

---

# Autoregressive Adversarial Post-Training for Real-Time Interactive Video Generation Supplementary Materials

---

Anonymous Author(s)

Affiliation

Address

email

## 1 A Corrections

2 In the main text, line 224 to 227 should actually be placed at the beginning of line 248. We apologize  
3 for the mistake in preparing the manuscript.

## 4 B Video Visualization

5 We provide video samples on <https://aaptneurips.blob.core.windows.net/assets/supplementary.html>.  
6 We strongly recommend reviewers to visit this webpage to visualize our model’s results.

## 7 C Model Architecture

8 **Diffusion Transformer** Our diffusion transformer largely follows the MMDiT design [4]. It has  
9 8B parameters and 36 transformer blocks. The discriminator adopts the same architecture. Therefore,  
10 our generator and our discriminator consist of 16B parameters for the adversarial training.

11 **Block Causal Attention** We implement block causal attention using Flash Attention 3 [17] in a  
12 for-loop. We find it to provide reasonable performance for training. We leave the exploration for  
13 more performance implementation to future work. For inference, recurrent autoregressive steps are  
14 taken, and Flash Attention 3 can be naturally adopted without a performance penalty.

15 **Positional Embedding** As the duration of the generation becomes agnostic to our causal archi-  
16 tecture, we modify the 3D rotary positional embeddings (RoPE) [19]. Specifically, the positional  
17 embeddings continue to stretch dynamically along the spatial dimension to help the model generalize  
18 to different resolutions, while the positional embeddings are changed to have a fixed interval along  
19 the temporal dimension to support arbitrary lengths of training and generation.

20 **Parallelism** We adopt FSDP [24] for data parallelism. We use ZERO 2 for the generator during  
21 student-forcing training that requires recurrent forward calls to avoid repeated parameter gathering,  
22 and ZERO 3 for all other modules to save memory. We also adopt Ulysses [8] as our context parallel  
23 strategy. We shard each video sample across 8 GPUs. Gradient checkpointing is also utilized per  
24 transformer block to fit the memory requirement.

## 25 D Training Details

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

We follow the original model to use the flow-matching parameterization [12]. Specifically, given sample  $x_0$  and noise  $\epsilon$ , 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 [1] 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 [13] with a learning rate of  $1e-5$  and a weight decay scale of  $0.01$  throughout the process. We first train on  $736 \times 416$  (equivalent to  $640 \times 480$  by area) 5-second videos for 20k iterations with a batch size of 256. Then, we add  $1280 \times 720$  to the mix for another 6k iterations with a batch size of 128. Finally, we turn up the maximum duration of  $736 \times 416$  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 [18] 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 [11].

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 [6]. 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 [18] 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 [11] 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 [9]:

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

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 [16, 14] through the approximation technique proposed in APT [11]:

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

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

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 [11].

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 [11] 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 ( $10 + 5 \times (10-1)$ ). 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 speed. The discriminator and text encoder still adopt ZERO 3 to shard all the model parameters for memory saving.

We only perform the adversarial training at  $640 \times 352$  resolution, and we find the model is able to zero-shot extend to higher resolutions since it has seen higher resolutions at the diffusion adaptation and the consistency distillation stages.

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

## E 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.

## F 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 1 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.

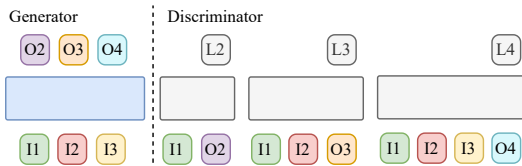


Figure 1: Teacher-forcing adversarial training

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.

## G 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

125 generate large motion. Some of the movements become incohesive as well. The video visualization  
126 is provided on our website.

## 127 H I2V Evaluation

128 The table in the main text compares our model under the  $736\times 416$  setting. For the other models we  
129 compare to, we largely follow the default sampling setting for each model, including the number of  
130 steps and CFG [6]. We also use the default resolution for each model to ensure that the model has  
131 been properly trained on the expected resolution. Specifically, we use  $896\times 544$  for Hunyuan [10],  
132  $832\times 464$  for Wan2.1 [22],  $960\times 544$  for SkyReel-V2 [2]. We note that we run 5 samples per prompt  
133 for all the comparisons per VBench-I2V [7] requirement, except for SkyReel-V2 which we only  
134 run 1 sample per prompt and reduce the sampling steps from its default 50 to 30. This is because  
135 SkyReel-V2 is incredibly slow to generate one-minute videos.

136 We additionally provide the evaluation metrics under the  $1280\times 720$  resolution in Tab. 1.

Table 1: Quantitative VBench-I2V [7] metrics on  $1280\times 720$  compared to  $736\times 416$ .

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	$736\times 416$	<b>89.79</b>	62.16	87.15	89.74	99.11	76.50	56.77	67.55	96.11	97.52
		$1280\times 720$	88.24	<b>64.30</b>	87.95	90.10	99.16	63.29	57.79	70.80	<b>96.51</b>	<b>98.18</b>

## 137 I Camera-Conditioned World Exploration

138 **Training** We make a few modifications on CameraCtrl II [5] to make it better support causal  
139 generation. First, CameraCtrl II uses Plücker embeddings to represent the camera position and  
140 orientation, where it treats the first frame as the initial position, and the other frames are relative to the  
141 first frame. This is problematic as the value can grow unbounded if the displacement forever increases.  
142 We change it so that each frame is only relative to the previous frame. Hence, the Plücker embeddings  
143 only represent the camera changes between immediate frames to prevent unbounded growth of values.  
144 Second, CameraCtrl II uses the original Plücker coordinate to represent each camera ray, which  
145 consists of a direction vector and a moment vector. The moment vector encodes the displacement  
146 information, which is computed as the cross product of a point on the line and the direction vector.  
147 We find that this implicit representation unnecessarily increases the complexity for the model to learn.  
148 Rather, we directly encode the camera ray’s origin and direction. Third, the input scaling to the model  
149 is in fact a hyperparameter that is not previously explored. We scale the coordinate inputs to roughly  
150 1 standard deviation to simplify model learning. We also drop samples whose camera embeddings  
151 have very large values. These outliers are caused by inaccurate camera estimation and are detrimental  
152 to the stability of adversarial training. Last, we use random initialization instead of zero initialization  
153 for the input projection of the new channels. We find that random initialization helps the model to  
154 adapt to the new inputs much more quickly.

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

160 **Evaluation** Our evaluation metrics follow CameraCtrl II [5]. Specifically, we compute Fréchet  
161 Video Distance (FVD) [21] against the ground-truth videos. We compute the movement strength  
162 (Mov) on RAFT-extracted [20] dense optical flow of foreground objects identified by TMO-  
163 generated [3] segmentation masks. Translational (Trans) and rotational (Rot) errors are computed by  
164 comparing estimated camera parameters using VGGSfM [23] with the ground truth. Geometric Con-  
165 sistency (Geo) is computed as the successful ratio of VGGSfM to estimate camera parameters. This  
166 indicates the quality of 3D geometry consistency of the generated scene. Appearance Consistency  
167 (Apr) is computed by comparing the cosine distance of each frame’s CLIP [15] vision embedding to  
168 the average of the entire video clip.

## J 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.

## References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [7] 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.
- [8] 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.
- [9] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.
- [10] 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.
- [11] 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.
- [12] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- [13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [14] 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.
- [15] 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.

- 219 [16] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative  
220 adversarial networks through regularization. *Advances in neural information processing systems*, 30, 2017.
- 221 [17] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3:  
222 Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing*  
223 *Systems*, 37:68658–68685, 2024.
- 224 [18] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint*  
225 *arXiv:2310.14189*, 2023.
- 226 [19] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced  
227 transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- 228 [20] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer*  
229 *Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II*  
230 *16*, pages 402–419. Springer, 2020.
- 231 [21] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and  
232 Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint*  
233 *arXiv:1812.01717*, 2018.
- 234 [22] Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao  
235 Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint*  
236 *arXiv:2503.20314*, 2025.
- 237 [23] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry  
238 grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision*  
239 *and pattern recognition*, pages 21686–21697, 2024.
- 240 [24] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid  
241 Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel.  
242 *arXiv preprint arXiv:2304.11277*, 2023.