audio-driven avatar generation methods. Best results are highlighted in **bold**. Resolution and FPS/throughput are reported when publicly specified or measured in our test environment; otherwise marked as “-”.

| Model | Instruction Following | Real-Time | Resolution | FPS / Throughput | CSIM \uparrow | Sync-D \downarrow | DOVER \uparrow |
|-------------------|-----------------------|-----------|-----------------------|------------------|-----------------|---------------------|------------------|
| OmniAvatar | × | × | 480p | - | 0.8062 | 9.242 | 0.5476 |
| StableAvatar-1.3B | × | × | 480p | - | 0.8358 | 11.18 | 0.5560 |
| Hallo3 | × | × | 480×720 | - | 0.7698 | 8.660 | 0.5313 |
| Wan2.2-S2V-14B | × | × | 480p/720p | - | 0.7936 | 8.255 | 0.5510 |
| LiveAvatar | × | ✓ | - | - | 0.8127 | 8.447 | 0.5639 |
| LemonSlice | × | ✓ | 368×560 | - | 0.8407 | 7.921 | 0.5196 |
| HeyGen | × | ✓ | - | 25 FPS | 0.9191 | 8.037 | 0.4864 |
| Kling Avatar 2.0 | ✓ | × | - | - | 0.8688 | 8.158 | 0.5406 |
| Vidu S1 | ✓ | ✓ | 540p (960×540) | 42 FPS | 0.9192 | 7.8470 | 0.5660 |

Multi-GPU Parallelism. To meet real-time latency targets within the available compute and memory budgets, we adopt Ulysses-style context parallelism [50] to distribute computation across multiple GPUs. Each GPU processes a slice of the sequence during the forward pass. Before and after attention, all-to-all collectives re-partition activations between sequence-parallel and head-parallel layouts, distributing both compute and activation memory across GPUs. We further optimize the underlying communication implementation to minimize its overhead.

3 Evaluations

3.1 Setup

Standard public benchmarks such as HDTF [51] evaluate audio-driven avatar generation under controlled settings, but they do not fully capture whether a model can follow action instructions and produce natural motion in real-time interaction. We therefore introduce Vidu-StreamBench, an in-house benchmark for real-time interactive avatar generation, while also reporting HDTF results as a standard public reference. We further measure inference efficiency on RTX 5090 GPUs to verify the real-time capability of Vidu S1.

Benchmarks. Vidu-StreamBench contains 500 samples, each composed of an action instruction, a reference first frame, and an audio clip. The benchmark covers diverse action commands, reference image styles, speaker attributes, emotions, and application scenarios. It is designed to test whether a model can generate avatars that move naturally, remain stable over time, preserve identity, follow user instructions, and maintain coherent audio-video behavior in realistic streaming settings. We conduct pairwise A/B preference tests against three leading commercial systems: HeyGen [52], LemonSlice [53], and Kling Avatar 2.0 [54].

For public benchmark evaluation, we use HDTF and compare Vidu S1 with the same commercial systems, as well as recent open-source models, including Wan2.2-S2V-14B [21], LiveAvatar [37], OmniAvatar-1.3B [55], Hallo3 [56], and StableAvatar-1.3B [57]. In detail, we report CSIM [58], Sync-D [59], and DOVER [60] to measure identity preservation, audio-visual synchronization, and perceptual video quality.

3.2 Results

3.2.1 Benchmark Evaluation

Analysis. Figure 3 summarizes the pairwise preference results on Vidu-StreamBench. Across comparisons with leading commercial avatar systems, Vidu S1 is consistently preferred overall. These results suggest that Vidu S1 is more effective at translating interactive user instructions into visible avatar behavior, including natural body motion, gestures, head movement, and controllable action execution, rather than only producing passive speech-driven facial animation.

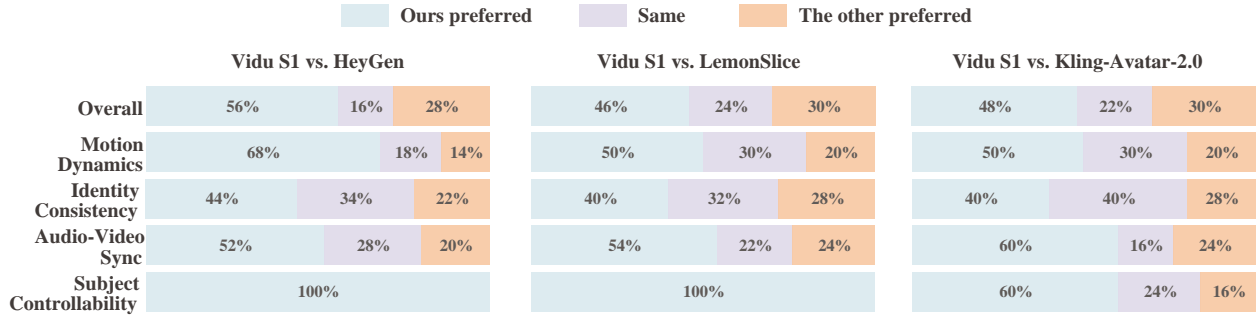


Figure 3 Human preference evaluation of Vidu S1 versus HeyGen, LemonSlice, and Kling-Avatar-2.0 on Vidu-StreamBench.

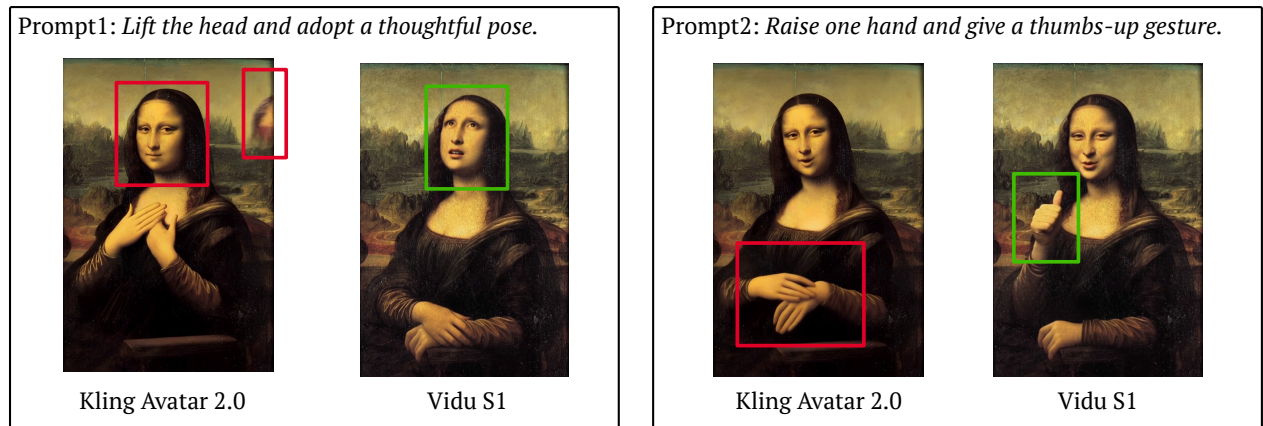


Figure 4 Qualitative comparison of instruction following and visual consistency. **Red boxes** highlight failure cases from the compared method: in Prompt 1, the subject does not raise the head and noticeable background artifacts appear; in Prompt 2, the thumbs-up gesture is not correctly performed. **Green boxes** highlight the results of Vidu S1: in Prompt 1, the model correctly follows the instruction by raising the head and adopting a thoughtful pose; in Prompt 2, it successfully performs the thumbs-up gesture.

The advantage is especially evident in subject controllability, where Vidu S1 achieves a 100% preference rate against HeyGen and LemonSlice. This indicates that human raters favored Vidu S1 when the task required the avatar to follow explicit action instructions. Vidu S1 also shows consistent gains in motion dynamics and audio-video synchronization, further supporting its suitability for real-time interactive avatar applications.

Visualization. Figure 4 compares Vidu S1 with Kling Avatar 2.0 on Vidu-StreamBench. Given the same reference image, audio, and action instruction, Vidu S1 demonstrates stronger visual stability and instruction following, leading to more engaging interactive behavior and better suitability for long-horizon use.

Table 1 reports complementary quantitative results on HDTF. Vidu S1 achieves leading performance across identity preservation, audio-visual synchronization, and perceptual quality, while also supporting real-time generation and instruction following.

3.2.2 Real-Time Interactive Generation Capability

Beyond benchmark quality, real-time interactive video generation requires low-latency inference, continuous user control, and stable streaming behavior. Vidu S1 is designed for this setting: it supports online prompt-level interaction and uses speech as an explicit control signal, allowing users to control the generated video during the generation process rather than specifying all conditions beforehand.

As shown in Table 1, Vidu S1 generates 540p video with a 3-step configuration and reaches an average throughput of 42 FPS on RTX 5090 GPUs. This exceeds the 30-FPS real-time playback threshold, demonstrating the practical feasibility of Vidu S1 for latency-sensitive applications such as live conversation, virtual hosts, entertainment avatars, and educational agents. Importantly, this inference efficiency is achieved while preserving the strong identity consistency, audio-visual synchronization, and perceptual quality demonstrated on HDTF, showing that Vidu S1 delivers real-time performance without sacrificing avatar generation quality.

4 Conclusion

We present Vidu S1, a real-time interactive video generation framework for voice-controlled digital characters. Vidu S1 enables users to guide video content at any moment through voice instructions and supports infinite-length generation with stable visual quality and subject identity. With an efficient inference and serving pipeline, it produces 540p videos at up to 42 FPS on consumer GPUs. Experiments show that Vidu S1 achieves the best overall performance while meeting real-time inference requirements. These results highlight the potential of Vidu S1 for practical, controllable, and personalized video generation. We hope this work provides a useful step toward real-time generative video systems that are realistic, consistent, and easy to control.

References

- [1] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.
- [2] Google DeepMind. Veo: A text-to-video generation system. Technical report, Google DeepMind, 2025. Veo 3 Tech Report.
- [3] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. [arXiv preprint arXiv:2503.20314](#), 2025.
- [4] Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. [arXiv preprint arXiv:2604.14148](#), 2026.
- [5] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. [Advances in Neural Information Processing Systems](#), 37:24081–24125, 2024.
- [6] Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. [Advances in Neural Information Processing Systems](#), 37:89834–89868, 2024.
- [7] Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In [Proceedings of the Computer Vision and Pattern Recognition Conference](#), pages 2568–2577, 2025.
- [8] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. [arXiv preprint arXiv:2502.06764](#), 2025.
- [9] Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. [arXiv preprint arXiv:2509.25161](#), 2025.
- [10] Mingzhen Sun, Weining Wang, Gen Li, Jiawei Liu, Jiahui Sun, Wanquan Feng, Shanshan Lao, SiYu Zhou, Qian He, and Jing Liu. Ar-diffusion: Asynchronous video generation with auto-regressive diffusion. In [Proceedings of the Computer Vision and Pattern Recognition Conference](#), pages 7364–7373, 2025.
- [11] Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models. In [Proceedings of the Computer Vision and Pattern Recognition Conference](#), pages 6322–6332, 2025.
- [12] Tianrui Feng, Zhi Li, Shuo Yang, Haocheng Xi, Muyang Li, Xiuyu Li, Lvmin Zhang, Keting Yang, Kelly Peng, Song Han, et al. Streamdiffusionv2: A streaming system for dynamic and interactive video generation. [arXiv preprint arXiv:2511.07399](#), 2025.

- [13] Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 22963–22974, 2025.
- [14] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. Advances in Neural Information Processing Systems, 38:167283–167308, 2026.
- [15] Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. arXiv preprint arXiv:2602.02214, 2026.
- [16] Min Zhao, Hongzhou Zhu, Kaiwen Zheng, Zihan Zhou, Bokai Yan, Xinyuan Li, Xiao Yang, Chongxuan Li, and Jun Zhu. Causal forcing++: Scalable few-step autoregressive diffusion distillation for real-time interactive video generation. arXiv preprint arXiv:2605.15141, 2026.
- [17] Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. arXiv preprint arXiv:2509.22622, 2025.
- [18] Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. arXiv preprint arXiv:2505.13211, 2025.
- [19] Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, et al. Skyreels-v2: Infinite-length film generative model. arXiv preprint arXiv:2504.13074, 2025.
- [20] Lvmin Zhang and Maneesh Agrawala. Packing input frame context in next-frame prediction models for video generation. arXiv e-prints, pages arXiv–2504, 2025.
- [21] Xin Gao, Li Hu, Siqi Hu, Mingyang Huang, Chaonan Ji, Dechao Meng, Jinwei Qi, Penchong Qiao, Zhen Shen, Yafei Song, et al. Wan-s2v: Audio-driven cinematic video generation. arXiv preprint arXiv:2508.18621, 2025.
- [22] Jintao Zhang, Kaiwen Zheng, Kai Jiang, Haoxu Wang, Ion Stoica, Joseph E Gonzalez, Jianfei Chen, and Jun Zhu. Turbodiffusion: Accelerating video diffusion models by 100-200 times. arXiv preprint arXiv:2512.16093, 2025.
- [23] Youhe Jiang, Haoxu Wang, Haotong Bao, Kai Jiang, Jianfei Chen, Jun Zhu, Fangcheng Fu, and Jintao Zhang. Turboserve: Serving streaming video generation efficiently and economically. arXiv preprint arXiv:2606.19271, 2026.
- [24] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. arXiv preprint arXiv:2509.17765, 2025.
- [25] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.
- [26] Kaifeng Gao, Jiabin Shi, Hanwang Zhang, Chunping Wang, Jun Xiao, and Long Chen. Ca2-vdm: Efficient autoregressive video diffusion model with causal generation and cache sharing, 2025.
- [27] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong MU, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In The Thirteenth International Conference on Learning Representations, 2025.
- [28] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 6613–6623, 2024.
- [29] Fu-Yun Wang, Zhaoyang Huang, Alexander W Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency models. Advances in neural information processing systems, 37:83951–84009, 2024.
- [30] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In International Conference on Learning Representations, volume 2024, pages 21875–21895, 2024.
- [31] Hidir Yesiltepe, Tuna Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 40256–40265, 2026.

- [32] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [33] Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression. *arXiv preprint arXiv:2512.05081*, 2025.
- [34] Haodong Li, Shaoteng Liu, Zhe Lin, and Manmohan Chandraker. Rolling sink: Bridging limited-horizon training and open-ended testing in autoregressive video diffusion. *arXiv preprint arXiv:2602.07775*, 2026.
- [35] Jintao Chen, Chengyu Bai, Junjun Hu, Xinda Xue, and Mu Xu. Grounded forcing: Bridging time-independent semantics and proximal dynamics in autoregressive video synthesis. *arXiv preprint arXiv:2604.06939*, 2026.
- [36] Youngraek Kim, Qixin Hu, C-C Jay Kuo, and Peter A Beerel. Memrope: Training-free infinite video generation via evolving memory tokens. *arXiv preprint arXiv:2603.12513*, 2026.
- [37] Yubo Huang, Hailong Guo, Fangtai Wu, Weiqiang Wang, Shifeng Zhang, Shijie Huang, Qijun Gan, Lin Liu, Sirui Zhao, Enhong Chen, et al. Live avatar: Streaming real-time audio-driven avatar generation with infinite length. *arXiv preprint arXiv:2512.04677*, 2025.
- [38] Kaiwen Zheng, Guande He, Min Zhao, Jintao Zhang, Huayu Chen, Jianfei Chen, Chen-Hsuan Lin, Ming-Yu Liu, Jun Zhu, and Qianli Ma. Causal-rcm: A unified teacher-forcing and self-forcing open recipe for autoregressive diffusion distillation in streaming video generation and interactive world models. *arXiv preprint arXiv:2606.25473*, 2026.
- [39] Ailing Zeng, Casper Yang, Chauncey Ge, Eddie Zhang, Garvey Xu, Gavin Lin, Gilbert Gu, Jeremy Pi, Leo Li, Mingyi Shi, et al. Lpm 1.0: Video-based character performance model. *arXiv preprint arXiv:2604.07823*, 2026.
- [40] Jintao Zhang, Rundong Su, Chunyu Liu, Jia Wei, Ziteng Wang, Haoxu Wang, Pengle Zhang, Huiqiang Jiang, Haofeng Huang, Chendong Xiang, et al. Efficient attention methods: Hardware-efficient, sparse, compact, and linear attention.
- [41] Jintao Zhang, Jia Wei, Pengle Zhang, Jun Zhu, and Jianfei Chen. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. In *International Conference on Learning Representations (ICLR)*, 2025.
- [42] Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization. In *International Conference on Machine Learning (ICML)*, 2025.
- [43] Jintao Zhang, Xiaoming Xu, Jia Wei, Haofeng Huang, Pengle Zhang, Chendong Xiang, Jun Zhu, and Jianfei Chen. Sageattention2++: A more efficient implementation of sageattention2. *arXiv preprint arXiv:2505.21136*, 2025.
- [44] Jintao Zhang, Jia Wei, Pengle Zhang, Xiaoming Xu, Haofeng Huang, Haoxu Wang, Kai Jiang, Jun Zhu, and Jianfei Chen. Sageattention3: Microscaling fp4 attention for inference and an exploration of 8-bit training. *arXiv preprint arXiv:2505.11594*, 2025.
- [45] Jintao Zhang, Marco Chen, Haoxu Wang, Kai Jiang, Ion Stoica, Joseph E Gonzalez, Jianfei Chen, and Jun Zhu. Sagebwd: A trainable low-bit attention. *arXiv preprint arXiv:2603.02170*, 2026.
- [46] Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattention: Accurate and training-free sparse attention accelerating any model inference. *arXiv preprint arXiv:2502.18137*, 2025.
- [47] Jintao Zhang, Kai Jiang, Chendong Xiang, Weiqi Feng, Yuezhou Hu, Haocheng Xi, Jianfei Chen, and Jun Zhu. Spargeattention2: Trainable sparse attention via hybrid top-k+ top-p masking and distillation fine-tuning. *arXiv preprint arXiv:2602.13515*, 2026.
- [48] Jintao Zhang, Haoxu Wang, Kai Jiang, Shuo Yang, Kaiwen Zheng, Haocheng Xi, Ziteng Wang, Hongzhou Zhu, Min Zhao, Ion Stoica, Joseph E. Gonzalez, Jun Zhu, and Jianfei Chen. Sla: Beyond sparsity in diffusion transformers via fine-tunable sparse-linear attention. *arXiv preprint arXiv:2509.24006*, 2025.
- [49] Jintao Zhang, Haoxu Wang, Kai Jiang, Kaiwen Zheng, Youhe Jiang, Ion Stoica, Jianfei Chen, Jun Zhu, and Joseph E Gonzalez. Sla2: Sparse-linear attention with learnable routing and qat. *arXiv preprint arXiv:2602.12675*, 2026.
- [50] Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.

- [51] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 3661–3670, 2021.
- [52] HeyGen. Heygen ai video avatar. <https://www.heygen.com/avatars/ai-video-avatar>, 2026. Accessed: 2026-07-02.
- [53] LemonSlice. Lemonslice studio: Create talking and singing ai avatar videos. <https://lemonslice.com/studio>, 2026. Accessed: 2026-07-02.
- [54] Kling Team, Jialu Chen, Yikang Ding, Zhixue Fang, Kun Gai, Yuan Gao, Kang He, Jingyun Hua, Boyuan Jiang, Mingming Lao, Xiaohan Li, Hui Liu, Jiwen Liu, Xiaoqiang Liu, Yuan Liu, Shun Lu, Yongsen Mao, Yingchao Shao, Huafeng Shi, Xiaoyu Shi, Peiqin Sun, Songlin Tang, Pengfei Wan, Chao Wang, Xuebo Wang, Haoxian Zhang, Yuanxing Zhang, and Yan Zhou. Klingavatar 2.0 technical report, 2025.
- [55] Qijun Gan, Ruizi Yang, Jianke Zhu, Shaofei Xue, and Steven Hoi. Omniaavatar: Efficient audio-driven avatar video generation with adaptive body animation. arXiv preprint arXiv:2506.18866, 2025.
- [56] Jiahao Cui, Hui Li, Yun Zhan, Hanlin Shang, Kaihui Cheng, Yuqi Ma, Shan Mu, Hang Zhou, Jingdong Wang, and Siyu Zhu. Hallo3: Highly dynamic and realistic portrait image animation with video diffusion transformer. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 21086–21095, 2025.
- [57] Shuyuan Tu, Yueming Pan, Yinming Huang, Xintong Han, Zhen Xing, Qi Dai, Chong Luo, Zuxuan Wu, and Yu-Gang Jiang. Stableavatar: Infinite-length audio-driven avatar video generation. arXiv preprint arXiv:2508.08248, 2025.
- [58] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4690–4699, 2019.
- [59] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. 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, pages 484–492, 2020.
- [60] Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In Proceedings of the IEEE/CVF international conference on computer vision, pages 20144–20154, 2023.