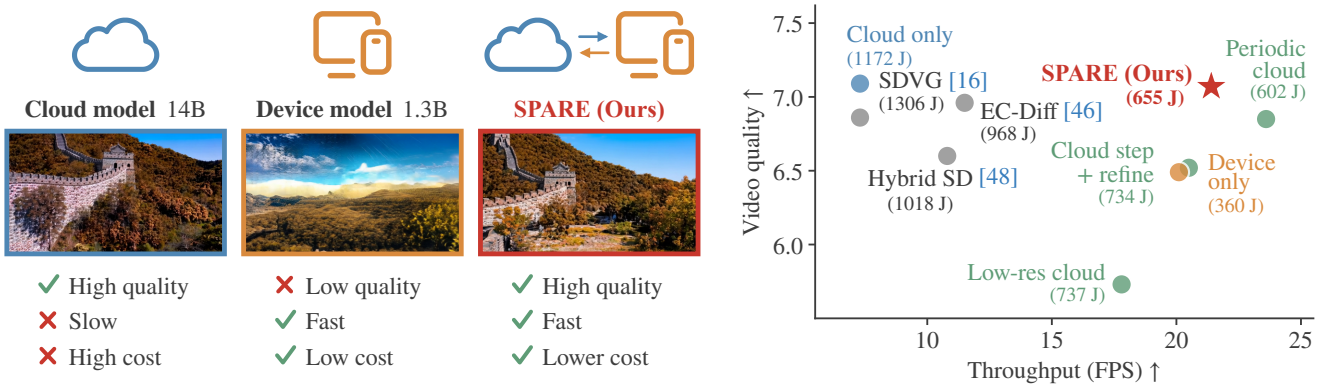
Heterogeneous Collaboration for Video World Models

Hyunsoo Kim Diana Marculescu Guandao Yang
The University of Texas at Austin



(a) Either model alone *vs.* their collaboration **(b) Quality *vs.* throughput (J: total energy)**

Figure 1. **SPARE combines the strengths of cloud and device models.** (a) The cloud model produces high-quality video but is slow and costly; the device model is fast and inexpensive but produces lower-quality video. SPARE retains similar video quality to the cloud model with real-time throughput and lower cloud energy. (b) Video quality (VLM judge [2]) versus throughput on WorldScore-300 [10]. Labels show total GPU energy per video second; green designs use 20-scene cost measurements. Neither axis starts at zero.

Abstract

Interactive video world models let users explore generated environments, empowering applications from games to robotics. To achieve good quality, video world models often need to run on cloud GPUs, which can be costly for model deployers. Cloud deployment can also lead to higher response latency. A small on-device model can respond locally, but it loses details and drifts over long rollouts. Motivated by the complementary strengths of cloud and edge computing, we study how to deploy a video world model across heterogeneous edge devices and cloud GPUs. Heterogeneous deployment is challenging because current models typically require substantial cloud computation to maintain video quality, increasing serving cost and potentially delaying responses to user inputs. We present SPARE (Single-step Prediction and Action-conditioned Regeneration at the Edge), a simple collaboration method that combines one-step cloud prediction with on-device decoding and regeneration when the action changes. Compared with the original cloud-only generation, SPARE achieves 2.95× *the throughput and reduces cloud GPU energy per second of generated video by a factor of* 2.90*, while maintaining similar visual quality.*

1. Introduction

Interactive video world models [1, 3, 4, 13, 15, 24, 43, 52] simulate how environments evolve in response to actions by generating future video frames conditioned on past observations and actions. These environments can support games, simulations, and the training of embodied agents. Recent models [17, 50] achieve real-time throughput on powerful cloud GPUs, but users typically access them through edge devices. Users expect continuous playback, low response latency, and videos that are visually realistic and physically plausible. The large cloud model offers high video quality but low FPS, while the small device model offers high FPS but lower video quality (see Fig. 1). Cloud deployment also introduces communication latency and serving costs. We study how the two models should collaborate to achieve good quality and responsiveness with lower cloud energy consumption.

We consider models that generate video in short groups of frames (*i.e.*, chunks) in response to user actions. Each chunk is conditioned on the preceding video and the current action, such as a movement direction or no movement. An action can remain unchanged across multiple frames. Each chunk is generated through denoising steps that predict a clean video latent from a noisy input. A model can run en-

tirely in the cloud or on the user's device. Some work studies how to use both a large and a small generative model. Hybrid SD [48] and RISE [20] divide the denoising steps of each sample between the two models. SDVG [16] instead uses the small model to generate drafts and the large model to regenerate rejected drafts. However, a cloud prediction can become outdated if the user changes the action before it is displayed, requiring additional computation and potentially delaying the response. We study how to allocate computation across successive chunks to handle such action changes.

In this paper, we systematically study how a large cloud model and a small on-device model should collaborate. We vary the number of cloud denoising steps, the device's role, and the generation history used by each model. We find that one cloud step retains much of the video quality of the four-step cloud model on LingBot-World (Table 1), allowing us to reduce the number of cloud denoising steps. The small model can regenerate a chunk using cloud-generated history. Its regenerated chunks are sharper with this history than when it generates the rollout alone (Appendix D). We therefore use the small model to regenerate chunks when the action changes.

We propose SPARE, a simple yet effective method that keeps both pretrained models frozen and combines one-step cloud lookahead with on-device decoding and regeneration. The cloud predicts the next chunk under the current action and sends its first-step x_0 prediction, an estimate of the clean video latent, for local VAE decoding (see Fig. 2). When the action changes, the device regenerates the chunk under the new action without waiting for the cloud. Both models continue from the displayed chunks. The action stream thus determines how generation is shared between the cloud and the device.

We evaluate SPARE on 300 WorldScore [10] scenes with actions changing every six seconds and no access to future actions, and on 158 WBench [51] navigation cases with future actions provided to the cloud. We report VBench quality and imaging quality, and WBench's individual quality dimensions on the navigation cases. Qwen3-VL-32B [2] provides additional video-quality and scene-drift assessments. On WorldScore-300, SPARE maintains video quality comparable to four-step cloud-only generation while increasing throughput from 7.3 to 21.4 FPS (2.95×) and reducing cloud GPU energy per second of generated video from 1,172 to 404 J (a factor of 2.90; Table 1). We also evaluate dynamic prompting with a different model backbone and measure reduced playback stalls during injected network outages (Appendix E).

Our contributions are summarized below.

- We study how the number of cloud denoising steps, device computation, and generation history affect the quality and cost of collaborative video generation.
- We propose SPARE, which combines one-step cloud lookahead with on-device decoding and regeneration when the action changes.
- We evaluate quality, responsiveness, and cost on predefined action sequences, WBench navigation cases, and a different model backbone, including playback during injected network outages.

2. Related Work

Interactive video world models. Interactive video world models [4, 13, 15, 26, 27, 35, 40, 54] generate action-conditioned environments, spanning neural game engines [9, 43], generative environments [4, 35], and controllable navigation and exploration [3, 13, 15, 26, 40]. Matrix-Game 3.0 [44] and WorldCrafter [53] use memory for long-horizon consistency. Few-step autoregressive generation builds on causal distillation in CausVid [50], rollout-based training in Self Forcing [17], and one-step training in One-Forcing [11]. minWM [55] provides a training pipeline for camera-controllable few-step models. Single-model efficiency improves through feature caching [12, 38], KV-cache compression [21, 45], sparse attention [29, 47], quantization [6], and small-model or low-bit local deployment [22, 36]. Light Interaction [28] combines training-free context management, caching, and sparse attention. Step requirements also depend on content: DIAMOND [1] reports fewer steps for deterministic transitions, DIDO [30] finds that interaction dynamics emerge after static content, and Speculative Forcing [8] uses token-level early exits and reports greater step requirements for viewpoint changes. These works improve a single generator. We study how two frozen models share generation as actions change.

Collaborative sampling with a large and a small model. Broader edge–cloud frameworks coordinate large cloud models and task-specific edge models for generative services [42]. For sampling, a line of work lets two diffusion models share each denoising trajectory. Some of these methods target edge–cloud deployment. Hybrid SD [48] and RISE [20] run the early steps on a large cloud model and the late steps on a small edge model, EC-Diff [46] extrapolates part of the cloud steps, and EcoVideo [5] generates keyframes in the cloud and interpolates the remaining frames on the device, which needs future frames. For autoregressive video, SDVG [16] lets a large model regenerate the drafts of a small model that a reward model rejects. Others share the steps between the two models to cut the sampling cost on one machine. T-Stitch [34] runs the small model first, SRDiffusion [7] switches from the large to the small model adaptively, FlowBlending [37] gives the large model the first and last steps, TRACK [33] calibrates the model of each step offline, and HybridStitch [39] selects the model per token. Except for SDVG, these methods were

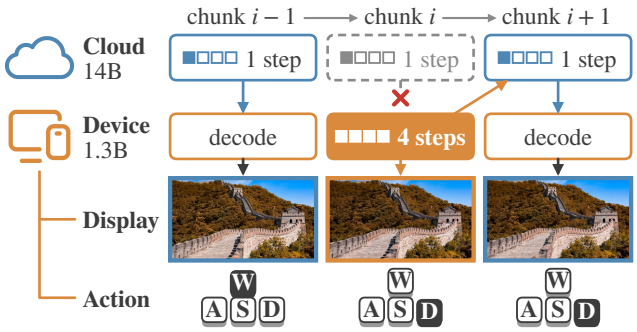


Figure 2. **Overview of SPARE** on three consecutive chunks around an action change. While the action stays the same (chunk $i-1$), the cloud predicts the next chunk under the last action it received with one denoising step, and the device decodes and displays this estimate. When the action changes (chunk i), the prediction made under the old action is discarded. The device regenerates the chunk under the new action with its own four-step sampler and uploads it, and the cloud continues from it (chunk $i+1$). Both models write only the displayed chunks into their caches. The border of a frame shows which model generated the chunk.

designed for offline generation with many steps, and each of them keeps the large model involved in every sample or verifies every draft. In contrast, SPARE follows the action stream of an interactive video. It displays the one-step estimate of the large model directly and runs a four-step denoising schedule only where the action changes.

Speculation in interactive systems. Interactive systems hide latency by computing ahead of the user. Outatime [25] renders cloud-gaming frames for likely inputs and corrects mispredictions. For video generation, speculative sampling [18] reuses predictions from a bidirectional window while their conditioning remains valid, and ActionSplice [41] adapts an in-flight sampler state to a revised action. For language models, edge–cloud speculative decoding [14, 31] over wide-area networks gains little latency and mainly saves server capacity. SPARE also predicts one chunk ahead, but unlike these systems it speculates with the large model and repairs a wrong prediction with a different, smaller model on the device, which receives the action first.

3. SPARE

In this section, we propose SPARE (Single-step Prediction and Action-conditioned Regeneration at the Edge), a heterogeneous collaboration method for video world models (Fig. 2). Section 3.1 introduces video world models and the problem setup for heterogeneous collaboration. Section 3.2 describes one-step lookahead in the cloud. Finally, Section 3.3 presents on-device decoding and regeneration, along with the shared history maintained by the two models.

3.1. Preliminaries

Chunk-wise autoregressive world models. We consider an action-conditioned video diffusion model [17, 50] that generates a video chunk by chunk. Chunk i is a latent $z_i \in \mathbb{R}^{F\times H\times W\times C}$ of F latent frames, $H \times W$ positions, and C channels, generated from the action a_i and the previous chunks $z_{<i}$, to which the model attends through a key–value (KV) cache. A VAE decoder maps it to video frames. The model is a few-step generator G_θ with parameters θ and timesteps $t_T > \cdots > t_1$ ($T = 4$ in our models). Starting from noise $z_{i,T} \sim \mathcal{N}(0, I)$, step $j = T, \ldots, 1$ predicts the clean chunk

$$\hat{z}_{i,j} = G_\theta\big(z_{i,j},\, t_j,\, a_i,\, z_{<i}\big), \tag{1}$$

which is re-noised to the next timestep, $z_{i,j-1} = (1 - \sigma_{t_{j-1}})\,\hat{z}_{i,j} + \sigma_{t_{j-1}}\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$. The estimate of the last step is the chunk, $z_i = \hat{z}_{i,1}$, and one more forward pass at $t = 0$ writes it into the KV cache.

Problem setup. The edge device, which we call the device from here on, runs a small generator G_{dev} and the cloud a large generator G_{cloud} of the same family. Both operate in the same latent space, and each keeps its own KV cache. The user issues actions online, and the device must display a new chunk every Δ seconds ($\Delta = 0.75$ s for 12 frames at 16 FPS). The goal is low cloud GPU usage with high quality and responsiveness.

Design space. A collaboration design fixes, for every chunk, (i) which model runs which denoising steps, (ii) where the chunk is decoded, and (iii) which chunks enter the history of each model. The two-model samplers of Sec. 2 allocate denoising steps or tokens between models, or regenerate rejected drafts. We instead use action changes to decide when the device regenerates a cloud prediction. The large model runs one step per chunk, and the small model runs its four-step schedule when the action changes. Sec. 4.3 evaluates other points of this design space under the same protocol.

3.2. One-Step Lookahead in the Cloud

While chunk $i-1$ is displayed, the device requests chunk i under the current action, which serves as the predicted action $\hat{a}_i$. The cloud computes only the clean-latent prediction from its first denoising step,

$$\hat{z}_i = G_{\text{cloud}}\big(z_{i,T},\, t_T,\, \hat{a}_i,\, z_{<i}\big), \tag{2}$$

and returns it with its chunk index and action. Lookahead overlaps cloud generation and communication with playback, provided that the predicted chunk arrives before its playback deadline. A cloud chunk therefore costs one denoising pass and one cache update instead of T passes and

an update. LingBot-World 2.0 combines consistency distillation with distribution matching over self-rollouts [13]. Related methods supervise a randomly selected denoising step [17, 49]. We base our use of the first-step prediction on its measured video quality (Table 1; Appendix D). The cloud never decodes. It sends latents, at 3.6 Mbit/s in FP16 and about half of that with 8-bit quantization (see Appendix E).

3.3. Decoding and Regeneration on the Device

Decoding. If $\hat{a}_i$ equals the action a_i that is in effect when chunk i starts, the device decodes $\hat{z}_i$ and writes it into its own cache. Decoding overlaps the next request. The VAE decoder, a substantial part of the per-chunk work, thus stays off the cloud. The decoder-only peak allocation in the 20-scene timing test is 8.8 GiB. Bidirectional latent traffic is 4.2 Mbit/s, three to five times H.264 streaming of the same frames (Table 1).

Regeneration when the action changes. If $\hat{a}_i \neq a_i$, the prediction was made under the previous action and is discarded as stale. The device regenerates the chunk with all T steps of G_{dev} under a_i, decodes it, and uploads the latent. Any mismatch triggers the same four-step regeneration, without a threshold on the magnitude of the action change within the model's action space. Action-change regeneration runs locally without waiting for a cloud response. For the outage tests, we additionally enable a timeout-based takeover: the device generates a chunk if none arrives from the cloud for 0.9 s. This extension is off in the ordinary delay tests (Appendix E). As the new action usually arrives while the previous chunk is displayed, the device starts regenerating as soon as the action arrives.

Shared history. Before predicting chunk $i+1$, the cloud commits the chunk selected for display at i: its own estimate if displayed, or the uploaded device latent otherwise, including during takeover. Both models therefore condition on exactly the displayed chunks $z_{\leq i}$, and no cache tensor is ever transmitted. We verify this property for both of our implementations (see Appendix G).

Decode-only devices. The T device denoising steps and decoding must fit into one chunk interval Δ. Otherwise, the device requests and decodes a one-step cloud estimate under a_i. Lookahead and shared history remain unchanged. This is SPARE without the device generator, evaluated in Secs. 4.4 and 5.

4. Experiments

4.1. Setup

Models and hardware. We use four-step distilled LingBot-World 2.0 causal-fast [13]: 1.3B on the device and 14B in the cloud, generating three latent frames (12 video frames) per chunk at about 624×352. Two available RTX PRO 6000 Blackwell GPUs (96 GB), one per model in one host with injected delays, control for hardware differences. This tests a workstation device, not a phone or a wide-area deployment. Peak memory is 12.1 GiB on the device (8.8 GiB for decoding alone) and 52.5 GiB in the cloud. The device achieves 20.1 FPS, above 16-FPS playback. A separate 24 GB RTX A5000 test falls below this rate and uses cloud regeneration with device decoding (Appendix E).

Benchmarks. Table 1 reports results on WorldScore-300 [10]: ten scenes per category across 30 categories, with 30-second rollouts under five six-second action segments. The action changes four times per rollout, and the cloud has no access to future actions, so SPARE regenerates four of the 39 chunks on the device. We also evaluate the 158 WBench navigation cases (WBenchNavi) [51], with all actions provided to the cloud before generation (full results in Appendix C). The design study, ablation, and analyses in Secs. 4.3–5 use 20 scenes under the same protocol with four or ten seeds (Appendix A).

Baselines. We compare three groups of methods. *(i) Single models* are the device and cloud models alone, the cloud model with one step per chunk, the 4-bit quantized 14B model on the device, and Speculative Forcing [8]. *(ii) Two models on two GPUs* are Hybrid SD [48] and EC-Diff [46], which divide every denoising trajectory between a cloud and an edge model, and SDVG [16], in which a large model regenerates the rejected drafts of a small one. In our synchronous implementations of all three, the cloud waits for the device on every chunk, without chunk-ahead pipelining. We also evaluate five additional step- and token-sharing designs from offline image and video diffusion (Appendix C). *(iii) Edge–cloud collaboration* is SPARE. Table 1 also includes its variant without the device generator (Sec. 4.4; Appendix D). All methods use the same model pair and hardware, and each published method is instantiated on our four steps with its configuration fixed before the evaluation (see Appendix A).

Metrics. For video quality, we report our six-dimension VBench [19] aggregate and its imaging-quality dimension (Imag.), the MUSIQ [23] score between 0 and 100. On WBenchNavi we also report seven WBench basic dimensions separately, not an overall score (Appendix B). For action following, we estimate camera poses from the gener-

Table 1. **Rollouts with changing actions.** WorldScore-300 (300 scenes, one seed), one RTX PRO 6000 per model. Cloud-only rows also decode in the cloud. Network is bidirectional payload per displayed frame; H.264 cloud-only streaming uses 6–10 kB/frame (Appendix E). VBench aggregates six quality dimensions. Rankings cover two-model methods and both SPARE variants. †Future actions known. Implementation details: Appendix A.

		VBench		Camera / motion		VLM judge		Energy (J / video s) ↓			Latency	Network
Method	FPS ↑	Quality ↑	Imag. ↑	RPE ↓	Resp. ↑	Quality ↑	Drift ↓	Device / GPU 1	Cloud / GPU 2	Total	(s) ↓	(kB / frame) ↓
Single model												
Device only (1.3B)	20.1	76.2	61.9	0.653	86.6	6.49	64.8	360	0	360	0.592	0
Cloud only (14B)	7.3	77.0	66.4	0.483	84.0	7.09	53.5	0	1172	1172	1.657	–
Cloud only (14B), 1-step†	14.1	77.5	65.2	0.371	83.5	7.03	48.2	0	567	567	0.856	–
14B on the device, 4-bit	5.0	76.4	65.4	0.454	85.2	6.99	51.2	1705	0	1705	2.404	0
Speculative Forcing [8]	22.5	73.2	61.0	0.834	85.9	5.37	71.1	340	0	340	0.565	0
Two models on two GPUs												
Hybrid SD [48]	10.8	76.1	61.8	0.507	84.2	6.60	58.7	373	645	1018	1.107	87.5
EC-Diff [46]	11.5	76.9	64.8	0.435	85.4	6.96	51.3	334	634	968	1.042	58.6
SDVG [16] ($\tau = -1.2$)	7.3	76.7	64.3	0.564	86.2	6.86	58.1	681	625	1306	1.152	**29.6**
Edge–cloud collaboration												
SPARE w/o device generator	21.3	77.5	65.2	**0.371**	83.5	7.03	48.2	**208**	422	**630**	0.801	32.5
SPARE (Ours)	**21.4**	**77.9**	**65.3**	0.382	**87.2**	**7.07**	**46.6**	251	**404**	655	**0.595**	32.5

ated video and report the rotation error (RPE) to the commanded trajectory and a Response score that rewards an early onset of motion after an action change. As additional measures, a VLM judge rates video quality from 1 to 10 and scene drift from 0 to 100. These assess appearance and its degradation, not action following; the appendix defines all metrics. For efficiency, we report the throughput in decoded frames per second, the energy of each GPU per second of generated video, the latency from an action to its first decoded frame, and the data exchanged between the device and the cloud per displayed frame. The video plays at 16 FPS, so a method runs in real time only if its throughput reaches this rate. Energy is measured from NVML total-energy counters, including idle power and discarded computations but excluding host and network energy. In all tables, bold and underlining mark the best and second-best observed values within each comparison group, unless stated otherwise. All costs are per video stream, with one cloud GPU per device. A difference is significant when its paired 95% bootstrap interval excludes zero, and Appendix B adds a Holm correction.

4.2. Main Results

Comparison with single models. The six-dimension VBench quality is 77.9 against 77.0 for the four-step cloud model, although imaging quality is significantly lower, 65.3 against 66.4 MUSIQ. Against the device model, both VBench quality and imaging quality rise, from 76.2 to 77.9 and from 61.9 to 65.3, at similar throughput. Cloud GPU energy falls by a factor of 2.90, and throughput rises 2.95×, from 7.3 to 21.4 FPS. These ratios use unrounded measured means. On LingBot-World, one-step cloud generation and on-device decoding improve throughput and cloud-energy efficiency, while local regeneration enables earlier decoded-frame availability and reduces playback stalls under the tested buffer and outage conditions. Against one-step cloud generation, the cloud-energy saving is 1.4×, but total energy is higher, 655 against 567 J. The additional VLM quality scores, 7.07 versus 7.09, meet the equivalence test in Appendix B. Drift and rotation error are lower than for the device and four-step cloud models, but SPARE also moves less. Mean dense flow on WorldScore-300 is 2.16 pixels/frame for SPARE, 2.15 for one-step cloud, and 2.68 for four-step cloud (Farnebäck, 312 × 176, alternate frame pairs; Appendix B). Lower drift and RPE therefore do not establish better camera following; Response and direction errors assess it separately. The first decoded frame is ready in 0.595 s, versus 0.592 s for the device model and 1.66 s for cloud-only generation. Bidirectional traffic is 32.5 kB/frame, 4.2 Mbit/s, three to five times H.264 streaming of the same frames but less than EC-Diff and Hybrid SD (Appendix E). Speculative Forcing falls below the device model at its published thresholds (Appendix F).

Comparison with two-model methods. Hybrid SD and EC-Diff split each denoising trajectory between the two models, while SDVG regenerates rejected device drafts in the cloud. All three remain below the 16-FPS playback rate. EC-Diff comes closest in VBench quality (76.9 versus 77.9) and imaging quality (64.8 versus 65.3). Its VLM-quality difference from SPARE is not significant after the Holm correction. SDVG uses about twice the total energy of SPARE, although it exchanges less data.

Appendix C reports comparisons with five additional diffusion baselines.

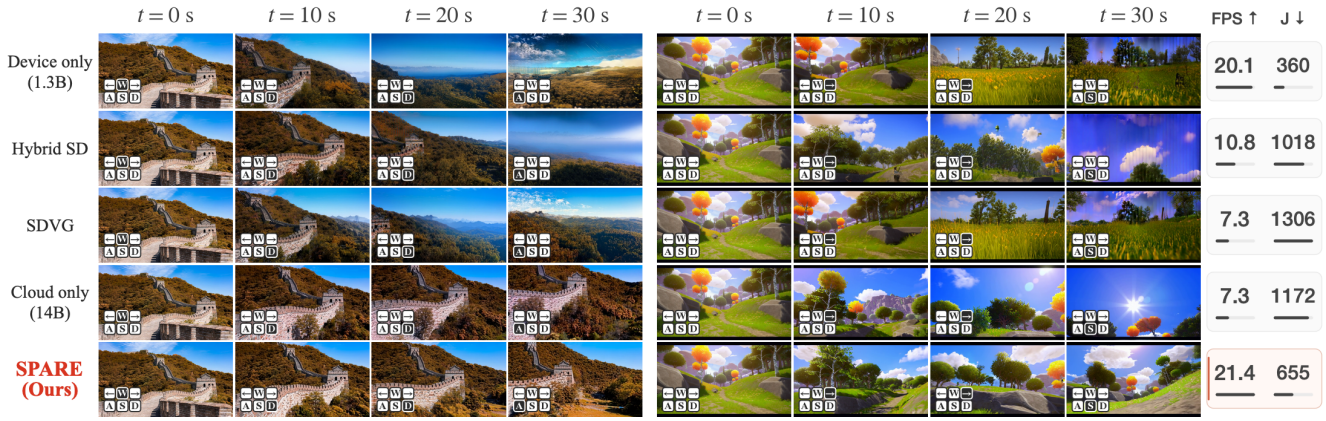


Figure 3. **Qualitative comparison under changing actions.** Two 30-second rollouts without access to future actions. Every method receives the same input image, actions, and seed, and the keys in each frame show the active action (W and S for forward and backward, A and D for left and right, arrows for turns, no key for no input). The device model, Hybrid SD, and SDVG lose the wall (left) and the stylized trees (right) during the rollout, whereas the cloud model and SPARE keep both scenes to the end. Right: FPS and total GPU energy (J per video second), averaged over WorldScore-300 (Table 1).

Qualitative comparison. Figure 3 shows selected 30-second rollouts with matched input images, action sequences, and seeds. The visible wall and trees illustrate appearance over time; differing generated camera motion prevents these frames alone from establishing action accuracy. Table 1 reports camera error and motion response.

WBench navigation sequences. On WBenchNavi [51], the cloud has access to future actions, so SPARE never regenerates and reduces to one-step cloud generation with on-device decoding. The six-dimension VBench quality rounds to 81.9 for both SPARE and four-step cloud, with imaging quality of 73.3 versus 73.4 MUSIQ. WBench imaging scores are 0.674 versus 0.675, while aesthetic quality is lower, 0.631 versus 0.646 (Appendix C). The additional VLM quality scores are 8.29 versus 8.37, a non-significant difference. T-Stitch scores 8.41, significantly above SPARE. Its 12.0 FPS with two cloud steps is measured separately on the 20-scene changing-action test (Appendix C).

SPARE and four-step cloud score 7.27 and 7.40 in VLM quality on 20 scenes × 10 seeds, and 6.61 and 7.00 for 120-second rollouts (12 scenes × 3 seeds; see Appendix C).

4.3. Collaboration Design Choices

Figure 4 varies the design of SPARE in two directions, with quality on WorldScore-300 and cost on 20 scenes.

Role of the device. If the device upsamples a low-resolution cloud chunk, the quality falls below that of the device model, to 5.73 against 6.49, and if it refines every one-step cloud estimate with three guided steps, the quality drops from 7.07 to 6.52 at a similar cost. Periodic cloud,

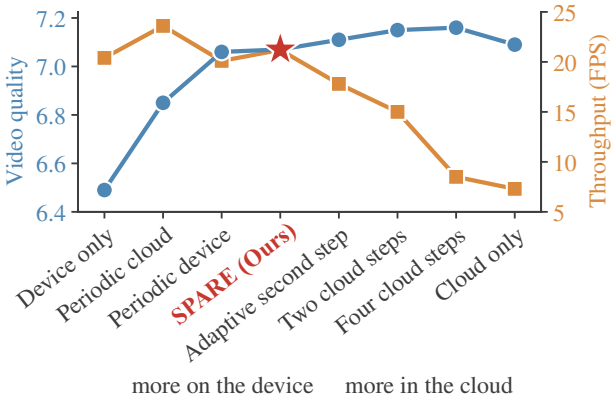


Figure 4. **Collaboration design choices.** Video quality on WorldScore-300 (blue) changes little beyond SPARE, whereas throughput (orange) decreases as more denoising steps are assigned to the cloud. See Appendices C and D.

in which the device generates every second chunk, has the lowest cloud energy, 307 against 398 J for SPARE, but its quality falls to 6.85 and its rotation error is significantly higher. Letting the device generate every fourth chunk lowers the cloud energy from 398 to 370 J at preserved quality (7.06) but with a significantly higher rotation error (see Appendix C). Which chunks the device generates thus matters more than how many.

Cloud computation. Two cloud steps raise quality from 7.07 to 7.15, not significantly, at 1.5× the cloud energy and 15.0 FPS. Four steps reach 7.16 at 2.5× the energy and 8.5 FPS, with significantly higher quality, drift, and rotation error. A learned predictor that selects half the chunks for

Table 2. **Ablation results.** The upper block removes components one after another, and the lower block changes how the invalidated chunk is handled. Rows 1–3 give identical videos, row 4 is the four-step cloud model. Lat. is the time to the first decoded frame in seconds ([†]quality on 40 clips, the new action is never shown). The scores differ from Table 1, which uses WorldScore-300.

Variant	FPS ↑	Quality ↑	Imag. ↑	RPE ↓	Lat. ↓	Cloud J ↓
Components removed cumulatively						
– device model	**21.3**	7.23	65.0	0.521	0.79	413
– lookahead	18.9	7.23	65.0	0.521	0.63	403
– device decoding	14.2	7.23	65.0	0.521	0.84	551
– one-step sampling	7.3	**7.33**	**68.7**	0.729	1.63	1141
Chunk invalidated by an action change						
No regeneration[†]	21.1	7.23	65.5	0.690	–	400
Stale-cloud guidance	20.1	7.31	66.5	0.462	0.80	401
SPARE (Ours)	21.2	7.28	67.0	**0.461**	**0.59**	**398**

a second step reaches 7.11 at 1.3× the energy: not significant on WorldScore-300, but significant on 20 scenes × ten seeds (Appendix C). All three significantly raise imaging quality, from 65.3 to 65.8–66.5. These significance tests compare with one-step SPARE. The four-step variant retains device decoding, lookahead, and local regeneration; its score of 7.16 alone does not establish an advantage over cloud-only generation at 7.09.

4.4. Ablation Study

Table 2 removes the components of SPARE one after another. Without the device generator, the cloud regenerates the invalidated chunks with one step and the device only decodes. A new action then waits for the cloud, and the latency rises from 0.59 to 0.79 s (0.60 to 0.80 s on WorldScore-300, see Appendix D), so the device model shortens the time to the first decoded frame. Whether this also reduces input-to-display latency depends on the playback buffer (Sec. 5). Removing lookahead as well lowers the throughput to 18.9 FPS but also the latency, to 0.63 s, because no chunk is predicted under an old action. Lookahead improves throughput but delays the first decoded frame after an action change, and local regeneration reduces this delay. Without lookahead the player stalls under delay (Appendix E). Without device decoding the throughput falls to 14.2 FPS, below the playback rate, and the cloud energy rises from 403 to 551 J. Four cloud steps per chunk then give a quality of 7.33, not significantly above SPARE, at 7.3 FPS and 2.9× the cloud energy of SPARE. If the invalidated chunk is shown instead of regenerated, the camera moves the wrong way in 58.6% of the first chunks after a change, against 7.1% with SPARE, and guiding device regeneration with the stale cloud prediction lowers the Response score from 87.1 to 82.3 (see Appendix D). Guidance also makes the device wait for the invalidated chunk, raising the latency to 0.80 s.

Table 3. **Local regeneration and network outages.** SPARE against its variant without the device generator; both use lookahead. Latencies are medians, and energy is in J per video second. (a,b) 20 scenes, 80 inputs per row. (c) Replay of (b) with a 0.3-s buffer and catch-up; only display latency and stalls are recomputed. CPU-ready latency and energy are from the source runs. (d) WorldScore-300, 1200 inputs per row, with takeover enabled for SPARE. See Appendix E.

	Latency (s) ↓		Stall ↓	Energy ↓	
Method	CPU-ready	Display	(s/clip)	Cloud	Total
(a) Real-time inputs, 0 ms one-way delay					
w/o device generator	0.59	0.78	**0.00**	431	**657**
SPARE (Ours)	**0.39**	0.77	**0.00**	**414**	683
(b) Real-time inputs, 150 ms one-way delay					
w/o device generator	0.90	**1.07**	**0.00**	431	**657**
SPARE (Ours)	**0.49**	**1.07**	**0.00**	**415**	684
(c) Logged-time replay, 150 ms delay					
w/o device generator	0.90	0.90	0.68	431	**657**
SPARE (Ours)	**0.49**	**0.76**	**0.17**	**415**	684
(d) WorldScore-300, 2-s network outage					
w/o device generator	**0.64**	1.92	1.33	441	**668**
SPARE (Ours)	**0.64**	**0.92**	**0.29**	**411**	723

5. Further analysis

Response to action changes. Table 3 uses wall-clock inputs rather than the chunk-boundary clock of Table 1. Local regeneration makes the first decoded frame available 0.20–0.41 s earlier, but the player's accumulated buffer hides this gain. Replaying the logged times at 150 ms with target buffers of 0.2–0.4 s reduces display latency by 0.04–0.24 s and lowers stalls; from 0.5 s the display latencies are equal. This is a playback analysis of recorded timings, not a new closed-loop run.

Network outages on WorldScore-300. On WorldScore-300, all messages are held for 2 s starting 10 s into each clip (Table 3). Takeover reduces stalls in every scene, from 1.33 to 0.29 s per clip on average, and median display latency from 1.92 to 0.92 s. CPU-ready latency remains 0.64 s for both paths, but total GPU energy rises from 668 to 723 J per video second. Outage video quality has not been scored.

Number of cloud steps. With future actions, one cloud step scores 7.03 versus 7.09 for four steps on WorldScore-300, and 7.21 versus 7.40 on 20 scenes × ten seeds. It retains most of the VLM quality gain over 1.3B but has lower imaging quality throughout the rollout. Its first-step estimate is closer to its four-step output than the 1.3B model's

Table 4. **Device denoising steps at action changes.** Twenty scenes × one seed. MUSIQ covers the first chunk after each change; rotation RPE and VLM drift cover the whole clip. †The cloud reference knows future actions. Paired intervals and later-chunk metrics are in the appendix.

Regeneration	MUSIQ ↑	Rot. RPE (°) ↓	VLM drift ↓
Cloud, 1 step†	63.04	0.593	46.0
Device, 1 step	57.86	0.488	44.8
Device, 2 steps	62.96	0.480	40.3
Device, 4 steps (SPARE)	**66.15**	**0.402**	**39.3**

(Appendix D). On the different backbone, one-step cloud quality falls below the device model.

Frequency of action changes. In the three-second setting, SPARE scores 7.16 in VLM-judged visual quality against 7.58 for the four-step cloud model (20 scenes × 4 seeds, Table 14). Action sequences differ across the frequency settings, so differences between settings cannot be attributed to frequency alone. These tests use piecewise-constant actions. Continuous per-frame control remains untested.

Device *vs*. cloud regeneration. Without the device generator, the cloud regenerates invalidated chunks with one step. Its VLM quality is equivalent to SPARE (7.03 versus 7.07), and throughput is nearly unchanged (21.3 versus 21.4 FPS). Local generation therefore does not explain the overall throughput gain over four-step cloud generation. The device model saves 4% of cloud energy but costs 4% more total energy. On WorldScore-300 it improves VBench quality and the Response score after translations, but Response is lower after turns, and the imaging and rotation gains on the 20-scene set do not carry over (Appendix D). Its benefits are earlier decoded-frame availability and fewer playback stalls during injected outages, subject to the playback buffer (Table 3; Appendix E). Table 4 isolates the number of device denoising steps at an action change.

Number of device steps. Table 4 holds the noise, cloud model, and four action changes fixed; only changed-action chunks use the device sampler. Four steps improve first-chunk MUSIQ over one step by 8.29 (95% CI $[4.62, 12.45]$) and over the one-step cloud reference by 3.11 ($[0.25, 6.22]$). On this one-seed test, whole-clip rotation error improves significantly over one device step but not two; the drift comparison is descriptive. Later-chunk MUSIQ does not differ significantly. Appendix D reports paired intervals and direction errors.

Table 5. **Evaluation on a different model backbone.** Self-Forcing 1.3B / Krea Realtime 14B, dynamic prompting. Quality/Drift are VLM scores; Cloud J is J/video s.

Method	FPS ↑	Quality ↑	Drift ↓	Imag. ↑	Cloud J ↓
Device only (1.3B)	**14.5**	7.72	29.0	<u>67.7</u>	0
Cloud only (14B)	3.8	**8.13**	<u>19.0</u>	**69.7**	2406
Cloud only (14B), 1-step	8.0	7.60	21.5	65.2	1098
SPARE w/o device generator	8.8	7.60	21.5	65.2	<u>1025</u>
SPARE (Ours)	<u>9.8</u>	<u>7.95</u>	**15.4**	66.6	**844**

Evaluation on a different model backbone. We apply SPARE to Self-Forcing 1.3B [17] and Krea Realtime 14B [32] with four prompt changes and no advance access. Final-segment MUSIQ is 66.1 against 63.6 for one-step cloud, though whole-clip imaging quality remains below four-step cloud. The additional VLM quality score is 7.95 versus 7.60 (95% CI of the difference $[0.26, 0.44]$), despite the reference knowing future prompts. Without the device generator, throughput falls from 9.8 to 8.8 FPS and cloud energy rises from 844 to 1025 J. Four device steps take 0.48 s versus 0.94 s for one cloud step. Throughput remains below playback rate. This dynamic-prompt test has no network or playback clock (Appendix F).

6. Conclusion

SPARE combines one-step cloud prediction with on-device decoding and regeneration when the action changes. On LingBot-World, its six-dimension VBench quality is 77.9 versus 77.0 for four-step cloud generation, with cloud GPU energy reduced by a factor of 2.90 and no access to future actions. Takeover reduces stalls in the tested 2-s outages, but requires a device fast enough for playback and increases total GPU energy over the variant without the device generator.

Limitations and future work. One-step chunks are softer than four-step chunks, so the imaging quality of SPARE stays below that of the cloud model. Local regeneration requires workstation-class compute at playback rate; the RTX A5000 test instead uses cloud regeneration (Appendix E). In the 20-scene 2-s outage test, the device generates 16 of the 39 chunks per clip on average while the cloud catches up. Quality also depends on the action sequence (Sec. 5). Chunks span 0.75 s, but response latency also includes computation, communication, and buffering; continuous inputs would leave most chunks to the device model. Evaluation uses predefined actions and automatic scores, not human play. Both GPUs share one host with injected delays and one cloud GPU per device, without user batching (Appendix G). Human evaluation, closed-loop play, wide-area deployment, and role-specific training remain open.

LLM Usage Statement

In this work, we used generative AI tools from Anthropic and OpenAI to assist with manuscript drafting and revision, experiment planning, code implementation and debugging, analysis of experimental outputs, and preparation of figures and qualitative comparisons. We also used Qwen3-VL-32B as an automatic video evaluator, as described in our evaluation protocol. These model-based scores do not constitute human evaluation. The authors are responsible for verifying all AI-assisted content and for the final methods, experiments, results, and claims presented in this work.

References

[1] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1, 2

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. 1, 2

[3] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 2

[4] Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024. 1, 2

[5] Jiayu Chen, Hengyi Zhang, Maoliang Li, Minyu Li, Zihao Zheng, Xuanzhe Liu, Guojie Luo, and Xiang Chen. EcoVideo: Entropy-orchestrated video generation paradigm in cloud-edge dynamics. In *European Conference on Computer Vision (ECCV)*, 2026. 2

[6] Yukang Chen, Luozhou Wang, Wei Huang, Shuai Yang, Bohan Zhang, Yicheng Xiao, Ruihang Chu, Weian Mao, Qixin Hu, Shaoteng Liu, Yuyang Zhao, Huizi Mao, Ying-Cong Chen, Enze Xie, Xiaojuan Qi, and Song Han. LongLive-2.0: An NVFP4 parallel infrastructure for long video generation. *arXiv preprint arXiv:2605.18739*, 2026. 2

[7] Shenggan Cheng, Yuanxin Wei, Lansong Diao, Yong Liu, Bujiao Chen, Lianghua Huang, Yu Liu, Wenyuan Yu, Jiangsu Du, Wei Lin, and Yang You. Srdiffusion: Accelerate video diffusion inference via sketching-rendering cooperation. *arXiv preprint arXiv:2505.19151*, 2025. 2

[8] Hyunwook Choi, Seungryong Kim, Dongyoon Han, and Sangdoo Yun. Speculative forcing for efficient autoregressive video generation. *arXiv preprint arXiv:2610.22347*, 2026. 2, 4, 5

[9] Decart AI, Julian Quevedo, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. https://oasis-model.github.io/, 2024. 2

[10] Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 1, 2, 4

[11] Jiaqi Feng, Justin Cui, Yuanhao Ban, and Cho-Jui Hsieh. One-Forcing: Towards stable one-step autoregressive video generation. *arXiv preprint arXiv:2605.23458*, 2026. 2

[12] Weilun Feng, Guoxin Fan, Haotong Qin, Mingqiang Wu, Yuqi Li, Xiangqi Li, Zhulin An, Libo Huang, Dingrui Wang, Longlong Liao, Michele Magno, Yongjun Xu, and Chuanguang Yang. WorldCache: Accelerating world models for free via heterogeneous token caching. In *International Conference on Machine Learning (ICML)*, 2026. 2

[13] Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, Ka Leong Cheng, Haojie Zhang, Jian Gao, Tianrui Feng, Yuzheng Liu, Yao Yao, Yinghao Xu, Xing Zhu, Yujun Shen, and Hao Ouyang. Infinite worlds with versatile interactions. *arXiv preprint arXiv:2607.07534*, 2026. 1, 2, 4

[14] Yunhe Han, Yunqi Gao, Bing Hu, Mahdi Boloursaz Mashhadi, Yitong Duan, Pei Xiao, and Yanfeng Zhang. PipeSD: An efficient cloud-edge collaborative pipeline inference framework with speculative decoding. *arXiv preprint arXiv:2605.13319*, 2026. 3

[15] Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Size Wu, Wei Li, Xuchen Song, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-Game 2.0: An open-source, real-time, and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025. 1, 2

[16] Yuezhou Hu and Jintao Zhang. Speculative decoding for autoregressive video generation. *arXiv preprint arXiv:2604.17397*, 2026. 1, 2, 4, 5

[17] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 1, 2, 3, 4, 8

[18] Yubo Huang, Yuxuan Zhang, Jiaming Liu, Sirui Zhao, and Enhong Chen. Speculative sampling in interactive video generation. *arXiv preprint arXiv:2610.20556*, 2026. 3

[19] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21807–21818, 2024. 4

[20] Zilan Huang, Zhiqing Tang, Hanshuai Cui, Tian Wang, Yuan Wu, Weijia Jia, and Wei Zhao. RISE: Relay inference and online scheduling for efficient edge-device collaborative diffusion model services. *arXiv preprint arXiv:2606.17378*, 2026. 2

[21] Yicheng Ji, Zhizhou Zhong, Jun Zhang, Qin Yang, XiTai Jin, Ying Qin, Wenhan Luo, Shuiyang Mao, Wei Liu, and Huan Li. Forcing-KV: Hybrid KV cache compression for efficient autoregressive video diffusion models. *arXiv preprint arXiv:2605.09681*, 2026. 2

[22] Fan Jiang, Zhaoxu Sun, Mengchao Wang, Ziyu Zhu, Chiyu Wang, Yunpeng Zhang, Wenlin Liu, Yun Wang, Xue Zheng, Rui Sun, Junfeng Ni, Hongyu Pan, Zhongxu Sun, Fei Yu, Zengye Ge, Mengmeng Du, Nianfei Fan, Mingchao Sun, Yu Liu, Yongchang, Yanqing Zhu, Jiahang Wang, Ning Ying, Yuze Xuan, Di Yang, Zhicheng Liu, Zhe Gao, Tingbing Xu, Jiacheng Sui, Wenjin Yang, Junnan Lai, Shufeng Liu, Yuan Liu, Zheng Zhou, Yingliang Peng, Dawei Cao, Kaifeng Sheng, Yuxiang Cai, Fei Lu, Mu Xu, and Ning Guo. ABot-World-0: Infinite interactive world rollout on a single desktop GPU. *arXiv preprint arXiv:2607.19191*, 2026. 2

[23] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 4

[24] Seung Wook Kim, Yuhao Zhou, Jonah Philion, Antonio Torralba, and Sanja Fidler. Learning to simulate dynamic environments with GameGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[25] Kyungmin Lee, David Chu, Eduardo Cuervo, Johannes Kopf, Yury Degtyarev, Sergey Grizan, Alec Wolman, and Jason Flinn. Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, 2015. 3

[26] Jiaqi Li, Junshu Tang, Zhiyong Xu, Longhuang Wu, Yuan Zhou, Shuai Shao, Tianbao Yu, Zhiguo Cao, and Qinglin Lu. Hunyuan-GameCraft: High-dynamic interactive game video generation with hybrid history condition. *arXiv preprint arXiv:2506.17201*, 2025. 2

[27] Jiuming Liu, Chaojun Ni, Mengmeng Liu, Chensheng Peng, Fangjinhua Wang, et al. Towards interactive video world modeling: Frontiers, challenges, benchmarks, and future trends. *arXiv preprint arXiv:2606.01164*, 2026. 2

[28] Jiacheng Lu, Haoyi Zhu, Sipei Yi, Enze Xie, Yu Li, and Cheng Zhuo. Light interaction: Training-free inference acceleration for interactive video world models. *arXiv preprint arXiv:2605.31158*, 2026. 2

[29] Chengtao Lv, Yumeng Shi, Yushi Huang, Ruihao Gong, Shen Ren, and Wenya Wang. Light forcing: Accelerating autoregressive video diffusion via sparse attention. In *International Conference on Machine Learning (ICML)*, 2026. 2

[30] Jing Lyu, Shuanghao Bai, Runze Xiao, Zhenyu Liao, Wenxing Tan, Zihan Tang, Ruochuan Shi, Cheng Peng, Yuheng Ji, Yihao Wang, Badong Chen, Pengwei Wang, Zhongyuan Wang, and Xiaoguang Zhao. DIDO: Distilling interaction-centric dynamics into one-step denoising for world action models. *arXiv preprint arXiv:2609.15570*, 2026. 2

[31] Yuan Lyu, Bharath Irukulapati, and Jaya Prakash Champati. Speculation at a distance: Where edge-cloud speculative decoding actually pays off. *arXiv preprint arXiv:2606.25091*, 2026. 3

[32] Erwann Millon. Krea realtime 14b: Real-time video generation. https://github.com/krea-ai/realtime-video, 2025. 8

[33] Mustafa Munir, Huy Vu, Shreyas Misra, Rohit Jena, Sajad Norouzi, Ali Taghibakhshi, Anis Ahmad, Anjul Patney, Pavlo Molchanov, and Nima Tajbakhsh. Accelerating video diffusion via training-free trajectory routing. *arXiv preprint arXiv:2609.30096*, 2026. 2

[34] Zizheng Pan, Bohan Zhuang, De-An Huang, Weili Nie, Zhiding Yu, Chaowei Xiao, Jianfei Cai, and Anima Anandkumar. T-Stitch: Accelerating sampling in pre-trained diffusion models with trajectory stitching. In *International Conference on Learning Representations (ICLR)*, 2025. 2

[35] Jack Parker-Holder and Shlomi Fruchter. Genie 3: A new frontier for world models. Google DeepMind blog, https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/, 2025. 2

[36] Rajit Rajpal, Shahbuland Matiana, Liew Wei Pyn, Anmol Agarwal, Ryan Craig, Andrew Lapp, Mithun Hunsur, Sami BuGhanem, Scottie Fox, Aaron Sanders, Carson Poole, Irene Park, Dave Rossi, Spencer Frazier, and Louis Castricato. Waypoint-1.5: A real-time video world model for consumer hardware. *arXiv preprint arXiv:2609.37107*, 2026. 2

[37] Jibin Song, Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Flowblending: Stage-aware multi-model sampling for fast and high-fidelity video generation. *arXiv preprint arXiv:2512.24724*, 2025. 2

[38] Shangye Song, Dong Gong, Hong Jia, Yun Sing Koh, and Xinyu Zhang. CtrlCache: Accelerating interactive video world models with control-aware caching. *arXiv preprint arXiv:2610.08777*, 2026. 2

[39] Desen Sun, Jason Hon, Jintao Zhang, and Sihang Liu. HybridStitch: Token-adaptive model stitching for efficient diffusion generation. *arXiv preprint arXiv:2603.07815*, 2026. 2

[40] Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. WorldPlay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025. 2

[41] Pardis Taghavi, Tingyu Guo, Jonas Lossner, Gaurav Pandey, and Reza Langari. ActionSplice: In-flight action editing for interactive world models. *arXiv preprint arXiv:2609.08230*, 2026. 3

[42] Yuqing Tian, Zhaoyang Zhang, Yuzhi Yang, Zirui Chen, Zhaohui Yang, Richeng Jin, Tony Q. S. Quek, and Kai-Kit Wong. An edge-cloud collaboration framework for generative AI service provision with synergetic big cloud model and small edge models. *IEEE Network*, 38(5):37–46, 2024. 2

[43] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations (ICLR)*, 2025. 1, 2

[44] Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, et al. Matrix-Game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026. 2

[45] Haocheng Xi, Shuo Yang, Yilong Zhao, Muyang Li, Han Cai, Xingyang Li, Yujun Lin, Zhuoyang Zhang, Jintao Zhang, Xiuyu Li, Zhiying Xu, Jun Wu, Chenfeng Xu, Ion Stoica, Song Han, and Kurt Keutzer. Quant VideoGen: Auto-regressive long video generation via 2-bit KV-Cache quantization. In *International Conference on Machine Learning (ICML)*, 2026. 2

[46] Jiajian Xie, Shengyu Zhang, Zhou Zhao, Fan Wu, and Fei Wu. EC-Diff: Fast and high-quality edge-cloud collaborative inference for diffusion models. *arXiv preprint arXiv:2507.11980*, 2025. 1, 2, 4, 5

[47] Boxun Xu, Yuming Du, Zichang Liu, Siyu Yang, Ziyang Jiang, Siqi Yan, Rajasi Saha, Albert Pumarola, Wenchen Wang, and Peng Li. Sparse forcing: Native trainable sparse attention for real-time autoregressive diffusion video generation. *arXiv preprint arXiv:2604.21221*, 2026. 2

[48] Chenqian Yan, Songwei Liu, Hongjian Liu, Xurui Peng, Xiaojian Wang, Fangmin Chen, Lean Fu, and Xing Mei. Hybrid SD: Edge-cloud collaborative inference for stable diffusion models. *arXiv preprint arXiv:2408.06646*, 2024. 1, 2, 4, 5

[49] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 4

[50] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo 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 (CVPR)*, 2025. 1, 2, 3

[51] Kaining Ying, Hengrui Hu, Siyu Ren, Jiamu Li, Fengjiao Chen, Ziwen Wang, Xuezhi Cao, Xunliang Cai, and Henghui Ding. WBench: A comprehensive multi-turn benchmark for interactive video world model evaluation. *arXiv preprint arXiv:2605.25874*, 2026. 2, 4, 6

[52] Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. GameFactory: Creating new games with generative interactive videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11590–11599, 2025. 1

[53] Wangbo Yu, Kunhao Liu, Wenbo Hu, Shenghai Yuan, Chaoran Feng, et al. WorldCrafter: Consistent video world model with implicit 3D-aware memory. *arXiv preprint arXiv:2609.24984*, 2026. 2

[54] Jie Zhang, Xiaoyue Chen, Anzhe Chen, Dayiheng Liu, Deqing Li, et al. Qwen-RobotWorld technical report: Unifying embodied world modeling through language-conditioned video generation. *arXiv preprint arXiv:2606.17030*, 2026. 2

[55] Min Zhao, Hongzhou Zhu, Bokai Yan, Zihan Zhou, Yimin Chen, Honglie Wang, Wenqiang Sun, Zhengwei Fang, Zizheng Xun, Zihao Li, Kaiwen Zheng, Guande He, Xiao Yang, Chongxuan Li, Fan Bao, and Jun Zhu. minWM: A full-stack open-source framework for real-time interactive video world models. *arXiv preprint arXiv:2605.30263*, 2026. 2