What Matters in Clean-Context Autoregressive Video Diffusion

Jeffrey Wei [1] **Steven Zhou** [1]

Abstract

Autoregressive video diffusion models accumulate error over long rollouts, a failure mode known as drift. Diffusion Forcing (Chen et al., 2024) corrupts every frame in a window by an independently sampled noise level, a partial masking of its content, and computes the denoising loss at all positions. At inference the past context is noise-free, a configuration that occurs with nearly zero probability under this training distribution, yielding a train–test mismatch that is often linked to drift. A popular remedy is clean-context training, which instead conditions the model on a clean, uncorrupted past during training so as to match how it is queried at inference. We study clean-context effects through a controlled, parameter- and compute-matched comparison across denoiser backbones and datasets. We find that the benefit is backbone-dependent: clean-context training substantially reduces drift for a convolutional U-Net but provides none for a parameter-matched diffusion transformer. A simple structural probe finds the cause: the U-Net's temporal normalization lets future frames affect past ones, while the transformer keeps frames separate. We find that clean-context training is most helpful when the denoiser leaks information across frames in this way, and that removing the leakage removes the benefit.

1. Introduction

Generative video models are increasingly used as general-purpose world simulators: they capture physical dynamics (Yang et al., 2024), act as playable neural game engines (Valevski et al., 2024; Savva et al., 2026), and serve as virtual environments for policy evaluation (Quevedo et al., 2025). Two paradigms dominate how they are generated. Autoregressive models rollout a sequence one token at a time, each step conditioned on the past, making them naturally causal and able to rollout to arbitrary length. Diffusion models instead denoise an entire window in parallel, yielding high per-frame fidelity and flexible conditioning but fixing the size of what is generated. Autoregressive video diffusion merges the best of both worlds through Diffusion Forcing (DF) (Chen et al., 2024), where a single denoiser gives each frame its own independent noise level so that it acts as a diffusion model within a window and autoregressively across windows, inheriting diffusion's per-frame quality together with autoregression's unbounded, causal rollout. This method now dominates the autoregressive video diffusion landscape.

Even so, the Diffusion Forcing paradigm degrades as the video context grows, accumulating error the longer the rollout. This failure mode is *drift*: small conditioning errors compound as the model is rolled forward, and rollouts that look plausible early gradually lose temporal consistency, geometric coherence, and physical plausibility. Reducing drift is a prerequisite for almost every downstream use of long-horizon video generation. The flexibility that makes Diffusion Forcing attractive is also a source of this drift. Because every frame is noised independently during training, including frames that later serve as context, the model is supervised almost entirely on noisy windows. At inference, the context is clean, a setting the model almost never sees in training, since independent per-frame noising makes a fully clean context extremely unlikely. Operating in this unfamiliar regime, DF inference tends to propagate small errors that feed back as context and compound into drift.

A growing body of work addresses this train–test gap by training on clean past frames and noisy future frames (Hu et al., 2024; Jin et al., 2025; Gao et al., 2025; Sand-AI, 2025; Gu et al., 2025). We defer a more in depth discussion of related work to Section A. Closest to our setup, FAR (Gu et al., 2025) introduces *Stochastic Clean Context*, replacing part of the noisy training context with clean frames and excluding those frames from the loss. This combines two changes. First, the loss is computed only on the future frames the model must generate, not on the past frames it is given as context, akin to standard autoregressive training. Second, context is kept noise-free during training so it matches inference. This second change is a form of teacher forcing:

[1]Yale University, New Haven, Connecticut, USA. Correspondence to: Jeffrey Wei <jeffrey.wei@yale.edu>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

the model only ever sees perfect, ground-truth context in training, so it never learns to handle the imperfect context it feeds itself during a rollout, risking exposure bias (Bengio et al., 2015; Ranzato et al., 2016). Whether keeping the context noise-free helps or hurts is unclear a priori. Because prior methods apply both changes at once and test only a single backbone, two questions remain open: which change actually reduces drift, and whether the gain comes from the training change at all rather than from the architecture it happens to use. We therefore test each change on its own, across two denoiser backbones.

We evaluate each configuration on long autoregressive rollouts, measuring how far the generated video drifts from the ground-truth distribution as the horizon grows. As our primary metric we report Fréchet Video Distance (FVD, Unterthiner et al., 2018), the de facto standard for video generation quality. Because FVD has been found to be content-biased and to under-weight long-range temporal coherence (Ge et al., 2024), we additionally report the JEPA Embedding Distance (JEDi) (Luo et al., 2025), a distance on features from a Joint-Embedding Predictive Architecture (JEPA) (Bardes et al., 2024) that is more sensitive to temporal structure, as a robustness check. We compute both at endpoint horizons and on contiguous 32-frame segments of the rollout, so that drift can be read as the rise across segments rather than from a single endpoint number.

Our main contributions are as follows.

1. We run a controlled, parameter- and compute-matched study of clean-context training, isolating its two changes (clean prefix, masked prefix loss) and crossing them with two denoiser backbones (convolutional U-Net, diffusion transformer) and two datasets, DMLab (Beattie et al., 2016) and Minecraft (Yan et al., 2023).
2. We find that the benefit of clean-context training is backbone-dependent: it substantially reduces drift for a convolutional U-Net but gives none for the diffusion transformer, where keeping the context noise-free is actually harmful. This finding is robust across both datasets and under both FVD and a JEPA-based metric.
3. We trace this difference to whether the denoiser is frame-causal: a weight-independent probe shows the U-Net's temporal normalization lets future frames influence past ones, while the diffusion transformer's does not. We confirm this directly: making the U-Net's normalization frame-causal removes the leak, and the benefit of clean-context training vanishes with it. Clean-context training still closes the train–test mismatch, but that mismatch hurts drift only in architectures with the leak, so the benefit is architecture-dependent.

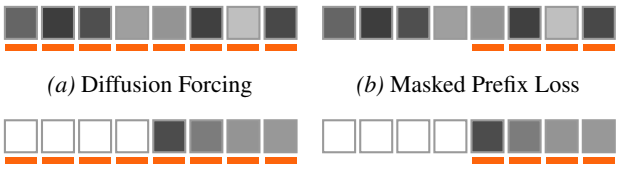
2. Setup

A causal video generator predicts future frames from a window of past context, and rolls out long videos by sliding that window forward and feeding its own outputs back as context. We make this setup precise and define our experimental configurations below.

Diffusion Forcing. A DF video model is trained on windows $x_{1:T}$ of T frames. Each frame t samples an independent noise level $k_t \in \{0, \dots, K\}$ ($k_t{=}0$ clean, $k_t{=}K$ pure noise) and is corrupted to $x_t^{k_t}$. A single denoiser f_θ reads the noised window together with its noise levels and predicts every frame, trained with the $\boldsymbol{v}$-objective (Salimans & Ho, 2022; Chen et al., 2024): $\mathcal{L}_{\mathrm{DF}} = \mathbb{E}_{x,\, k_{1:T}}\big[\sum_{t=1}^{T} \| f_\theta(x_{1:T}^{k_{1:T}}, k_{1:T})_t - v_t \|^2\big]$. Drawing each k_t independently allows one model to support any noise pattern at test time, whether autoregressive or full-sequence diffusion.

The train–test mismatch. At inference the past is observed rather than noised. The initial c context frames are clean ($k_{1:c}{=}0$) while only the future $x_{c+1:T}$ is denoised, and long rollouts slide this window forward, with each step using the newly generated frames as the next conditional context. During training, every k_t is drawn independently and uniformly, so the single window that inference always uses, a clean prefix beside a noisy future, arises only with probability $(K{+}1)^{-c}$ per draw. With K on the order of 10^3 diffusion steps, even a single clean context frame is seen essentially never, and a four-frame prefix is a roughly 10^{-12} event. The denoiser is therefore trained almost entirely outside the inference regime. At test time it must condition on a clean past it never learned to denoise, so its small errors feed back as context and compound into drift.

Two training-side axes. Clean-context training bundles two changes, which we separate into independent axes. The first is a *clean prefix*: we clamp the first c frames to the clean noise level, matching the schedule used at inference. In practice we hold them not at exactly zero noise but at a small stabilization level, because at inference the context is the model's own previously generated frames rather than ground truth; a little noise keeps training from conditioning on a context cleaner than any it will encounter. The second is a *masked prefix loss*: we stop supervising the prefix and pay loss only on the frames the model must predict, reallocating the supervision budget toward them, as in standard autoregressive loss masking. Crossing the two axes yields the four configurations of Figure 1: DF (neither), Clean-only (clean prefix only), Mask-only (masked loss only), and Full (both). All four share the same denoiser, sampler, parameter count, and per-step compute, and run identical sliding-window inference on a clean past, and differ only in



(a) Diffusion Forcing *(b)* Masked Prefix Loss

(c) Clean Prefix Only *(d)* Full (clean-context)

Figure 1. The four training configurations as a 2×2 over the two axes: the shading is the per-frame noise level (white clean, dark noised) and the orange bar marks the supervised frames, with the $c{=}4$ prefix noised (top, DF) or kept noise-free (bottom) and its loss calculated for all positions (left) or masked (right).

training.

Two backbones. Prior clean-context work couples the scheme with a particular denoiser, so a gain reported on one architecture does not reveal whether the scheme or the architecture is responsible. We therefore run the entire 2×2 on two denoisers that share the per-frame-noise interface of Diffusion Forcing but differ in how they mix information across time.

U-Net. We instantiate the 3D-convolutional U-Net backbone with causal temporal attention and rotary position encoding taken directly from the public Diffusion Forcing codebase (Chen et al., 2024). It has spatial-only convolutions, so frames interact only through the temporal attention and through the temporal GroupNorm in its ResNet blocks, which pools statistics across all frames in the window.

DiT. We implement a strictly frame-causal diffusion transformer following the per-frame-noise formulation of Song et al. (2025) and the transformer design of Peebles & Xie (2023), with per-frame conditioning and a causal mask on its temporal attention. The causal mask lets each frame attend only to itself and earlier frames, so its prediction depends on the present and past but never on future frames. The DiT of Song et al. (2025), by contrast, attends over the whole window in both directions, letting later frames influence earlier ones. We train and sample the DiT under Diffusion Forcing exactly as the U-Net, with the same per-frame-noise objective, the four configurations of Figure 1, and the same sliding-window inference, so only the architecture differs between the two backbones.

We match the two backbones to within $1.01\times$ in parameter count (18.65M for the U-Net and 18.84M for the DiT; Section B), so that any difference between them is not an artifact of capacity.

Datasets. We run the study on two procedurally generated navigation environments standard in the autoregressive-video literature (Yan et al., 2023): DMLab (DML) (Beattie et al., 2016), a first-person 3D maze with random textures and a moving camera, and Minecraft (MC), an open-world game with richer scene content and faster camera motion. Both are controlled proxies for long-horizon state drift at a resolution that admits parameter- and compute-matched training within a single GPU-day, and we roll out 128 frames over 128 held-out clips per configuration.

Evaluating drift. Two properties of drift shape how we measure it. First, drift is *distributional*: individual frames can still look realistic while the rollout as a whole drifts away from real video. We therefore score each rollout as a distribution, using FVD and JEDi, rather than comparing its frames to a single ground-truth video. Pixel metrics are a poor fit, since they compare against one ground-truth trajectory and can reward a degenerate, near-static rollout. Second, drift *accumulates* over the rollout, so a single endpoint score hides when a model starts to fail. We compute each metric on the four contiguous 32-frame segments and measure drift as the rise from the first segment to the last. Because FVD and JEDi use different feature backbones, we trust results that agree across both.

Altogether, the study crosses the 2×2 of training axes with two denoiser backbones and two datasets, scoring each rollout by two distributional metrics of long-horizon consistency (FVD and JEDi). We additionally report standard per-frame fidelity metrics (LPIPS (Zhang et al., 2018), PSNR, and SSIM (Wang et al., 2004)) in Table 5.

3. Experiments

U-Net results. On the convolutional U-Net, clean-context training is a large and consistent win over Diffusion Forcing (Table 1). In every U-Net setting the best configuration is a clean-context variant, on both datasets and under both metrics, and the gap to DF is substantial rather than marginal. Which clean-context variant wins shifts from one dataset to the next, so the ordering within the family is not stable enough to draw a conclusion from. The improvement is also immediate: the U-Net's DF deficit is already present on the very first segment, before autoregressive errors have had time to accumulate (Table 3). Because little drift has compounded that early, this opening gap reflects a conditioning mismatch the model faces from the first frame rather than accumulated drift, a distinction we return to in Section 4.

DiT results. Now the best configuration in every setting is a random-prefix variant, DF or Mask-only, and the clean-prefix configurations consistently perform worse (Table 1). The clean prefix that improves the U-Net is precisely what hurts the DiT. The contrast is sharpest at the opening of the rollout: where DF carried a large first-segment penalty on the U-Net, the DiT shows none, DF being its best configuration from the very first frames (Table 3). A frame-causal

Table 1. Endpoint FVD and JEDi (both ↓) at horizon h=128 over 128 held-out clips, for the four configurations across two backbones (U-Net, DiT) and two datasets (DML=DMLab, MC=Minecraft). Bold marks the best configuration per column: every U-Net winner is a clean-context variant (Clean-only or Full), every DiT winner a random-prefix variant (DF or Mask-only).

	FVD↓				JEDi↓			
	U-Net		DiT		U-Net		DiT	
Config	DML	MC	DML	MC	DML	MC	DML	MC
DF	487	3692	**568**	1379	5.78	88.7	**17.6**	**33.7**
Mask-only	217	1969	608	**1192**	4.25	42.9	20.1	39.3
Clean-only	**165**	2910	631	2193	**2.36**	41.1	23.0	40.9
Full	214	**1605**	686	1940	3.55	**26.6**	21.3	43.1

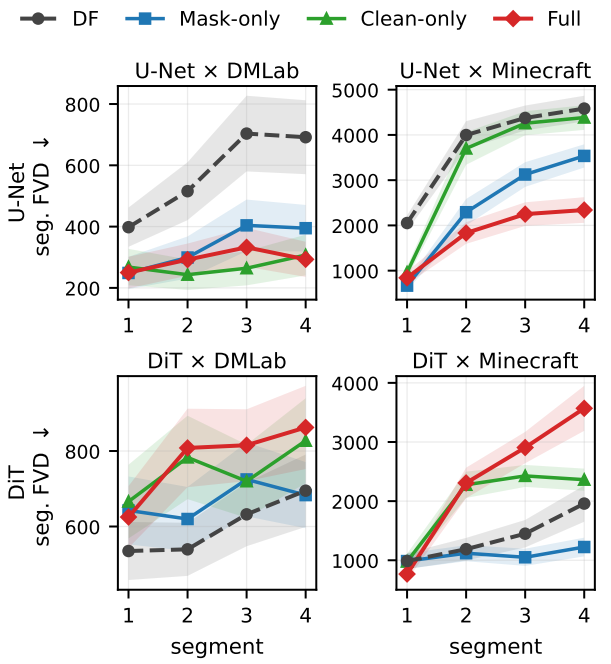


Figure 2. Segment-wise FVD (↓) across the four 32-frame segments of the rollout with bootstrap confidence intervals over 128 clips. On the U-Net (top row) the clean-context configurations sit below DF (gray dashed) at every segment. On the DiT (bottom row) DF is already the lowest.

DiT shows no sign of the immediate conditioning mismatch that clean-context training tries to fix.

Segment-wise drift. The endpoint scores in Table 1 average distributional quality over the whole rollout, which hides when a model fails: a configuration that produces a few sharp opening frames and then degrades earns the same score as one that is uniformly mediocre. We therefore split each 128-frame rollout into four contiguous 32-frame segments, score each against its matched ground-truth window, and read drift as the rise in FVD from the first segment to the last (Figure 2, numerics in Table 3). The segment view tells the same story as the endpoints, but more sharply.

On the U-Net (top row), Clean-only, Mask-only, and Full all sit below DF at every segment. DF's deficit is already clear in the opening segment—a conditioning mismatch present before much autoregressive error has accumulated—and it widens as DF climbs over the rollout while the others stay lower. Where the variants differ is in how flat they stay. Full is the standout: it posts the lowest final-segment FVD on both datasets and is the only configuration that stays flat on Minecraft as well as DMLab. Clean-only and Mask-only flatten the DMLab rollout but rise steeply on Minecraft, where only Full rises more slowly than DF. On the DiT (bottom row) the pattern reverses: DF sits among the lowest and flattest trajectories while the clean-prefix configurations climb above it, and DF is the best or near-best configuration in nearly every setting. The two metrics agree on this split—in every setting JEDi and FVD pick the same family winner, clean-prefix on the U-Net and random-prefix on the DiT (Table 1).

Across both datasets and both metrics, clean-context training improves the U-Net's generation quality but degrades it on the DiT.

4. Why the Benefit Is Backbone-Dependent

Frame-causality. The two denoisers have the same parameter count, so capacity cannot explain the reversal. What differs is whether the output for one frame can depend on the noise levels of the *other* frames in the window. The U-Net's ResNet blocks normalize every frame with a mean and variance pooled over the whole window using a temporal GroupNorm. Since a clean frame and a noisy frame have very different statistics, the shared statistics are dominated by the noisy ones, and the clean frame is normalized as if it were noisy too, leaking information from the future to the past. Thus, with temporal GroupNorm, we lose the frame-causal nature that autoregressive rollouts should have.

In normal DF training every frame is given a random noise level, so the model sees windows with all kinds of clean and noisy mixtures, but almost never the one situation it meets at every inference step: a fully clean past beside a

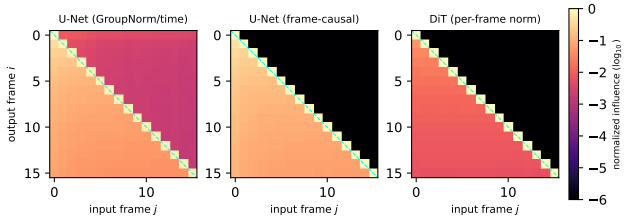


Figure 3. Cross-frame influence $M[i, j]$: the mean change in output frame i when input frame j is perturbed, so a frame-causal denoiser is lower-triangular. The U-Net's temporal GroupNorm leaks across the boundary (nonzero upper triangle), while the frame-causal variant and the DiT's per-frame conditioning leave it exactly zero.

noisy future. Because of the leak, that combination of a clean past and a noisy future yields normalization statistics the model essentially never saw in training, and since this happens on the very first generated frame, the U-Net drifts right away rather than only late in the rollout, as seen in Figure 2. Clean-context training fixes this for the U-Net by training on exactly that case. A frame-causal denoiser never runs into it: each frame is normalized on its own, so a clean past frame is processed the same way regardless of the noisy future, and clean frames are familiar from training.

Clean-context training, however, comes at a cost. As the rollout progresses the model conditions not on ground truth but on its own earlier output, which carries small, compounding errors, whereas clean-context training only ever showed it a perfect, error-free past. The model therefore learns to lean on a flawless history it will not actually have at test time. By keeping the past slightly noisy, plain DF is a closer match to the imperfect self-generated context, so forcing the prefix perfectly clean trades that robustness away, the classical exposure-bias problem (Bengio et al., 2015; Ranzato et al., 2016). The two effects then settle the backbone split: for the U-Net, fixing the leak is worth more than the exposure-bias cost and clean-context training wins overall; for the frame-causal DiT there is no leak to fix, so only the cost remains and the clean prefix makes things slightly worse.

A cross-frame leakage probe. To make frame-causality measurable, we perturb input frame j for each denoiser and measure the resulting mean change in output frame i, which gives an influence matrix $M[i, j]$ (Figure 3). We define the acausal mass as the mean of its strict upper triangle ($j{>}i$, a future input moving a past output), which must be zero for a strictly frame-causal model. The DiT's acausal mass is exactly zero. The U-Net's is nonzero, 1.18×10^{-3}, or about 1.6% of its causal mass. Its convolutions are spatial-only (kernel $1{\times}3{\times}3$) and its attention is causal, which leaves the temporal GroupNorm, which pools mean and variance over all frames in the window, as the only acausal path. Replacing it with a GroupNorm that folds the time axis into the batch dimension (see Section D) drives the acausal mass to exactly zero while leaving the causal mass essentially unchanged, confirming that the temporal GroupNorm is the sole leak. This is a structural property of the architectures and is independent of the trained weights.

To confirm the leak is the cause, we retrain the U-Net with the leak removed. We replace its temporal GroupNorm with the frame-causal variant and train both DF and the full scheme again, at the same budget (Section D). With the leaky GroupNorm the full scheme clearly helps, as demonstrated in the main U-Net result. However, with the frame-causal GroupNorm, the same change no longer helps and is mildly harmful, just as on the DiT. Removing the leak also lowers DF's own drift, so the leak hurts even before any change to training. The segment-wise results agree: with the leak the clean prefix helps most in the early frames, where the mismatch is worst, and costs little later. Without the leak it gives no early gain and only a late cost. Clean-context training is worth as much as the information leak it corrects, and no more.

5. Conclusion

Clean-context training is often presented as a general remedy for autoregressive drift, but in our controlled study its value is set entirely by the denoiser: a clear win for the convolutional U-Net, and nothing, even a small loss, for the parameter-matched frame-causal transformer. The cause is a single acausal path, the U-Net's temporal GroupNorm, and removing it removes the benefit, so the scheme is not a general fix but a correction for a specific architectural leak, worth as much as the leak it offsets. This matters in practice: as autoregressive video denoisers move toward strictly frame-causal designs (Gu et al., 2025; Sand-AI, 2025), gains reported for clean-context training on convolutional backbones should not be assumed to carry over, and drift interventions are best reported together with the architecture they were measured on.

Limitations. We train a single seed per setting and substitute cross-dataset and cross-metric replication for seed repeat robustness. The causal GroupNorm ablation that grounds the mechanism (Section D) is trained on a reduced 30k-step budget and $n{=}32$ validation clips, compared to the 100k-step budget and $n{=}128$ validation clips of the other models. Our transformer is strictly frame-causal and differs from the bidirectional DiT of Song et al. (2025), so the finding is scoped to frame-causal versus temporally leaky denoisers rather than to transformers in general. We also operate at $64{\times}64$ resolution for 100k steps, below the scale of the diffusion-forcing transformer literature.

Impact Statement

This paper studies methods for generative video modeling. As with generative models in general, the techniques could be used in both beneficial and harmful ways; we do not see consequences specific to this work beyond those already widely discussed for generative modeling.

Acknowledgements

This work builds on the public Diffusion Forcing codebase of Chen et al. (2024) and the research template of Boyuan Chen (https://github.com/buoyancy99/research-template). We thank the authors for releasing them.

References

Bardes, A., Garrido, Q., Ponce, J., Chen, X., Rabbat, M., LeCun, Y., Assran, M., and Ballas, N. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., Schrittwieser, J., Anderson, K., York, S., Cant, M., Cain, A., Bolton, A., Gaffney, S., King, H., Hassabis, D., Legg, S., and Petersen, S. DeepMind Lab. *arXiv preprint arXiv:1612.03801*, 2016.

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2015.

Carreira, J. and Zisserman, A. Quo vadis, action recognition? a new model and the Kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., and Sitzmann, V. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Gao, K., Shi, J., Zhang, H., Wang, C., Xiao, J., and Chen, L. Ca2-VDM: Efficient autoregressive video diffusion model with causal generation and cache sharing. In *International Conference on Machine Learning (ICML)*, 2025.

Ge, S., Mahapatra, A., Parmar, G., Zhu, J.-Y., and Huang, J.-B. On the content bias in Fréchet video distance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Gu, Y., Mao, W., and Shou, M. Z. Long-context autoregressive video modeling with next-frame prediction. *arXiv preprint arXiv:2503.19325*, 2025.

Harvey, W., Naderiparizi, S., Masrani, V., Weilbach, C., and Wood, F. Flexible diffusion modeling of long videos. In *Advances in Neural Information Processing Systems*, 2022.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. In *Advances in Neural Information Processing Systems*, 2022.

Hu, J., Hu, S., Song, Y., Huang, Y., Wang, M., Zhou, H., Liu, Z., Ma, W.-Y., and Sun, M. ACDiT: Interpolating autoregressive conditional modeling and diffusion transformer. *arXiv preprint arXiv:2412.07720*, 2024.

Huang, X., Li, Z., He, G., Zhou, M., and Shechtman, E. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Jin, Y., Sun, Z., Li, N., Xu, K., Jiang, H., Zhuang, N., Huang, Q., Song, Y., Mu, Y., and Lin, Z. Pyramidal flow matching for efficient video generative modeling. In *International Conference on Learning Representations (ICLR)*, 2025.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

Luo, G. Y., Favero, G., Luo, Z. H., Jolicoeur-Martineau, A., and Pal, C. Beyond FVD: Enhanced evaluation metrics for video generation quality. In *International Conference on Learning Representations*, 2025. arXiv:2410.05203.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Quevedo, J., Sharma, A. K., Sun, Y., Suryavanshi, V., Liang, P., and Yang, S. WorldGym: World model as an environment for policy evaluation, 2025. URL https://arxiv.org/abs/2506.00613.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks. In

International Conference on Learning Representations, 2016.

Ruhe, D., Heek, J., Salimans, T., and Hoogeboom, E. Rolling diffusion models. In *International Conference on Machine Learning (ICML)*, 2024.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. *International Conference on Learning Representations*, 2022.

Sand-AI. MAGI-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.

Savva, G., Michel, O., Lu, D., Waiwitlikhit, S., Meehan, T., Mishra, D., Poddar, S., Lu, J., and Xie, S. Solaris: Building a multiplayer video world model in minecraft, 2026. URL https://arxiv.org/abs/2602.22208.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Song, K., Chen, B., Simchowitz, M., Du, Y., Tedrake, R., and Sitzmann, V. History-guided video diffusion. In *International Conference on Machine Learning (ICML)*, 2025.

Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Valevski, D., Leviathan, Y., Arar, M., and Fruchter, S. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Williams, R. J. and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989.

Yan, W., Hafner, D., James, S., and Abbeel, P. Temporally consistent transformers for video generation. In *International Conference on Machine Learning (ICML)*, 2023.

Yang, M., Du, Y., Ghasemipour, K., Tompson, J., Schuurmans, D., and Abbeel, P. Learning interactive real-world simulators. In *International Conference on Learning Representations (ICLR)*, 2024.

Yu, L., Lezama, J., Gundavarapu, N. B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Birodkar, V., Gupta, A., Gu, X., Hauptmann, A. G., Gong, B., Yang, M.-H., Essa, I., Ross, D. A., and Jiang, L. Language model beats diffusion – tokenizer is key to visual generation. In *International Conference on Learning Representations*, 2024.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

A. Related Work

Our work builds on Diffusion Forcing (Chen et al., 2024), whose objective noises every frame in a window to an independent level and supervises all of them, and on the clean-context family that modifies it. These methods close the train–inference gap on the training side by holding past frames at clean noise levels and, in most cases, masking the loss there: ACDiT (Hu et al., 2024) attends from noised blocks to clean past blocks through a skip-causal mask, Pyramidal Flow Matching (Jin et al., 2025) and Ca^2-VDM (Gao et al., 2025) interleave clean and noised blocks for efficiency, MAGI-1 (Sand-AI, 2025) scales a clean-history variant, and FAR (Gu et al., 2025), closest to our setup, replaces part of the noisy context with clean frames excluded from the loss. These works bundle the two changes and validate each on a single backbone; we instead separate the two axes and cross them with the denoiser, which is where the effect turns out to live. Several of these systems are transformers run at scale, but their causal models do not contradict our reading: the DiT of Song et al. (2025) (DFoT) is bidirectional, and its causal video models use U-Net and U-ViT backbones. We therefore scope our claim to the frame-causality of the denoiser, not to convolution versus transformers or to any system's headline performance.

Other methods attack the same gap along different axes. Self Forcing (Huang et al., 2025) works on rollout supervision, generating short autoregressive rollouts during training and supervising them against a teacher, in the spirit of scheduled sampling (Bengio et al., 2015; Ranzato et al., 2016); History-Guided Video Diffusion (Song et al., 2025) works on the sampler, leaving the DF objective unchanged and instead guiding inference with classifier-free guidance (Ho & Salimans, 2021) over noisy past frames; and Rolling Diffusion (Ruhe et al., 2024) changes the noise schedule itself, correlating neighboring frames' levels. Clean-context training, which we do not compare against these empirically, is purely a training-data change. It also sits on the teacher-forcing side of a classical tension (Williams & Zipser, 1989): conditioning on ground-truth context risks exposure bias, since the model never trains on its own imperfect outputs, which is why a small stabilization noise is kept on the prefix in practice.

Finally, long-context and world-model video diffusion extend the architecture for longer context (Harvey et al., 2022; Ho et al., 2022; Yu et al., 2024; Valevski et al., 2024); clean-context training is complementary, since any such system still chooses a noise schedule for its visible context. Throughout we use standard diffusion training (Ho et al., 2020) and DDIM sampling (Song et al., 2021) with the $\boldsymbol{v}$-objective and a cosine schedule (Karras et al., 2022; Salimans & Ho, 2022).

B. Reproducibility

Dataset preparation. We use the standard video-prediction preparation of DMLab and Minecraft: source clips at 64×64 RGB from a random-/scripted-policy agent, sub-sampled (DMLab `frame_skip`=2). We use each dataset's default training/validation split, and evaluate on 128 held-out validation clips, fixed across configurations, backbones, and runs.

Architectures. The two backbones are described in Section 2 and their sizes are listed in Table 2; the DiT additionally uses factorized spatial-full and temporal-causal attention with patch size 8. The U-Net is the convolutional video backbone of the Diffusion Forcing codebase, not the recurrent (GRU) temporal variant of the original paper (Chen et al., 2024).

Table 2. Architecture and training hyperparameters for the two denoisers. Training, diffusion, and sampling settings are shared; only the architecture differs.

	U-Net	DiT
Architecture		
hidden size	48	256
depth	$(1, 2, 4, 8)$ mults	11 blocks
attention heads	4	8
head dim	32	32
patch size	—	8
normalization	temporal GroupNorm	per-frame AdaLN-Zero
parameters	18.65M	18.84M
Training (shared)		
optimizer	AdamW, β=$(0.9, 0.99)$	
learning rate	8×10^{-5}, 10k warmup	
weight decay	2×10^{-3}	
batch size	8	
steps	100k	
gradient clip	1.0	
precision	`fp16` (mixed)	
window × resolution	32 frames × 64×64	
Diffusion / sampling (shared)		
objective	$\boldsymbol{v}$-prediction	
noise schedule	cosine, K=1000	
sampler	DDIM, 100 steps, η=0	
stabilization level	15	

Training. Every setting (four configurations across two backbones and two datasets) is trained on a single NVIDIA H200 GPU; the optimizer, learning-rate schedule, and diffusion settings are shared across all of them and are listed in Table 2. The configurations differ only in how the training-step noise schedule and loss mask treat the clean prefix.

Sampling. At inference we sample with DDIM (100 steps, η=0) and the $\boldsymbol{v}$-objective. For autoregressive rollout we use a sliding 32-frame window with a c=4-frame clean prefix; for horizons longer than the window, the window slides, each step emitting new frames and re-using the previous window's last 4 frames as the clean prefix. All four configurations of a backbone run identical sliding-window denoising at inference, so they differ only in their trained weights.

Endpoint and segment-wise metrics. Each model is rolled out autoregressively to 128 generated frames per validation clip, over n=128 held-out clips. Endpoint FVD/JEDi at horizon h use features over the $[0, h]$ window relative to the matched ground-truth window; segment-wise FVD/JEDi partition the rollout into four contiguous 32-frame windows and score each against the matched ground-truth window. FVD uses I3D features (Unterthiner et al., 2018; Carreira & Zisserman, 2017); JEDi is a polynomial-MMD on V-JEPA features (Luo et al., 2025; Bardes et al., 2024). We compute the Fréchet distance with the eigenvalue identity $\mathrm{tr}\sqrt{\Sigma_a\Sigma_b} = \sum\sqrt{\mathrm{eigvals}(\Sigma_a\Sigma_b)}$ rather than `scipy.linalg.sqrtm` for numerical stability.

Code and checkpoints. Code and all sixteen trained checkpoints (four configurations × two backbones × two datasets), together with the frame-causal-norm probe and the evaluation scripts that reproduce Tables 1, 3 and 4 and Figures 2 and 3, can be found at `https://github.com/jwei302/cct`.

C. Segment-wise Numerics

Table 3 gives the per-segment FVD behind Figure 2. Each entry is FVD computed independently on a contiguous 32-frame segment of the rollout against the matched ground-truth segment: Δdrift is segment 4 minus segment 1.

Table 3. Segment-wise I3D-FVD↓ over 128 held-out clips, all four settings. Δdrift is $\mathrm{FVD}_{97-128} - \mathrm{FVD}_{1-32}$.

Backbone / Dataset	Config	1–32	33–64	65–96	97–128	Δdrift
U-Net / DMLab	DF	398	516	704	692	+294
	Mask-only	249	299	404	395	+146
	Clean-only	268	243	264	307	+39
	Full	250	292	332	293	+43
U-Net / Minecraft	DF	2053	3998	4375	4582	+2530
	Mask-only	668	2292	3125	3536	+2868
	Clean-only	962	3700	4258	4384	+3422
	Full	841	1832	2250	2340	+1499
DiT / DMLab	DF	535	540	632	695	+160
	Mask-only	643	620	725	683	+41
	Clean-only	666	783	720	827	+161
	Full	625	808	815	862	+237
DiT / Minecraft	DF	987	1187	1449	1958	+971
	Mask-only	986	1119	1050	1224	+238
	Clean-only	978	2276	2427	2363	+1385
	Full	767	2308	2906	3570	+2803

D. Causal GroupNorm Ablation

The standard temporal GroupNorm in the U-Net's ResNet blocks normalizes each 5D activation (B, C, F, H, W) over the channel and *frame* axes jointly, pooling mean and variance across all F frames in the window; a noisy future frame therefore shifts the normalization statistics of a clean past frame, the acausal leak of Section 4. The frame-causal variant closes this path with a single change: it reshapes the activation to $(B{\cdot}F, C, H, W)$ before normalizing, so each frame is normalized on its own statistics, then reshapes back. Every other component, including the spatial-only convolutions and the causal temporal attention, is left untouched, so the variant differs from the U-Net only in that its one acausal path is removed.

We isolate the temporal GroupNorm as the cause of the U-Net's clean-context benefit by training the same backbone under both DF and the full clean-context scheme, once with the standard (leaky) GroupNorm and once with the frame-causal variant. All four runs share a matched 30k-step budget on DMLab and are evaluated on 32 held-out clips; the budget is below the main runs, so absolute FVD is higher, but the four runs are matched and only the GroupNorm and the training scheme vary.

With the leaky GroupNorm, clean-context training lowers FVD substantially, most of the gain arriving in the early segments where the inference-time schedule mismatch is largest. With the frame-causal GroupNorm the same change no longer helps and is mildly harmful, and the endpoint benefit $\mathrm{FVD(DF)}-\mathrm{FVD(Full)}$ flips from strongly positive to slightly negative. Independently, making the GroupNorm frame-causal sharply lowers DF's own drift across every segment, which identifies the leak as a source of drift in its own right rather than only as the reason clean-context helps.

Table 4. Causal GroupNorm ablation: segment-wise and endpoint (h=128) FVD↓ for DF and Full under the leaky and frame-causal GroupNorm.

GroupNorm	scheme	seg1	seg2	seg3	seg4	h=128	Δ
leaky	DF	1367	1481	1794	1591	1492	
	Full	841	1011	1416	1620	**1038**	+454
frame-causal	DF	735	765	694	736	**461**	
	Full	702	738	773	902	530	−69

E. Pixel-Level Metrics

For completeness, Table 5 reports per-clip pixel-level metrics (LPIPS, PSNR, SSIM) against the matched ground-truth rollout for one representative setting (U-Net / DMLab). The scores are close across configurations and do not track the distributional separation, confirming that the gap between methods is distributional rather than pixel-wise.

Table 5. Pixel-level metrics (LPIPS↓, PSNR↑, SSIM↑) on autoregressive rollouts at $h \in \{32, 64, 128\}$ on U-Net / DMLab.

	LPIPS↓			PSNR↑			SSIM↑		
Config	h=32	h=64	h=128	h=32	h=64	h=128	h=32	h=64	h=128
DF	**0.450**	**0.481**	0.498	11.85	11.15	10.74	0.175	0.140	0.121
Mask-only	0.455	0.487	0.500	11.70	11.09	10.92	0.173	0.141	0.124
Clean-only	0.455	**0.481**	**0.493**	**11.88**	**11.31**	**11.05**	**0.176**	**0.146**	**0.127**
Full	0.455	0.483	0.502	11.75	11.13	10.76	0.174	0.145	**0.127**

F. Qualitative Rollouts

The displayed clip is the one at the median per-clip FVD improvement of Full over DF.

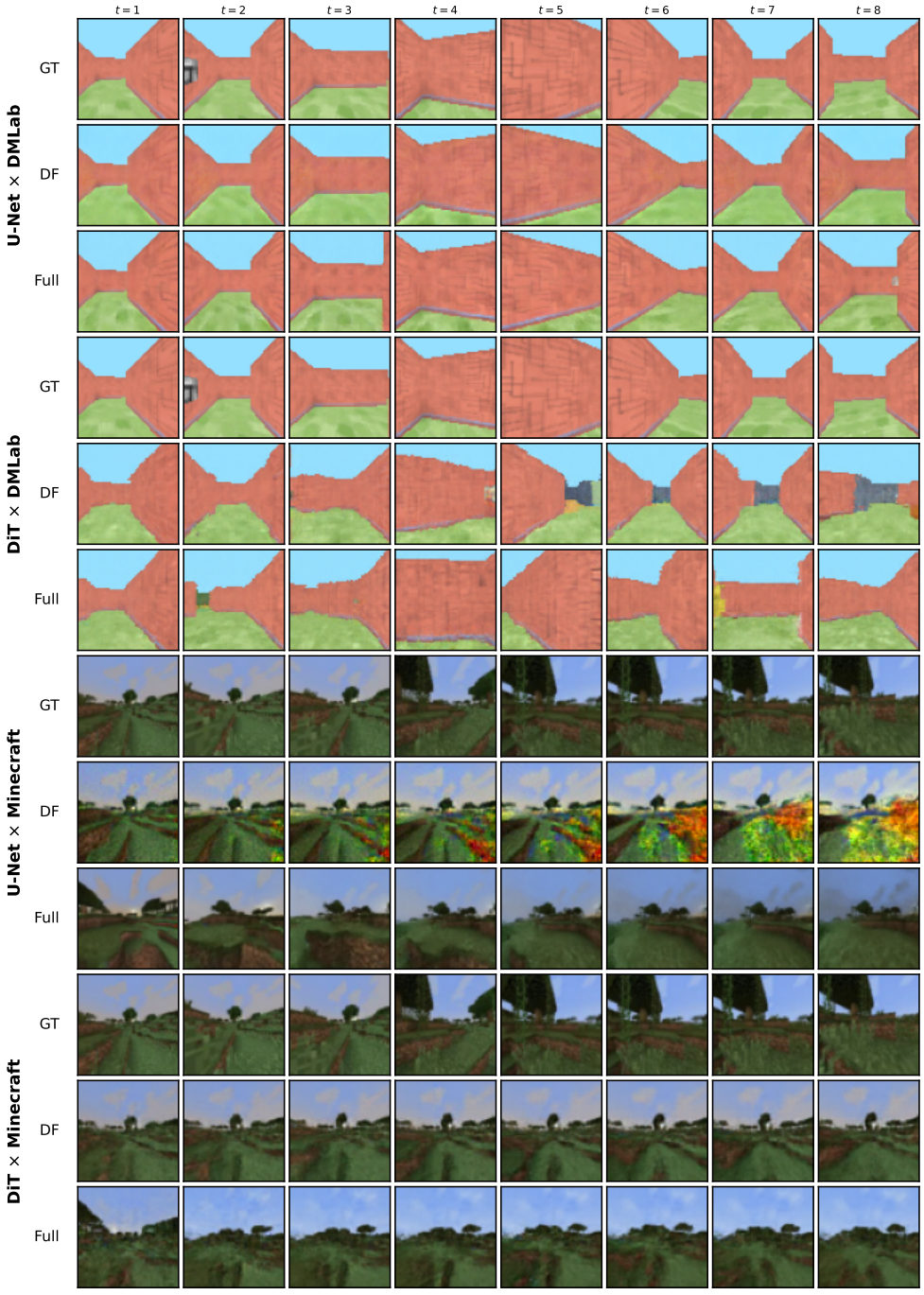


Figure 4. Qualitative rollout over the first eight generated frames (t=1–8).

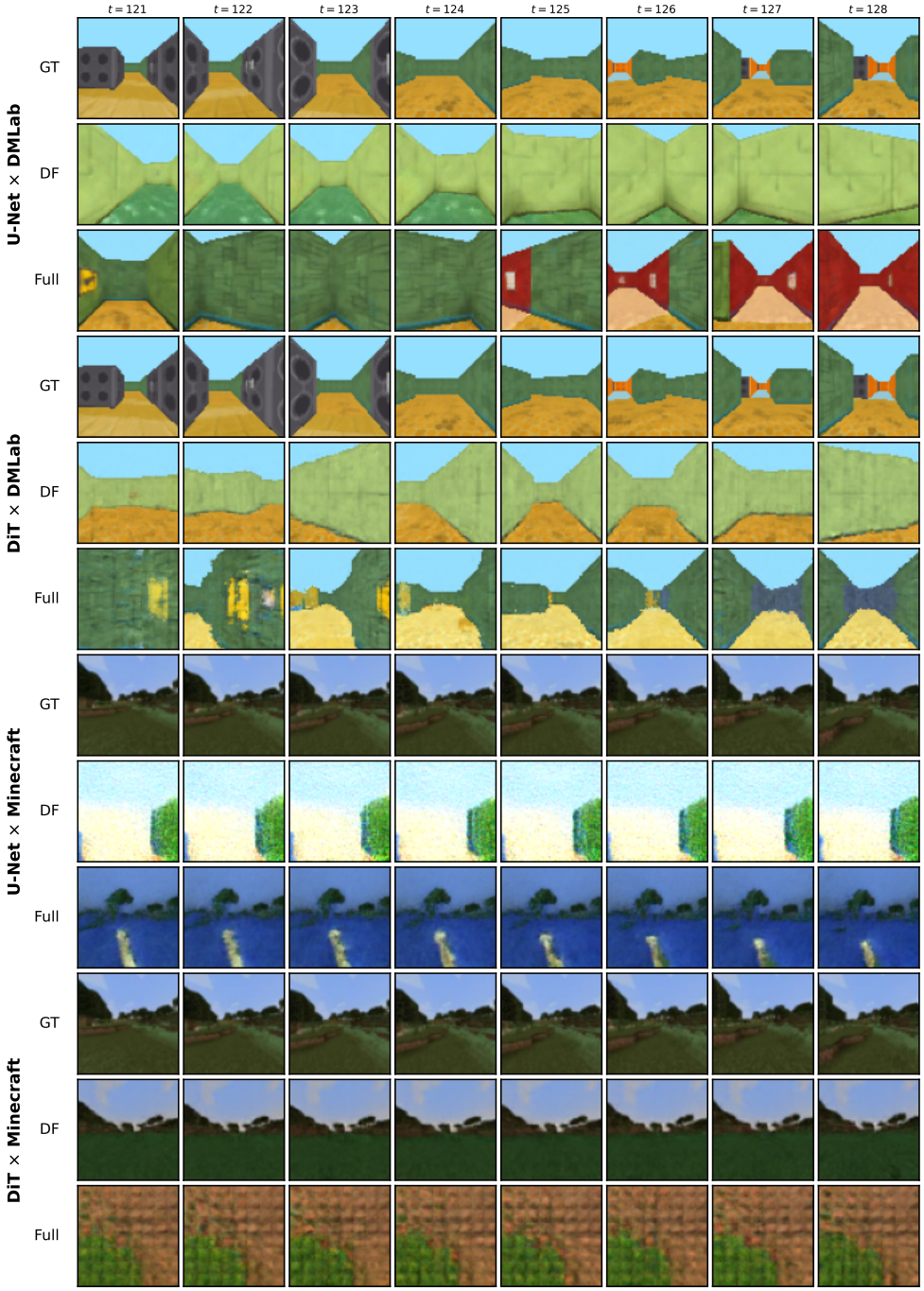


Figure 5. Qualitative rollout over the last eight generated frames (t=121–128).