Autoregressive Adversarial Post-Training for Real-Time Interactive Video Generation

Shanchuan Lin* **Ceyuan Yang** **Hao He†** **Jianwen Jiang‡**
Yuxi Ren **Xin Xia** **Yang Zhao** **Xuefeng Xiao** **Lu Jiang**
ByteDance Seed
https://seaweed-apt.com/2

Abstract

Existing large-scale video generation models are computationally intensive, preventing adoption in real-time and interactive applications. In this work, we propose autoregressive adversarial post-training (AAPT) to transform a pre-trained latent video diffusion model into a real-time, interactive video generator. Our model autoregressively generates a latent frame at a time using a single neural function evaluation (1NFE). The model can stream the result to the user in real time and receive interactive responses as controls to generate the next latent frame. Unlike existing approaches, our method explores adversarial training as an effective paradigm for autoregressive generation. This not only allows us to design an architecture that is more efficient for one-step generation while fully utilizing the KV cache, but also enables training the model in a student-forcing manner that proves to be effective in reducing error accumulation during long video generation. Our experiments demonstrate that our 8B model achieves real-time, 24fps, streaming video generation at 736×416 resolution on a single H100, or 1280×720 on 8×H100 up to a minute long (1440 frames).

In recent years, the field of visual content creation has been transformed by the rise of foundation models for video generation [4, 78, 69, 44, 95]. These models have enabled a wide range of powerful applications, including text-to-video generation, image-to-video synthesis, and controllable video creation conditioned on various multi-modal signals.

Building on this progress, researchers are beginning to explore more ambitious applications. One exciting direction is using video generation models as interactive game engines and world simulators [93, 6, 67, 4]. Unlike offline video synthesis, interactive video generation requires the model to respond to user inputs in real time and continuously generate coherent video as the world evolves.

While diffusion models produce high-quality videos, they are very expensive for real-time interactive video generation. Early approaches applied diffusion models frame-by-frame [93, 111]. However, these approaches incur high redundancy due to the need to reprocess the context frames at every frame generation step. To address this, diffusion forcing [7, 108, 40, 23] introduced progressive noise to parallelize denoising across frames. Recent work further reduced inference costs by incorporating causal attention, KV caching, and step distillation [117, 75], with the current best model [117] achieving four denoising steps.

Meanwhile, token-based autoregressive generation—popularized by large language models (LLMs) [5, 1, 19]—offers an alternative. Models like VideoPoet [43] treat video generation as a next-token prediction task, which can straightforwardly leverage KV caching to improve generation

*Shanchuan Lin: Corresponding author: peterlin@bytedance.com
†Hao He: The Chinese University of Hong Kong. Internship at ByteDance Seed.
‡Jianwen Jiang: ByteDance Intelligent Creation Lab.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

efficiency. However, per-token decoding remains sequential, limiting parallelism and making it difficult to meet real-time demands.

In this work, we aim to address the three core challenges of interactive video generation: (1) achieving real-time video generation throughput, (2) maintaining a low latency for interactive signals, and (3) enabling causal video generation of an extended duration. To this end, we explore adversarial training as a new paradigm and propose autoregressive adversarial post-training (AAPT) as an effective strategy for transforming a pretrained video diffusion transformer into a highly efficient autoregressive generator.

Our approach offers several advantages. First, it is fast. Our model autoregressively predicts each latent frame in a single forward pass (1NFE) while fully exploiting the KV cache. Our architecture design further enables 2× higher efficiency than equivalent diffusion-forcing models distilled to one step. Second, it maintains better quality over long durations. Our adversarial approach enables full student-forcing training, which mitigates error accumulation for long video generation. Furthermore, our student-forcing approach does not require paired ground-truth targets, allowing us to train long video generators and bypass the limitations of short-duration training data. This is important, as single continuous shots of tens of seconds are extremely rare in most datasets.

We demonstrate these benefits empirically. In terms of speed, our 8B-parameter model achieves real-time 24fps video generation at 736×416 resolution on a single H100 GPU, and 1280×720 resolution on 8×H100 GPUs, with a latency of only 0.16 seconds, substantially outperforming CausVid [117], a 5B model that operates at 640×352 9.4fps with a 1.30-second latency. In terms of duration, our model can generate continuous 60-second (1440-frame) video streams while fully utilizing the KV cache. This significantly exceeds the previous best one-step generator, APT [49], which supports only 49 frames.

Our experiments focus on the image-to-video (I2V) generation scenario, where the first frame is provided by the user, as most interactive applications adopt this setting. We showcase our method on two interactive applications—pose-conditioned virtual human generation and camera-controlled world exploration—where users can steer video generation in real time through interactive inputs. Evaluations show that our model achieves performance comparable to the state of the art.

1 Related Work

One-Step Video Generation Early video generation models [3, 81] using generative adversarial networks (GANs) [18] can achieve fast generation using a single network evaluation. However, the quality, duration, and resolution are poor by modern standards. Diffusion models [28, 84] are the current state-of-the-art, yet their iterative generation process is slow and expensive. Generating a few seconds of high-resolution videos can take minutes. Existing research has attempted to reduce the inference cost by proposing more efficient formulations [51, 55, 35], samplers [59, 60, 90], architecture [109, 121, 120, 119, 98, 17, 66], caching [63, 53, 125], and distillation, *etc.* In particular, step distillation [74, 83, 82, 58, 49, 48, 72, 116, 115, 77, 76, 97, 50, 55, 110, 8, 61, 56, 112, 42, 37] emerges as one of the most effective approaches and has been widely studied in the image domain and is also adopted in video models. Seaweed [78] and FastHunyuan [15] report that the generation of 5-second 1280×720 24fps videos can be distilled to 8 or 6 steps without much degradation in quality. For further reduction in steps, SF-V [123] and OSV [64] explore 2 seconds of 1024×768 7fps image-to-video generation using only a single step. Recently, APT [49] achieves real-time text-to-video generation of 2-second 1280×720 24fps videos on 8×H100 GPUs using a single step. This has inspired more downstream applications to explore one-step video generation [99, 11]. Our method extends adversarial post-training (APT) to the autoregressive video generation scenario.

Streaming Long-Video Generation Early research in streaming and long video generation [26, 41, 96] applies training-free or pipeline approaches on small-scale image and video generation models but is limited in quality. Modern large-scale video diffusion models, *e.g.* MovieGen [69], Hunyuan [44], Wan2.1 [95], and Seaweed [78], adopt transformer architecture and are trained on much higher resolutions and frame rates. However, due to the quadratic increase in attention computation, these models are commonly trained to only generate videos up to 5 seconds. To support long-video generation, these models are also trained on the video extension task, which gives the model the first few frames as a condition. At inference, this allows the model to extend the generation and stream the result to users as 5-second chunks. The extension can only be performed a few times before the

error accumulation catches up. Recent works have also explored architectures with linear complexity to directly generate long videos [98, 17, 66], but they are not designed for streaming applications.

More recently, diffusion forcing [7] has been proposed for video generation. It assigns progressive noise levels to frame chunks so the decoding proceeds in a causal streaming fashion. Earlier work uses bidirectional attention [108, 40]. Recent works have moved toward causal attention with KV cache [117, 9, 75, 23]. Most notably, SkyReel-2 [9] and MAGI-1 [75] are diffusion-forcing video generation models trained from scratch. CausVid [117] explores converting existing bidirectional video diffusion models to causal diffusion-forcing generators. Some of these methods also apply step distillation to improve speed. MAGI-1 [75] distills the model to 8 steps and outputs 24 frames as a chunk. It reports real-time 1280×720 24fps generation on 24×H100 GPUs. However, this amount of computation limits wide adoption. CausVid [117] distills the model to 4 steps and outputs 16 frames as a chunk. It can generate 640×320 videos at 9.4fps on a single H100 GPU. In comparison, our method is significantly faster. Our model uses only a single step and achieves 24fps streaming at 736×416 resolution on a single H100 GPU, or 1280×720 on 8×H100 GPUs. Moreover, ours generates a single latent frame (4 video frames) at a time to minimize latency.

It is important to note that these diffusion-forcing models are still only trained up to a fixed-duration window, *e.g.* 5 seconds. Early approaches without KV cache can run a sliding window, but this becomes an issue for KV cache because the receptive field grows indefinitely. Applying a sliding window and dropping out KV tokens can't help because the remaining tokens in the cache were computed in the past and still carry the receptive field. Naive extrapolation at inference leads to out-of-distribution behaviors. Therefore, methods like CausVid [117], SkyReel-V2 [9], and MAGI-1 [75] still need to apply the extension technique at inference by restarting and re-computing some overlapping context frames to generate long videos. Except that the diffusion forcing objective naturally supports input tokens with different noise levels, so the context frames can be given as clean latent frames at the beginning, with no additional training necessary. However, this is not ideal as it causes wait time on real-world streaming applications. In contrast, our method supports streaming generations of minute-long videos using KV cache without stopping and reprocessing.

LLMs for Video Generation Large language models (LLMs) [5, 1, 19] have widely adopted the causal transformer architecture [94] for autoregressive generation. Most notably, attention is masked to prevent attending to future tokens, the inputs are past predictions, and the output targets are shifted by one for predicting the next tokens. Recent research has shown that images and videos can also be generated in such an autoregressive fashion [87, 102, 106, 10]. Although causal generation with KV cache is computationally efficient, generating token-by-token prevents parallelization and is slow for high-resolution generation. Some research has explored the decoding of multiple tokens at once during inference [103, 71, 114], but there is a tradeoff for quality, and it is challenging to decode an entire frame at once. Our architecture is inspired by LLMs, but ours generates a frame of tokens at a time, trained using an adversarial objective. This is optimized for fast generation.

Interactive Video Generation Our paper showcases our model's real-time interactive generation ability on two applications: pose-controlled virtual human video generation and camera-controlled world exploration. We briefly introduce the related works in each subfield.

Recent research has explored the use of video generation models to create interactive environments for gameplay and world simulation [2, 6, 93, 22, 67, 16, 13]. Typically, the first frame is given, and the model continuously predicts the next frame given user control (image-to-world). The control can be the discrete states in an action space or general-purpose camera position embeddings [24, 25]. However, the high computation cost of the existing video generation approaches greatly limits the resolution and frame rates. For example, GameNGen [93] and MineWorld [22] only generate videos around 320×240 resolution at 6∼20fps with small models of a few hundred million parameters. Recent works, *e.g.* Genie-2 [67], Oasis [13], Matrix [16], *etc.*, have moved toward large-scale architectures and higher resolutions. Though many report their methods can operate in real-time, the specific hardware requirements are not specified.

Interactive video generation also holds significant potential in the domain of virtual humans. Typically, the first frame is given to establish the identity, then the pose [30, 62] or other multimodal [34, 46, 45, 89] conditions are given to drive the subject. Existing works employ diffusion models with the extension technique to generate long videos [85]. The inference speed remains a major bottleneck that limits their applicability to offline human video generation tasks.

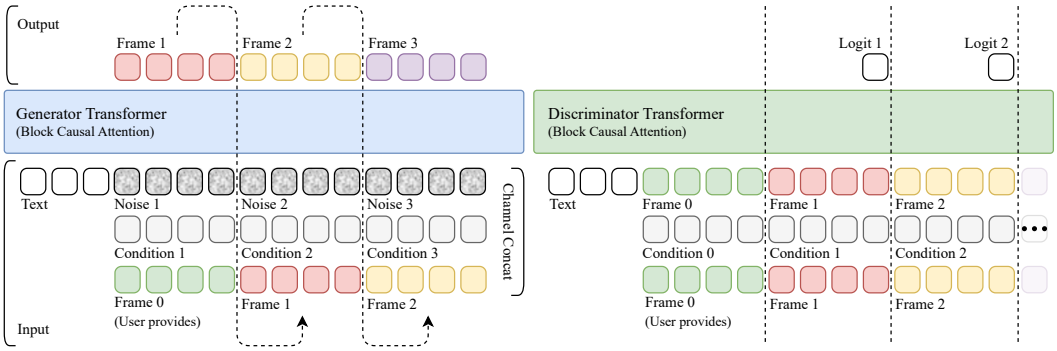

Figure 1: **Generator (left)** is a block causal transformer. The initial frame 0 is provided by the user at the first autoregressive step, along with text, condition, and noise as inputs to the model to generate the next frame in a single forward pass. Then, the generated frame is recycled as input, along with new conditions and noise, to recursively generate further frames. KV cache is used to avoid recomputation of past tokens. A sliding window is used to ensure constant speed and memory for the generation of arbitrary lengths. **Discriminator (right)** uses the same block causal architecture. Condition inputs are shifted to align with the frame inputs. Since it is initialized from the diffusion weights, we replace the noise channels with frame inputs following APT.

2 Method

Our objective is to transform a pre-trained video diffusion model into a fast, per-latent-frame causal generator suitable for real-time interactive applications. We achieve this through a new method called autoregressive adversarial post-training (AAPT). This section discusses AAPT's architectural transformations and training procedures.

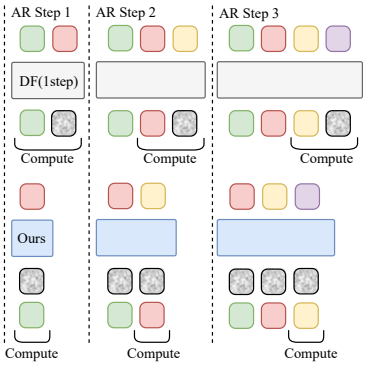

Figure 2: Ours is more efficient than one-step diffusion forcing (DF).

2.1 Causal Architecture

We build our method on a pre-trained video diffusion model that employs a diffusion transformer (DiT) [68] architecture and operates in a spatially and temporally compressed latent space through a 3D variational autoencoder (VAE) [118]. Since our model operates in the latent space, we will refer to latent frames simply as frames unless otherwise specified. Our diffusion transformer has 8 billion (8B) parameters. It takes text embedding tokens, noisy visual tokens, and diffusion timesteps as input, and calculates bidirectional full attention over all the text and video tokens.

First of all, we transform the bidirectional DiT into a causal autoregressive architecture by replacing full attention with block causal attention. Specifically, text tokens only attend to themselves, and visual tokens attend to text tokens and visual tokens of previous and current frames. Afterward, we change the model inputs. As illustrated in Fig. 1, in addition to the regular noise and conditional inputs used by the original diffusion model, we change the model to also take in the past generated frame from the previous autoregressive step through channel concatenation, except the first autoregressive step where the input frame given by the user is used instead. During inference, our model runs autoregressively. At each autoregressive step, it reuses the attention KV cache and generates the next frame in a single forward pass. The generated frame is recycled, along with a new control condition, as inputs for the next autoregressive step.

To prevent the unbounded growth of attention computation and KV cache size, visual tokens attend to at most N past frames while always attending to the text tokens and the first frame. It is worth noting that although each attention layer uses a window size of N, stacking multiple layers results in a much larger effective receptive field.

Our architecture resembles that of large language models (LLMs), but with one important distinction: unlike conventional next-token prediction that outputs the token probabilities using a softmax layer,

our model generates all tokens for the next frame in a single forward pass sampled by noise. In addition, our input recycling approach is also more efficient than the one-step diffusion forcing, as shown in Fig. 2. Diffusion forcing is not optimized for the one-step generation scenario. When using KV cache, diffusion forcing requires computation on two frames on every autoregressive step, while ours only needs one.

2.2 Training Procedure

To create a one-step, per-frame, autoregressive generator, our training process involves three sequential stages: (1) diffusion adaptation, (2) consistency distillation, and (3) adversarial training.

Diffusion adaptation We load the pre-trained weights and finetune the model with the diffusion objective for architectural adaptation. We apply teacher-forcing training, where the ground-truth frames from the dataset are given as past-frame inputs. The output target is shifted by one frame to let the model perform next-frame prediction. Instead of pure noise, the noisy latent and the diffusion timestep $t \sim \mathcal{U}(0, T)$ are still used per regular diffusion training. The same noise level is applied for all frames. This resembles LLMs training, where all the autoregressive steps are trained in parallel.

Consistency distillation We apply consistency distillation [83] before adversarial training as an initialization step to accelerate convergence following APT [49]. Our modified formulation and architecture are fully compatible with the original consistency distillation process without the need for modification. We omit classifier-free guidance (CFG) as we find it introduces artifacts in our autoregressive generation setting.

Adversarial training We extend APT [49] to the autoregressive setting with improved discriminator design, training strategy, and loss objective.

For the discriminator model, we use the same causal generator architecture as our discriminator backbone, initialize it from the diffusion weights post-adaptation, and insert logit output projection layers. We replace the noise input to frames and randomly sample timestep $t \sim \mathcal{U}(0, T)$ for fast adaptation. A notable difference to APT discriminator design is that ours computes output logit for every frame instead of for the whole clip. This design naturally enables parallel multi-duration discrimination, as inspired by multi-resolution discrimination [39, 38].

We find models trained with teacher-forcing incur significant error accumulation at inference. To address this, we introduce a student-forcing approach within the adversarial training framework. Specifically, the generator only uses the ground-truth first frame and recycles the actual generated results as input for the next autoregressive step. In each training step, the generator is autoregressively invoked with KV cache to produce the video, exactly matching the inference behavior, while the discriminator evaluates all the generated frames in a single forward pass in parallel. We find detaching the pass-frame input from the gradient graph improves stability. We allow the gradient to flow through the KV cache to update all the parameters.

For the loss, we use R3GAN [31] objective as our preliminary experiments find that it is more stable than the non-saturating loss [18]. Specifically, we adopt the relativistic loss [36] and apply both the approximated R1 and R2 regularizations [73, 65] as proposed in APT [49].

Long-Video Training For the model to learn continuous generation of long videos, one must train it on single-shot videos of long duration (*e.g.*, 30–60 seconds). However, such long single-shot videos are rare in most training datasets, where the average shot duration is only 8 seconds. The lack of long-duration training leads to poor temporal extrapolation during inference.

To address the data limitation, we let the generator produce a long video, *e.g.* 60 seconds, and break it down into short segments, *e.g.* 10 seconds, for discriminator evaluation. We keep an overlapping 1-second duration for discriminator evaluation to encourage segment continuation. The discriminator is trained on generated segments and real videos from the dataset. This objective ensures that every segment of a generated long video fits the data distribution.

To fit the GPU memory, we also let the generator only produce a segment at a time to be evaluated by the discriminator. To produce the next segment, the generator reuses the detached KV cache from the last segment. The gradient is backpropagated after every segment evaluation for loss accumulation.

This technique can be used to train very long generators, with the trade-off of an increase in training time. We find this technique significantly improves the quality of long-duration video generation. This is made possible by the discriminator in adversarial training. Unlike supervised objectives that require ground-truth targets, the discriminator does not need explicit supervision for each input frame. Instead, it learns to distinguish real videos from generated ones. As a result, the model can learn from every video sample, rather than relying on a limited number of long-duration videos.

2.3 Interactive Generation Applications

We first train a model for the general image-to-video generation task without interactive conditions. This allows us to evaluate the generation quality on standard benchmarks. We then train two separate models on the pose-conditioned human generation task and the camera-conditioned world exploration task. This allows us to evaluate the controllability using two distinct condition signals. For the pose-conditioned human video generation task, we extract and encode the human pose from the training videos and provide it as a per-frame condition to the model following [46]. Similarly, for the camera-conditioned world exploration generation task, we follow [25] to extract and encode the camera origin and orientation as Plücker embeddings, with a few modifications to have it better support causal generation. We use similar training datasets as used in these prior works [46, 25]. We refer readers to our supplementary materials for additional details on our architecture, implementation, and training parameters.

3 Evaluation

Experimental Setups We use causal 3D convolution VAE [118] to compress the video temporally by 4 and spatially by 8. Therefore, our model autoregressively generates 4 video frames. The first input frame is independently compressed as a latent frame by the VAE. Since our VAE is causal, it naturally supports streaming decoding. We use attention window size $N = 30$ to attend to 30 latent frames (5 seconds). Additional details on the training setup are provided in the supplementary materials.

Baseline and Metrics Following prior work [117], we evaluate our method on the standard VBench-I2V benchmark [32] on both 120-frame short-video generation and 1440-frame long-video generation. For comparison, we select CausVid [117], Wan2.1 [95], Hunyuan [44], MAGI-1 [75], SkyReel-V2, and our own diffusion model as baseline. These models are selected because CausVid is the state-of-the-art for fast streaming generation, and other models are available open-source video generation foundational models that support I2V. Note, CausVid is a closed-source model and only reports VBench-I2V for 120-frame 12fps generation. Wan2.1 and Hunyuan are bidirectional diffusion models that only support up to 120-frame generation. MAGI-1 and SkyReel-V2 are diffusion-forcing models that support arbitrary-length streaming decoding, so we include them for the 1440-frame comparison. Our model is evaluated and compared at 736×416 resolution. Additional inference settings and 1280×720 results are provided in the supplementary materials.

Main Results Figure 3 qualitatively compares our method on one-minute (1440-frame) video generation against SkyReel-V2, MAGI-1, and our diffusion baseline. All three of them exhibit strong error accumulation after 20 to 30 seconds. For our diffusion baseline, we experiment using a lower CFG scale or using rescale [47] but it does not mitigate the exposure problem and can further cause more structural deformation, so we keep it at CFG 10. We also show that our AAPT model trained on only a 10-second duration cannot generalize to long videos in Fig. 3d. Long video training is critical, as shown in Fig. 3e. Figure 4 shows more results of our model across subjects and scenes.

Table 1 shows that our method achieves competitive performance compared to the state-of-the-art methods on the quantitative metrics. For 120-frame I2V generation, AAPT improves frame quality score and image conditioning scores compared to the diffusion baseline and is the best across all compared methods. The frame quality improvement concurs with the findings in APT [49] that adversarial training can improve visual quality. AAPT has resulted in a slight decline in temporal quality score compared to the diffusion baseline, but is still above Wan and closely follows Hunyuan. We note that CausVid has an exceptionally high temporal quality score, likely because it was trained on 12fps data, which usually results in a higher dynamic degree than other 24fps models, and the dynamic degree score is the main differentiator for the overall temporal quality. For 1440-frame

Figure 3: Qualitative comparison on one-minute, 1440-frame, VBench-I2V generation.

Figure 4: More results of our AAPT model for one-minute, 1440-frame, VBench-I2V generation.

I2V generation, AAPT achieves the best quality scores across the comparison and has improved conditioning scores compared to the diffusion baseline. We note that SkyReel-V2 and MAGI-1 have a higher image-conditioning score compared to our AAPT and diffusion baseline which is because most of the videos by MAGI-1 are stationary. This is reflected in its much lower dynamic degree score and the qualitative visualization in Fig. 3.

Table 1: Quantitative comparisons on VBench-I2V [32]. * denotes metrics that need special interpretation as discussed in the main text. The 6 quality metrics are aggregated as temporal quality and frame quality according to VBench-Competition. The best metrics are highlighted in bold.

Frames	Method	Quality								Condition	
		Temporal Quality	**Frame Quality**	Subject Consistency	Background Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	**I2V Subject**	**I2V Background**
120	CausVid [117]	***92.00**	65.00	Not Reported							
	Wan 2.1 [95]	87.95	66.58	93.85	96.59	97.82	39.11	63.56	69.59	96.82	98.57
	Hunyuan [44]	89.80	64.18	93.06	95.29	98.53	54.80	60.58	67.78	97.71	97.97
	Ours (Diffusion)	90.40	66.08	94.58	96.76	98.80	52.52	62.44	69.71	97.89	99.14
	Ours (AAPT)	89.51	**66.58**	96.22	96.66	99.19	42.44	62.09	71.06	**98.60**	**99.36**
1440	SkyReel-V2 [9]	82.19	53.67	78.43	86.38	99.28	47.15	53.68	53.65	96.50	98.07
	MAGI-1 [75]	80.79	60.01	82.23	89.27	98.54	25.45	52.26	67.75	***96.90**	***98.13**
	Ours (Diffusion)	86.65	60.49	82.38	89.48	98.29	66.26	56.46	64.51	95.01	97.72
	Ours (AAPT)	**89.79**	**62.16**	87.15	89.74	99.11	76.50	56.77	67.55	96.11	97.52

Table 2: Quantitative comparison on pose-conditioned human video generation task. Metrics better than ours are highlighted in bold.

Method	AKD↓	IQA↑	ASE↑	FID↓	FVD↓
DisCo	9.313	3.707	2.396	57.12	64.52
AnimateAnyone	5.747	3.843	2.718	26.87	37.67
MimicMotion	8.536	3.977	2.842	23.43	22.97
CyberHost	3.123	**4.087**	2.967	**20.04**	**7.72**
OmniHuman-1	**2.136**	**4.111**	**2.986**	**19.50**	**7.32**
Ours (AAPT)	**2.740**	**4.077**	**2.973**	**22.43**	**11.78**

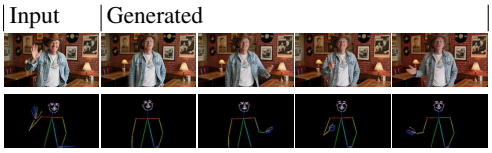

Figure 5: Pose-conditioned virtual human

Table 3: Quantitative comparison on camera-conditioned world exploration task. Metrics better than ours are highlighted in bold.

Method	FVD↓	Mov↑	Trans↓	Rot↓	Geo↑	Apr↑
MotionCtrl	221.23	102.21	0.3221	2.78	57.87	0.7431
CameraCtrl	199.53	133.37	0.2812	2.81	52.12	0.7784
CameraCtrl2	73.11	**698.51**	0.1527	**1.58**	**88.70**	0.8893
Ours (AAPT)	**61.33**	**521.23**	**0.1185**	**1.63**	**81.25**	**0.9012**

Figure 6: Camera-controlled world exploration

Pose-Conditioned Human Video Generation We evaluate our method on post-conditioned human video generation using the protocol and test set from previous work [45]. The pose accuracy is assessed via average keypoint distance (AKD) with keypoints extracted using DWPose [113]. For visual quality, we use Q-Align [107], a vision-language model, to evaluate image quality (IQA) and aesthetics (ASE). Additionally, Fréchet Inception Distance (FID) [27] and Frechet Video Distance (FVD) [91] measure the distributional alignment between the generated and the ground-truth samples. For comparison, we include four recent UNet-based diffusion models, *i.e.* Disco [101], AnimateAnyone [30], MimicMotion [122], CyberHost [45], and the state-of-the-art DiT-based OmniHuman-1 [46]. Table 2 presents the quantitative metrics. Among the six compared methods, ours is strong in pose accuracy and is ranked second only after the state-of-the-art baseline OmniHuman-1. In terms of visual quality, ours is consistently ranked second or third and is closely after CyberHost. Figure 5 shows visualization of our method.

Camera-Conditioned World Exploration We verify our method on the camera-conditioned world exploration task, also following the protocol of previous work [25]. We compute the FVD, movement strength (Mov), translational error (Trans), rotational error (Rot), geometric consistency (Geo), and appearance consistency (Apr). The details are provided in the supplementary materials. We compare against previous state-of-the-arts, *i.e.* MotionCtrl [104], can CameraCtrl 1 & 2 [24, 25]. Table 3 shows that our method achieves new state-of-the-art in three out of six metrics and closely follows CameraCtrl2 for the rest.

Inference speed We compare the throughput and latency of our method to other streaming video generation methods in Tab. 4. Our method is significantly faster while achieving performance comparable to the state of the arts.

Table 4: Latency and throughput comparison.

Method	Params	H100	Resolution	NFE	Latency	FPS
CausVid	5B	1×	640×352	4	1.30s	9.4
Ours	**8B**	**1×**	**736×416**	**1**	**0.16s**	**24.8**
MAGI-1	24B	8×	736 ×416	8	7.00s	3.43
SkyReelV2	14B	8×	960×544	60	4.50s	0.89
Ours	**8B**	**8×**	**1280×720**	**1**	**0.17s**	**24.2**

4 Ablation Studies

Long Video Training Table 5 reports VBench-I2V metrics on models trained with different durations for one-minute video generation. Specifically, the model trained for 60s significantly outperforms the model trained for only 10s, showing the effectiveness of long video training. Visualization is provided in Fig. 3d.

Table 5: One-minute generation performance using different training durations.

Training Duration	Temporal Quality	Frame Quality
10s	85.86	57.92
20s	85.60	65.69
60s	89.79	62.16

Teacher-Forcing and Student-Forcing Although diffusion adaptation and consistency distillation only support teacher forcing, adversarial training can be done in either teacher-forcing or student-forcing fashion. We describe the setup in the supplementary materials.

We find that models trained with teacher-forcing adversarial objective fail to generate proper videos at inference time, as shown in Figure 7. The content starts to drift significantly only a few frames into the generation process. Student-forcing training is critical in mitigating error accumulation. Although prior work has found that adding Gaussian noise to the input at training can reduce drifting at inference [93], it does not resolve the distribution gap from a fundamental level as student-forcing training. We leave additional explorations to future works.

Figure 7: Models trained with teacher-forcing adversarial objective fail to generate proper content at inference.

Limitations For consistency, our model can have difficulty maintaining the subject and the scene. This is caused by both the generator and the discriminator. Our generator adopts the basic sliding window for simplicity. We leave the exploration of more architectures and optimizations [105, 80, 52, 21, 20] to future works. For the discriminator, current segment-based discrimination cannot enforce long-range consistency. This may be mitigated by adding identity embeddings [54] to the discriminator. For training speed, the long video training process can be slow. For quality, we find that one-step generation can still create defects, and once the defects emerge, they can be kept in the scene for a long time since the discriminator also enforces temporal consistency. More research is needed to improve the quality of one-step generation. For the duration, we test our model on zero-shot five-minute generation. Our model can still generate content but with artifacts. We provide examples in the supplementary material.

5 Conclusion

We have introduced autoregressive adversarial post-training (AAPT), a method that uses adversarial training as a paradigm to transform video diffusion models into a fast autoregressive generator suitable for real-time interactive applications. Our model achieves performance comparable to that of the best methods while being significantly more efficient. We also analyze its limitations and aim to address them in future work.

Acknowledgment and Disclosure of Funding

We thank Weihao Ye for assistance with the evaluation. We thank Zuquan Song and Junru Zheng for assistance with the computing infrastructure. We thank Jianyi Wang and Zhijie Lin for their discussions during the work. This work is fully funded by ByteDance Seed.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

[3] Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei Efros, and Tero Karras. Generating long videos of dynamic scenes. *Advances in Neural Information Processing Systems*, 35:31769–31781, 2022.

[4] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1:8, 2024.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[7] 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.

[8] Dar-Yen Chen, Hmrishav Bandyopadhyay, Kai Zou, and Yi-Zhe Song. Nitrofusion: High-fidelity single-step diffusion through dynamic adversarial training. *arXiv preprint arXiv:2412.02030*, 2024.

[9] Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Juncheng Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengchen Ma, et al. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025.

[10] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

[11] Zheng Chen, Zichen Zou, Kewei Zhang, Xiongfei Su, Xin Yuan, Yong Guo, and Yulun Zhang. Dove: Efficient one-step diffusion model for real-world video super-resolution. *arXiv preprint arXiv:2505.16239*, 2025.

[12] Suhwan Cho, Minhyeok Lee, Seunghoon Lee, Chaewon Park, Donghyeong Kim, and Sangyoun Lee. Treating motion as option to reduce motion dependency in unsupervised video object segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5140–5149, 2023.

[13] Etched Decart, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. *URL: https://oasis-model. github. io*, 2024.

[14] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[15] FastHunyuan. `https://huggingface.co/FastVideo/FastHunyuan`.

[16] Ruili Feng, Han Zhang, Zhantao Yang, Jie Xiao, Zhilei Shu, Zhiheng Liu, Andy Zheng, Yukun Huang, Yu Liu, and Hongyang Zhang. The matrix: Infinite-horizon world generation with real-time moving control. *arXiv preprint arXiv:2412.03568*, 2024.

[17] Yu Gao, Jiancheng Huang, Xiaopeng Sun, Zequn Jie, Yujie Zhong, and Lin Ma. Matten: Video generation with mamba-attention. *arXiv preprint arXiv:2405.03025*, 2024.

[18] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[19] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[20] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[21] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[22] Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. Mineworld: a real-time and open-source interactive world model on minecraft. *arXiv preprint arXiv:2504.08388*, 2025.

[23] Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and Lu Jiang. Long context tuning for video generation. *arXiv preprint arXiv:2503.10589*, 2025.

[24] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

[25] Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. *arXiv preprint arXiv:2503.10592*, 2025.

[26] 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. *arXiv preprint arXiv:2403.14773*, 2024.

[27] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[29] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[30] Li Hu. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8153–8163, 2024.

[31] Nick Huang, Aaron Gokaslan, Volodymyr Kuleshov, and James Tompkin. The gan is dead; long live the gan! a modern gan baseline. *Advances in Neural Information Processing Systems*, 37:44177–44215, 2024.

[32] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[33] 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.

[34] Jianwen Jiang, Chao Liang, Jiaqi Yang, Gaojie Lin, Tianyun Zhong, and Yanbo Zheng. Loopy: Taming audio-driven portrait avatar with long-term motion dependency. In *The Thirteenth International Conference on Learning Representations*, 2025.

[35] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

[36] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

[37] Minguk Kang, Richard Zhang, Connelly Barnes, Sylvain Paris, Suha Kwak, Jaesik Park, Eli Shechtman, Jun-Yan Zhu, and Taesung Park. Distilling diffusion models into conditional gans. In *European Conference on Computer Vision*, pages 428–447. Springer, 2024.

[38] Animesh Karnewar and Oliver Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7799–7808, 2020.

[39] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[40] Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. *arXiv preprint arXiv:2405.11473*, 2024.

[41] Akio Kodaira, Chenfeng Xu, Toshiki Hazama, Takanori Yoshimoto, Kohei Ohno, Shogo Mitsuhori, Soichi Sugano, Hanying Cho, Zhijian Liu, and Kurt Keutzer. Streamdiffusion: A pipeline-level solution for real-time interactive generation. *arXiv preprint arXiv:2312.12491*, 2023.

[42] Jonas Kohler, Albert Pumarola, Edgar Schönfeld, Artsiom Sanakoyeu, Roshan Sumbaly, Peter Vajda, and Ali Thabet. Imagine flash: Accelerating emu diffusion models with backward distillation. *arXiv preprint arXiv:2405.05224*, 2024.

[43] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

[44] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[45] Gaojie Lin, Jianwen Jiang, Chao Liang, Tianyun Zhong, Jiaqi Yang, Zerong Zheng, and Yanbo Zheng. Cyberhost: A one-stage diffusion framework for audio-driven talking body generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[46] Gaojie Lin, Jianwen Jiang, Jiaqi Yang, Zerong Zheng, and Chao Liang. Omnihuman-1: Rethinking the scaling-up of one-stage conditioned human animation models. *arXiv preprint arXiv:2502.01061*, 2025.

[47] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5404–5411, 2024.

[48] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

[49] Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*, 2025.

[50] Shanchuan Lin and Xiao Yang. Animatediff-lightning: Cross-model diffusion distillation. *arXiv preprint arXiv:2403.12706*, 2024.

[51] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[52] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[53] Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. *arXiv preprint arXiv:2411.19108*, 2024.

[54] Lijie Liu, Tianxiang Ma, Bingchuan Li, Zhuowei Chen, Jiawei Liu, Qian He, and Xinglong Wu. Phantom: Subject-consistent video generation via cross-modal alignment. *arXiv preprint arXiv:2502.11079*, 2025.

[55] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[56] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.

[57] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[58] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.

[59] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[60] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

[61] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36:76525–76546, 2023.

[62] Yuxuan Luo, Zhengkun Rong, Lizhen Wang, Longhao Zhang, Tianshu Hu, and Yongming Zhu. Dreamactor-m1: Holistic, expressive and robust human image animation with hybrid guidance. *arXiv preprint arXiv:2504.01724*, 2025.

[63] Xinyin Ma, Gongfan Fang, Michael Bi Mi, and Xinchao Wang. Learning-to-cache: Accelerating diffusion transformer via layer caching. *Advances in Neural Information Processing Systems*, 37:133282–133304, 2024.

[64] Xiaofeng Mao, Zhengkai Jiang, Fu-Yun Wang, Jiangning Zhang, Hao Chen, Mingmin Chi, Yabiao Wang, and Wenhan Luo. Osv: One step is enough for high-quality image to video generation. *arXiv preprint arXiv:2409.11367*, 2024.

[65] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[66] Shentong Mo and Yapeng Tian. Scaling diffusion mamba with bidirectional ssms for efficient image and video generation. *arXiv preprint arXiv:2405.15881*, 2024.

[67] Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model. 2024.

[68] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[69] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

[70] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[71] Shuhuai Ren, Shuming Ma, Xu Sun, and Furu Wei. Next block prediction: Video generation via semi-auto-regressive modeling. *arXiv preprint arXiv:2502.07737*, 2025.

[72] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. *arXiv preprint arXiv:2404.13686*, 2024.

[73] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. *Advances in neural information processing systems*, 30, 2017.

[74] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[75] Sand-AI. Magi-1: Autoregressive video generation at scale, 2025.

[76] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[77] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024.

[78] Team Seawead, Ceyuan Yang, Zhijie Lin, Yang Zhao, Shanchuan Lin, Zhibei Ma, Haoyuan Guo, Hao Chen, Lu Qi, Sen Wang, et al. Seaweed-7b: Cost-effective training of video generation foundation model. *arXiv preprint arXiv:2504.08685*, 2025.

[79] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing Systems*, 37:68658–68685, 2024.

[80] Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

[81] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3626–3636, 2022.

[82] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

[83] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

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

[85] Michał Stypułkowski, Konstantinos Vougioukas, Sen He, Maciej Zięba, Stavros Petridis, and Maja Pantic. Diffused heads: Diffusion models beat gans on talking-face generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5091–5100, 2024.

[86] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[87] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[88] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[89] Linrui Tian, Qi Wang, Bang Zhang, and Liefeng Bo. Emo: Emote portrait alive generating expressive portrait videos with audio2video diffusion model under weak conditions. In *European Conference on Computer Vision*, pages 244–260. Springer, 2024.

[90] Ye Tian, Xin Xia, Yuxi Ren, Shanchuan Lin, Xing Wang, Xuefeng Xiao, Yunhai Tong, Ling Yang, and Bin Cui. Training-free diffusion acceleration with bottleneck sampling. *arXiv preprint arXiv:2503.18940*, 2025.

[91] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation.

[92] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[93] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

[94] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[95] Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[96] Fu-Yun Wang, Wenshuo Chen, Guanglu Song, Han-Jia Ye, Yu Liu, and Hongsheng Li. Gen-l-video: Multi-text to long video generation via temporal co-denoising. *arXiv preprint arXiv:2305.18264*, 2023.

[97] Fu-Yun Wang, Zhaoyang Huang, Weikang Bian, Xiaoyu Shi, Keqiang Sun, Guanglu Song, Yu Liu, and Hongsheng Li. Animatelcm: Computation-efficient personalized style video generation without personalized video data. In *SIGGRAPH Asia 2024 Technical Communications*, pages 1–5. 2024.

[98] Hongjie Wang, Chih-Yao Ma, Yen-Cheng Liu, Ji Hou, Tao Xu, Jialiang Wang, Felix Juefei-Xu, Yaqiao Luo, Peizhao Zhang, Tingbo Hou, et al. Lingen: Towards high-resolution minute-length text-to-video generation with linear computational complexity. *arXiv preprint arXiv:2412.09856*, 2024.

[99] Jianyi Wang, Shanchuan Lin, Zhijie Lin, Yuxi Ren, Meng Wei, Zongsheng Yue, Shangchen Zhou, Hao Chen, Yang Zhao, Ceyuan Yang, et al. Seedvr2: One-step video restoration via diffusion adversarial post-training. *arXiv preprint arXiv:2506.05301*, 2025.

[100] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21686–21697, 2024.

[101] Tan Wang, Linjie Li, Kevin Lin, Yuanhao Zhai, Chung-Ching Lin, Zhengyuan Yang, Hanwang Zhang, Zicheng Liu, and Lijuan Wang. Disco: Disentangled control for realistic human dance generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9326–9336, 2024.

[102] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[103] Yuqing Wang, Shuhuai Ren, Zhijie Lin, Yujin Han, Haoyuan Guo, Zhenheng Yang, Difan Zou, Jiashi Feng, and Xihui Liu. Parallelized autoregressive visual generation. *arXiv preprint arXiv:2412.15119*, 2024.

[104] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[105] Jeffrey Willette, Heejun Lee, Youngwan Lee, Myeongjae Jeon, and Sung Ju Hwang. Training-free exponential context extension via cascading kv cache. *arXiv preprint arXiv:2406.17808*, 2024.

[106] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024.

[107] Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090*, 2023.

[108] Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models. *arXiv preprint arXiv:2410.08151*, 2024.

[109] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

[110] Yanwu Xu, Yang Zhao, Zhisheng Xiao, and Tingbo Hou. Ufogen: You forward once large scale text-to-image generation via diffusion gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8196–8206, 2024.

[111] Hanshu Yan, Jun Hao Liew, Long Mai, Shanchuan Lin, and Jiashi Feng. Magicprop: Diffusion-based video editing via motion-aware appearance propagation. *arXiv preprint arXiv:2309.00908*, 2023.

[112] Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, Qiang Liu, and Jiashi Feng. Perflow: Piecewise rectified flow as universal plug-and-play accelerator. *arXiv preprint arXiv:2405.07510*, 2024.

[113] Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. Effective whole-body pose estimation with two-stages distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4210–4220, 2023.

[114] Yang Ye, Junliang Guo, Haoyu Wu, Tianyu He, Tim Pearce, Tabish Rashid, Katja Hofmann, and Jiang Bian. Fast autoregressive video generation with diagonal decoding. *arXiv preprint arXiv:2503.14070*, 2025.

[115] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.

[116] 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.

[117] Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast causal video generators. *arXiv preprint arXiv:2412.07772*, 2024.

[118] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

[119] Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. Sageattention2 technical report: Accurate 4 bit attention for plug-and-play inference acceleration. *arXiv preprint arXiv:2411.10958*, 2024.

[120] Jintao Zhang, Haofeng Huang, Pengle Zhang, Jun Zhu, Jianfei Chen, et al. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. *arXiv preprint arXiv:2410.02367*, 2024.

[121] Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhenghong Liu, and Hao Zhang. Fast video generation with sliding tile attention. *arXiv preprint arXiv:2502.04507*, 2025.

[122] Yuang Zhang, Jiaxi Gu, Li-Wen Wang, Han Wang, Junqi Cheng, Yuefeng Zhu, and Fangyuan Zou. Mimicmotion: High-quality human motion video generation with confidence-aware pose guidance. *arXiv preprint arXiv:2406.19680*, 2024.

[123] Zhixing Zhang, Yanyu Li, Yushu Wu, Anil Kag, Ivan Skorokhodov, Willi Menapace, Aliaksandr Siarohin, Junli Cao, Dimitris Metaxas, Sergey Tulyakov, et al. Sf-v: Single forward video generation model. *Advances in Neural Information Processing Systems*, 37:103599–103618, 2024.

[124] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

[125] Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. *arXiv preprint arXiv:2410.05317*, 2024.

A Model Architecture

Diffusion Transformer Our diffusion transformer largely follows the MMDiT design [14]. It has 8B parameters and 36 transformer blocks. The discriminator adopts the same architecture. Therefore, our generator and our discriminator consist of 16B parameters for the adversarial training.

Block Causal Attention We implement block causal attention using Flash Attention 3 [79] in a for-loop. We find it to provide reasonable performance for training. We leave the exploration for more performance implementation to future work. For inference, recurrent autoregressive steps are taken, and Flash Attention 3 can be naturally adopted without a performance penalty.

Positional Embedding As the duration of the generation becomes agnostic to our causal architecture, we modify the 3D rotary positional embeddings (RoPE) [86]. Specifically, the positional embeddings continue to stretch dynamically along the spatial dimension to help the model generalize to different resolutions, while the positional embeddings are changed to have a fixed interval along the temporal dimension to support arbitrary lengths of training and generation.

Parallelism We adopt FSDP [124] for data parallelism. We use ZERO 2 for the generator during student-forcing training that requires recurrent forward calls to avoid repeated parameter gathering, and ZERO 3 for all other modules to save memory. We also adopt Ulysses [33] as our context parallel strategy. We shard each video sample across 8 GPUs. Gradient checkpointing is also utilized per transformer block to fit the memory requirement.

B Training Details

Diffusion Adaptation After changing the architecture to block causal attention and adding the recycled input channels, we first adapt the model with diffusion training.

We follow the original model to use the flow-matching parameterization [51]. Specifically, given sample x_0 and noise ϵ, input is derived through linear interpolation $x_t = (1-t)\cdot x_0 + t\cdot\epsilon$. The diffusion timestep is sampled uniformly $t \sim \mathcal{U}(0,1)$, then passed through a shifting function $\text{shift}(t,s) := (s\times t)/(1+(s-1)\times t)$, where $s = 24$. Note that the same timestep is used for the entire clip without the diffusion-forcing [7] approach of assigning independent timesteps for each frame. Our model predicts the velocity $v = \epsilon - x_0$ and is penalized with the mean squared error loss. We apply the teacher-forcing paradigm and provide the ground-truth frames without noise as recycled input. The noisy input and the output target are shifted by one frame to facilitate next frame prediction.

We use AdamW optimizer [57] with a learning rate of 1e-5 and a weight decay scale of 0.01 throughout the process. We first train on 736×416 (equivalent to 640×480 by area) 5-second videos for 20k iterations with a batch size of 256. Then, we add 1280×720 to the mix for another 6k iterations with a batch size of 128. Finally, we turn up the maximum duration of 736×416 resolution videos to 15 seconds for 4k iterations with a batch size of 32. This curriculum allows our model to see enough samples in the early stages and see longer samples in the final stage.

Consistency Distillation Then we apply consistency distillation [82] to create a one-step generator. Although the results after consistency distillation are blurry, it provides a better initialization for the adversarial training stage, as discovered by APT [49].

We inherit the same AdamW settings and the dataset settings as in the last diffusion adaptation stage. We distill the model on 32 fixed steps, which are uniformly selected and then passed through the shifting function with a shifting factor $s = 24$. We do not apply classifier-free guidance [29]. We continue to use the teacher-forcing paradigm to provide ground-truth frames as recycled inputs, and shift the noisy inputs and output targets by one frame following the diffusion adaptation approach. We follow the improved consistency distillation technique [82] and do not apply exponential moving average on the consistency target. No additional modification is needed for consistency distillation. The model is trained for 5k iterations.

Adversarial Training Finally, we perform adversarial training. In this stage, we switch to the student-forcing paradigm, where the generator only takes the first frame as input and recycles the actual generated frame for the next autoregressive step, strictly following the inference procedure. Then, the discriminator evaluates the generated results in parallel, producing logits after each frame for multi-duration discrimination.

We follow APT [49] to initialize the generator from the consistency distillation weights, and to initialize the discriminator from the diffusion adaptation weight. We change to use the relativistic pairing loss [36]:

$$\mathcal{L}_{RpGAN}(x_0,\epsilon) = f(D(G(\epsilon,c),c) - D(x_0,c)), \tag{1}$$

where G,D denote the generator and the discriminator respectively, $f_G(x) = -\log(1+e^{-x})$ or $f_D(x) = -\log(1+e^{x})$ is used each of their update steps respectively, c denotes the text condition and other interactive conditions. We calculate R1 and R2 regularization [73, 65] through the approximation technique proposed in APT [49]:

$$\mathcal{L}_{aR1} = \lambda \| D(x_0, c) - D(\mathcal{N}(x_0, \sigma \mathbf{I}), c) \|_2^2, \tag{2}$$

$$\mathcal{L}_{aR2} = \lambda \| D(G(\epsilon, c), c) - D(\mathcal{N}(G(\epsilon, c), \sigma \mathbf{I}), c) \|_2^2, \tag{3}$$

where $\epsilon = 0.1$ and $\lambda = 1000$. Since the discriminator is initialized from the diffusion model, we follow APT to provide timesteps by random uniform sampling $t \sim \mathcal{U}(0, 1)$. We do not shift the timestep for the discriminator. We use RMSProp optimizer with $\alpha = 0.9$ following APT [49].

We first perform training without the long-video extension training technique. The videos are 5s to 10s in duration. We train it using a low learning rate of 3e-6 following APT [49] and a batch size of 256 for 500 generator updates. The resulting model can only generate up to 10 seconds and will drift for videos longer than 10 seconds.

Then we apply the long video training technique. The training videos are still from 5s to 10s, and we extend it once with an overlap of 1s to a total maximum duration of 19s (10 + (10-1)). This stage is trained for 500 generator updates. Then we turn up the extension to 5 times, to a total maximum duration of 55s (10 + 5×(10-1)). We find it necessary to decrease the batch size to 64 and increase the learning rate to 1e-5 for the extension training for the model to make adequate changes in a reasonable amount of time.

Since the generator in student-forcing mode must recurrently perform model forward for each autoregressive step during training, we switch FSDP to ZERO 2 mode to save all the model parameters on each machine. This avoids repeated parameter gathering and improves the training seed. The discriminator and text encoder still adopt ZERO 3 to shard all the model parameters for memory saving.

Computational Resources We use 256 H100 GPUs for our final training and employ gradient accumulation where necessary to reach our final batch size. The model is trained in approximately 7 days, where the diffusion adaptation and the long-video adversarial training take the majority of the time.

C Variational Autoencoder

We train a lightweight VAE decoder to fit the real-time budget. Specifically, our original VAE decoder has 3 residual blocks per resolution scale, and has channels [128, 256, 512, 512] at each resolution scale. Our lightweight VAE decoder reduces the number of residual blocks per resolution to 2, and reduces the channels to [64, 128, 256, 512]. This results in nearly 3 times speed-up without visible quality degradation.

D Teacher-Forcing Adversarial Training

The adversarial training supports both student-forcing and teacher-forcing modes. To implement student forcing, the generator runs autoregressively with KV cache and recycles the actual generated frame as input for the next autoregressive step. The discriminator evaluates the results in parallel. To implement teacher forcing, the generator takes ground-truth video frames as past prediction inputs and predicts the next frames in parallel. The discriminator runs autoregressively and always uses the KV cache from the real videos to attend to the ground-truth past frames.

Figure 8 visualizes teacher-forcing adversarial training. Specifically, in teacher-forcing mode, the generator given input $I1, I2, I3$ generates independent output $O2, O3, O4$. Namely, the output $O3$ only has a correlation with $I2$ but not with $O2$. Therefore, the discriminator must independently evaluate the generated results with their correct dependencies to produce logits $L2, L3, L4$. Since the discriminator transformer is causal, the repeated computation can be saved using KV cache.

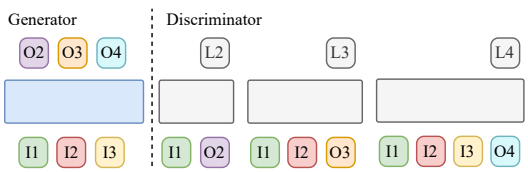

Figure 8: Teacher-forcing adversarial training

We have conducted experiments with teacher-forcing adversarial training, and the model fails to generate reasonable videos as discussed in the main paper. We suspect LLMs are able to train with teacher-forcing mode because they use a discrete codebook to encode words, where slight inaccuracy is less relevant. But our model predicts continuous latent values for the entire frame, where slight inaccuracy accumulates.

E The Importance of Result Recycling

We conduct an experiment to study the importance of result recycling. Specifically, we keep the exact architecture and training settings, and we mask the recycled input as zero tensors, except the first frame, which takes in the user image. We find that models trained without recycling input cannot generate large motion. Some of the movements become incohesive as well. The video visualization is provided on our website.

F I2V Evaluation

The table in the main text compares our model under the 736×416 setting. For the other models we compare to, we largely follow the default sampling setting for each model, including the number of steps and CFG [29]. We also use the default resolution for each model to ensure that the model has been properly trained on the expected resolution. Specifically, we use 896×544 for Hunyuan [44], 832×464 for Wan2.1 [95], 960×544 for SkyReel-V2 [9]. We note that we run 5 samples per prompt for all the comparisons per VBench-I2V [32] requirement, except for SkyReel-V2 which we only run 1 sample per prompt and reduce the sampling steps from its default 50 to 30. This is because SkyReel-V2 is too computationally intensive to generate one-minute videos.

We additionally provide the evaluation metrics under the 1280×720 resolution in Tab. 6. Note that 1280×720 is trained and inference with a smaller attention window size $N = 15$ to fit the memory.

Table 6: Quantitative VBench-I2V [32] metrics on 1280×720 compared to 736×416.

Frames	Method	Resolution	Quality								Condition	
			Temporal Quality	**Frame Quality**	Subject Consistency	Background Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	**I2V Subject**	**I2V Background**
1440	Ours	736×416	**89.79**	62.16	87.15	89.74	99.11	76.50	56.77	67.55	96.11	97.52
		1280×720	88.24	**64.30**	87.95	90.10	99.16	63.29	57.79	70.80	**96.51**	**98.18**

G Camera-Conditioned World Exploration

Training We make a few modifications on CameraCtrl II [25] to make it better support causal generation. First, CameraCtrl II uses Plücker embeddings to represent the camera position and orientation, where it treats the first frame as the initial position, and the other frames are relative to the first frame. This is problematic as the value can grow unbounded if the displacement forever increases. We change it so that each frame is only relative to the previous frame. Hence, the Plücker embeddings only represent the camera changes between immediate frames to prevent unbounded growth of values. Second, CameraCtrl II uses the original Plücker coordinate to represent each camera ray, which consists of a direction vector and a moment vector. The moment vector encodes the displacement information, which is computed as the cross product of a point on the line and the direction vector. We find that this implicit representation unnecessarily increases the complexity for the model to learn. Rather, we directly encode the camera ray's origin and direction. Third, the input scaling to the model is in fact a hyperparameter that is not previously explored. We scale the coordinate inputs to roughly 1 standard deviation to simplify model learning. We also drop samples whose camera embeddings have very large values. These outliers are caused by inaccurate camera estimation and are detrimental to the stability of adversarial training. Last, we use random initialization instead of zero initialization for the input projection of the new channels. We find that random initialization helps the model to adapt to the new inputs much more quickly.

The camera-conditioned model is trained separately from the I2V model. We start from the I2V diffusion adaptation weights and continue training on the camera-conditioned task. The consistency distillation and adversarial training are done separately for this dedicated model. The training settings are mostly the same as the I2V model. For the long-video extension training, we randomly sample new camera trajectories for the extended parts.

Evaluation Our evaluation metrics follow CameraCtrl II [25]. Specifically, we compute Fréchet Video Distance (FVD) [92] against the ground-truth videos. We compute the movement strength (Mov) on RAFT-extracted [88] dense optical flow of foreground objects identified by TMO-generated [12] segmentation masks. Translational (Trans) and rotational (Rot) errors are computed by comparing estimated camera parameters using VGGSfM [100] with the ground truth. Geometric Consistency (Geo) is computed as the successful ratio of VGGSfM to estimate camera parameters. This indicates the quality of 3D geometry consistency of the generated scene. Appearance Consistency (Apr) is computed by comparing the cosine distance of each frame's CLIP [70] vision embedding to the average of the entire video clip.

H Societal Impacts

Our work proposes a new approach for real-time streaming video generation for interactive applications. Our approach is faster and more computationally efficient than existing approaches. This potentially enables the adoption of more real-time interactive applications. We do not consider our work to bring risk for significant negative societal impacts. The videos generated by our method still contain imperfections that are easy to identify as generated videos, which prevents the technology from being used for malicious purposes.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes, the main claims in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Yes, a limitation section is provided toward the end of the paper.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The details of our architecture, implementation, hyperparameters, training procedures, and evaluation setups are provided in supplementary materials for reproducibility.

   Guidelines:

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

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

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

   Answer: [No]

   Justification: The model, data, and code used in this paper is proprietary and not currently scheduled for release. However, we describe our methodology in details for reproducibility.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The details are provided in supplementary materials.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: We follow established benchmarks and the evaluation protocols of prior works, which do not report error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report the resources in supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Yes, the research in the paper fully conforms to the NeurIPS Code of Ethics in every respect.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Yes, we analyze the potential social impact in the supplementary material due to space limitations.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The assets used are properly credited and licensed.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.