Causality in Video Diffusers is Separable from Denoising

Xingjian Bai[1,2] Guande He[2,3] Zhengqi Li[2]
Eli Shechtman[2] Xun Huang[2] Zongze Wu[2]
[1]Massachusetts Institute of Technology [2]Adobe Research
[3]The University of Texas at Austin

Abstract

Causality—referring to temporal, uni-directional cause-effect relationships between components—underlies many complex generative processes, including videos, language, and robot trajectories. Current causal diffusion models entangle temporal reasoning with iterative denoising, applying causal attention across all layers, at every denoising step, and over the entire context. In this paper, we show that the causal reasoning in these models is separable from the multi-step denoising process. Through systematic probing of autoregressive video diffusers, we uncover two key regularities: (1) early layers produce highly similar features across denoising steps, indicating redundant computation along the diffusion trajectory; and (2) deeper layers exhibit sparse cross-frame attention and primarily perform intra-frame rendering. Motivated by these findings, we introduce Separable Causal Diffusion (SCD), a new architecture that explicitly decouples once-per-frame temporal reasoning, via a causal transformer encoder, from multi-step frame-wise rendering, via a lightweight diffusion decoder. Extensive experiments on both pretraining and post-training tasks across synthetic and real benchmarks show that SCD significantly improves throughput and per-frame latency while matching or surpassing the generation quality of strong causal diffusion baselines.

1 Introduction

Modeling causality[1] is a core problem in diffusion generation modeling. Starting from fitting image distributions [5, 13, 17, 55, 58, 60–62], diffusion models [30, 44] have achieved great success across modalities such as videos [6, 25, 31, 32, 75, 79, 86], audio [39, 45], and language [29, 50, 54]. In its basic form, diffusion models denoise all tokens simultaneously, generating the entire output all at once. This is still the design of many state-of-the-art video diffusion models. However, this formulation overlooks the temporal evolution inherent in sequential data—allowing the future information to influence the past, and preventing crucial applications such as long-term, real-time video streaming [34, 38, 81, 90]. To incorporate temporal causal dependencies and enable autoregressive video generation, researchers have attempted to replace bidirectional full attention inside the denoiser with causal attention [23, 34, 90], as commonly used in the LLM community. This mechanism applies bidirectional attention within a frame (or chunk of frames) and causal attention across frames (or chunks). When combined with the diffusion process, every token within a frame must pass through the entire network iteratively, computing both intra-frame and cross-frame attention at every layer and every denoising step.

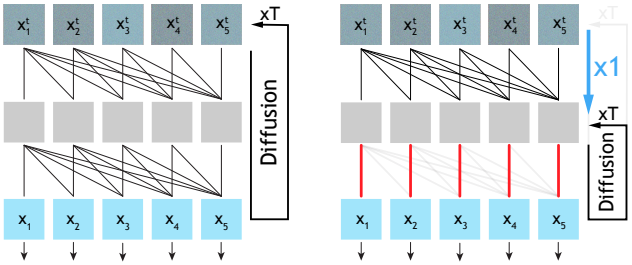


Figure 1. **Causality in autoregressive video diffusion models is separable from the denoising process.** The prevailing design of causal diffusion models for visual generation performs causal attention densely across *all layers and all denoising steps* (left). However, we uncover two important observations (right): 1) early denoiser layers share highly repetitive computation across denoising steps (blue); 2) deep layers primarily attend to intra-frame tokens, with sparse cross-frame connections (red).

While causal attention is essential for modeling temporal evolution, directly transplanting it from LLMs overlooks a key difference: diffusion models typically perform multi-step refinement for each frame, rather than generating in a single pass. The current design of causal diffusion tightly entangles temporal reasoning with iterative denoising, with each layer at every step repeatedly performing causal reasoning. This raises a fundamental question: Is multi-step refinement truly required for temporal reasoning?

[1]In this paper we use *causality* narrowly to mean the temporal arrow-of-time—*the past determines the future, not vice versa.*

To answer this question, we conduct detailed probing analysis and finetuning experiments on autoregressive (AR) video diffusion models. We consistently observe that temporal reasoning in AR models is separable from the denoising process (Fig. 1). In particular, we find that causal reasoning in early layers is highly redundant across denoising timesteps, as indicated by the high similarity in middle-layer output features across denoising steps. We also observe that temporal computation in deeper layers is far less frequent: careful attention visualizations reveal that deeper layers predominantly perform intra-frame attention while rarely attending across frames.

Motivated by the sparsity and redundancy we uncover, we introduce Separable Causal Diffusion (SCD), a novel decoupled causal architecture in which a temporal causal-reasoning module operates once per frame, while a lightweight frame-wise diffusion renderer handles visual refinement. Concretely, a causal transformer reads the historical clean frame tokens through KV cache and produces a latent that summarizes the entities, layout, and expected motion from its context. This context latent is then reused across all denoising steps for that frame. A diffusion module receives both the current noisy frame tokens and the context latent, and performs a frame-wise iterative denoising process without any cross-frame computation. Taken together, our design mirrors next-token prediction in LLMs (except that we perform next-*frame* prediction here followed by continuous rendering), reallocating compute from repeated cross-frame operations to per-frame refinement, thereby reducing latency and memory while preserving generation quality.

We conduct extensive experiments at both pretraining and post-training stages for causal video diffusion models across synthetic and real datasets. We show that SCD trained from scratch matches or surpasses causal diffusion baselines in generation quality while achieving 2–3× lower latency. Furthermore, to demonstrate scalability, we finetune SCD from a pretrained bidirectional teacher diffusion model, achieving strong video generation quality with substantially higher throughput compared with AR baselines.

Contributions. In summary, we make the following contributions: 1) Through careful probing and finetuning experiments, we observe that causal reasoning in existing causal video diffusion models is redundant across denoising steps and sparse across time. 2) We introduce a novel Separable Causal Diffusion (SCD) architecture that fully leverages these observations. On both pretraining and post-training tasks, SCD demonstrates strong effectiveness across multiple datasets compared with baseline models.

2 Related Work

From Bidirectional to Autoregressive Video Diffusion. Diffusion-based video generative models have achieved remarkable fidelity by employing spatio-temporal Transformers with bidirectional attention over entire video sequences. Recent methods [6, 11, 12, 16, 21, 22, 26, 59, 79] advance this paradigm through careful architectural design and large-scale training, achieving state-of-the-art visual quality. However, their non-causal design requires generating all frames simultaneously, resulting in high latency and preventing real-time streaming or interactive applications.

To enable online, low-latency generation, recent efforts have shifted toward autoregressive (AR) video generation, particularly using diffusion models with causal transformers. Instead of producing all frames at once, AR diffusion models generate videos in a causal manner, conditioning each frame only on past frames. This causal dependence not only aligns with the arrow of time but also enables efficient inference via KV caching, making it attractive for interactive settings. Pioneering AR approaches include models trained from scratch [8, 11, 23, 38, 56] and techniques that distill a causal generator from a pretrained video diffusion model [10, 15, 34, 83, 90].

AR-Diffusion Hybrid Models. To leverage the strengths of both paradigms, a growing body of work combines an AR module with a diffusion module. In the image domain, several recent works [18, 41] have demonstrated that an AR transformer can operate on continuous tokens to generate a coarse layout, which is then refined by a diffusion module to produce high-fidelity images. In the video domain, MarDini [46] and VideoMAR [91] both employ an AR module to produce a context representation of the video, which is subsequently used by a diffusion module to generate visual tokens. Notably, VideoPoet [38] also adopts a frame-wise autoregressive strategy, but it uses a single-pass decoder operating on discrete tokens and lacks a diffusion module for refinement, leading to low-quality generation. In parallel, another line of work aims to unify understanding and generation tasks through hybrid AR transformers paired with diffusion heads [19, 52, 65, 70, 98].

Separability and Sparsity in Video Models. Separability and sparsity have long been central themes in video modeling: because the space-time dimension is dense, naively porting image architectures becomes prohibitive, motivating early/late fusion and factorized designs that decouple spatial and temporal processing [1, 7, 9, 37, 42, 48, 67, 71]. In video diffusion models, researchers have recently leveraged inherent 3D attention patterns from pretrained video models to accelerate generation [80, 84, 93, 94]. Our work can be viewed as a continuation of this discussion on separability and sparsity in video models, specifically in the context of temporally causal video diffusion. Beyond video,

separability in diffusion models has likewise been studied and exploited in other modalities, including images [76] and language [3].

3 Preliminaries: Causal Diffusion Models

In this section, we review the causal diffusion paradigm, a variant of diffusion models that generates a step-indexed sequence in a causal manner. This is the predominant pipeline for frame-autoregressive video generation [10, 23, 34, 90]. We formalize the continuous-time objective and highlight why it embeds causal dependence throughout the entire diffusion trajectory. Finally, we briefly introduce Teacher Forcing and Diffusion Forcing [10] as training techniques for causal diffusion models.

A causal generator models the joint distribution of a sequence $x_{1:N} = (x_1, \ldots, x_N)$ by predicting each element from its past. With optional per-step controls $a_{1:N}$[2], the joint distribution factorizes as

$$p_\theta(x_{1:N} \mid a_{1:N}) \;=\; \prod_{i=1}^{N} p_\theta\big(x_i \mid C_i = (x_{<i},\, a_{\le i})\big), \quad (1)$$

where each conditional probability is implemented by a diffusion renderer that attends to its context, C_i.

We adopt a continuous notion of time, $t \in [0, 1]$ and define a forward diffusion path, connecting the data distribution with a standard Gaussian $\mathcal{N}(0, I)$:

$$x_i^t \;=\; (1 - t)\, x_i \;+\; t\, \epsilon_i, \text{where } \epsilon_i \sim \mathcal{N}(0, I). \quad (2)$$

A causal diffusion network takes the noisy samples as input, and v_θ predicts its velocity on the diffusion path, conditioning on (t, C_i)

$$\hat{v}_{i,\theta} \;=\; v_\theta(x_i^t,\; t,\; C_i)\,,$$

while the ground-truth velocity is the time derivative of the diffusion path

$$u(x_i^t,\, t \mid x_i) \;=\; \frac{d}{dt} x_i^t \;=\; \epsilon_i - x_i. \quad (3)$$

Training loss is defined on the gap between the predicted and ground-truth velocity under a time-weighted expectation:

$$L(\theta) = \mathbb{E}_{x,\, i,\, t,\, \epsilon}\Big[\, w(t) \,\big\|\, u(x_i^t,\, t \mid x_i) - v_\theta(x_i^t,\, t,\, C_i) \,\big\|^2\Big], \quad (4)$$

where $w(t)$ is a standard time weighting. Crucially, because $v_\theta(\cdot, t, C_i)$ is conditioned on C_i *for every* $t \in [0, 1]$ and $L(\theta)$ integrates over t, the model must repeatedly consult the context along the whole reverse trajectory: causal reasoning is therefore entangled with the entire diffusion path (and, in common implementations of the denoiser, propagated across all denoiser layers at each step).

[2]Global controls (e.g., a text prompt) can be treated as constant per-step conditioning.

Teacher Forcing and Diffusion Forcing. *Teacher Forcing (TF)* trains next-frame prediction with *clean* history: at each step the denoiser predicts the current frame while attending to ground-truth context frames. This provides a standard causal diffusion training recipe but induces a train–test mismatch: at inference, the model conditions on its own imperfect past outputs, resulting in severe error accumulation during roll-out [10, 68]. *Diffusion Forcing (DF)* [10] addresses this by *noising the context* during training: each context frame is independently perturbed to a sampled noise level, and the denoiser predicts the current frame while attending to these partially noised contexts. This simple augmentation better matches inference-time erroneous conditions. However, diffusion forcing conditions on noisy ground truth inputs during training but relies on clean past rollout at inference, leading to another form of mismatch between training and test conditions.

4 Uncovering Causal Separability

In this section, we study where the main *causal reasoning* actually occurs inside AR video diffusers. As a testbed, we adopt *WAN-2.1 T2V-1.3B* [79], one of the most capable open-source text-to-video models, and convert it to a frame-wise AR generator via teacher forcing [74, 79, 90]. To ensure that our findings do not hinge on this particular choice, we repeat similar observations on autoregressive video models trained from scratch and on other conditioning; consistent behaviors are summarized in Appendix A.2. For all probing experiments, we fix prompts/seeds and capture per-layer, per-step activations and attention maps.

4.1 Repetitive Computation across Denoising Steps

In causal video diffusion models, the historical context is typically attended at every denoising step. We investigate whether this repeated use of history is necessary in AR video generation. Prior work on accelerated sampling suggests that useful structure can be established early in the denoising trajectory of image diffusion models [24, 51, 64, 92]. We identify a similar but distinct phenomenon in AR video diffusion models: middle-layer activations within the same frame show extremely high cosine similarity (above 0.95), as illustrated in Fig. 2. Given the high dimensionality of each feature vector (1536-d in Wan 2.1 1.3B [78]), such consistently high cosine similarity indicates that the features are nearly identical across denoising steps. The effect is already visible at the earliest denoising steps, suggesting that the activations at early denoising steps stabilize quickly during the diffusion process.

The PCA visualization in Fig. 2 corroborates this finding: the principal components derived from the first denoising step closely align with those from later steps and successfully capture the object's global shape, pose, and fine structural

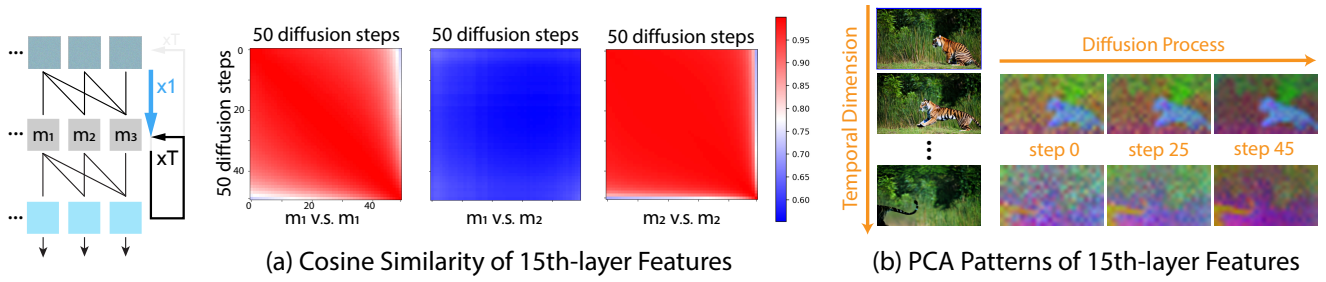


Figure 2. **Strong middle-block feature consistency across denoising steps.** (a) When denoising the same frame over 50 steps, the middle-block (15th block out of 30) features exhibit consistently high cosine similarity (above 0.95), suggesting that the features generated in the middle block are mostly shared across different diffusion steps. (b) PCA analysis further confirms that the middle-block features at the first and later diffusion steps are highly aligned, indicating that structures are effectively established even in the first step.

details (e.g., the curved, hook-like tail in the last frame) in the corresponding generated frame. We attribute this pronounced feature similarity to the redundancy inherent in AR video generation—the features of the current frame are largely determined by historical contextual frames. Consequently, content and motion dynamics are effectively established in a single step, while subsequent denoising iterations primarily refine low-level pixel details and rendering quality (see Appendix Fig. 10 for extended analysis across layers).

To further verify the redundancy observation, we finetune the baseline with a skip-layer design (detailed in Fig. 3 caption). Specifically, except for the first few denoising steps that run all 30 layers, subsequent steps skip layers 8–22 (15 middle layers), directly connecting early-layer outputs to late-layer inputs via residual connections. As shown in Fig. 3, the skipped model successfully generates high-quality videos that faithfully preserve the object identities, spatial layout, and motion dynamics of the baseline model. This demonstrates that the new architecture does not learn a new generative manifold but instead operates within the same manifold as the baseline model.

Figure 3. **Skipping the middle layers across denoising steps.** To take advantage of the repetitive computation, we finetune with a skip-layer design: except for the starting denoising steps, the denoiser skips a large chunk of 15 (out of 30) middle layers during diffusion. After short finetuning, semantics, layout, and motion are preserved and visual fidelity is restored. Full details on the design of this finetuning are provided in Appendix A.1.

4.2 Deep Layers are Separable in Time

To quantify how much temporal context is actually read at each layer, we compute the cross-frame attention mass

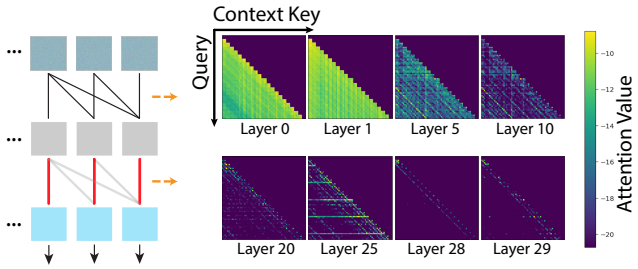


Figure 4. **Cross-frame attention becomes sparse with depth.** For a newly denoised frame i, we aggregate, for each transformer layer and attention head, the attention mass that query tokens at i assign to keys from its context frames. Results indicate that deeper layers allocate markedly less mass to past frames, indicating they focus on intra-frame refinement, and cross-frame attention is largely unnecessary.

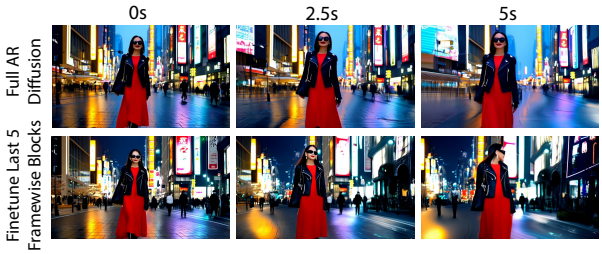


Figure 5. **Removing deep cross-frame attention.** We switch the last 5 (of 30) layers from a frame-causal mask to a frame-diagonal mask, removing their access to context-frame KV caches. A brief 5k-step finetune with the frame-diagonal mask stabilizes the semantics, layout, and motion and restores visual fidelity.

across the AR video diffuser: for each transformer layer, we sum the attention from queries at frame i to keys in frames $j < i$ (Fig. 4). This observation reveals a functional split in the model: early layers perform most temporal reasoning, while late layers focus on per-frame rendering with little long-range attention. Notably, although training uses a standard frame-wise causal mask that permits dense cross-frame attention, long-range sparsity nonetheless emerges in deeper layers as an intrinsic property of the learned model.

Motivated by the observed long-range sparsity, we investigate whether cross-frame attention in deep layers can be removed in the architecture. As shown in Fig. 5, a brief

5K-step finetuning on our partially frame-diagonal model effectively recovers the baseline visual generation quality. We validate these observations on additional model families, including a 4-step block-autoregressive Self-Forcing model (Appendix Figs. 11, 12) and a 3D UNet trained with Diffusion Forcing (Appendix Fig. 13), demonstrating that the separability patterns hold across different architectures and training objectives.

5 Separable Causal Diffusion

The analyses of causal video diffusion models in §4.1 and §4.2 reveal two complementary regularities—step-wise invariance in early layers and temporal independence across frames in deeper layers. Together, these findings imply a functional separation within causal video diffusion models, whose operations consist of (1) producing and reasoning over clean context tokens, and (2) leveraging these context priors for iteratively denoising corrupted video-frame tokens. As a result, applying fully causal attention throughout the entire AR diffusion process leads to substantial redundant computation. Pruning these inactive paths naturally reduces both computational and memory overhead, motivating our efficient decoupled architecture, **Separable Causal Diffusion (SCD)**, an encoder–decoder–style design that disentangles temporal causal reasoning from iterative denoising. Specifically, SCD comprises a causal-reasoning encoder, which performs AR computations to produce context tokens without requiring iterative denoising, and a lightweight frame-wise diffusion decoder, which focuses on synthesizing and refining the current frame conditioned on the context tokens from the encoder.

5.1 Temporal Causal Encoder

Motivated by the step-wise redundancy observed in early layers (§4.1), we design a causal transformer encoder $\mathcal{E}_\phi$ that runs once per generated frame, *outside* the diffusion process, using causal attention over historical contexts stored as KV caches. Specifically, it computes a compact *causal context* for the next frame:

$$c_i \;=\; \mathcal{E}_\phi(x_{<i},\, a_{\le i})\,, \tag{5}$$

where $x_{<i}$ are the previously generated frames before time i, and $a_{\le i}$ denotes conditioning signals (e.g., actions). The context c_i is a *sequence* of latent tokens with the same spatial dimensions as the frame tokens (e.g., $H/p \times W/p$ tokens for patch size p), produced by the final layer of the encoder $\mathcal{E}_\phi$. The causal context tokens c_i summarize the history and are reused by the diffusion decoder across all denoising steps when generating the current video frame x_i at time i. Intuitively, c_i encodes entities, layout, and motion cues anticipated for the generated frame at time i. Note that within $\mathcal{E}_\phi$, attention among spatial tokens within each frame is bidirectional, whereas temporal attention across frames is causal.

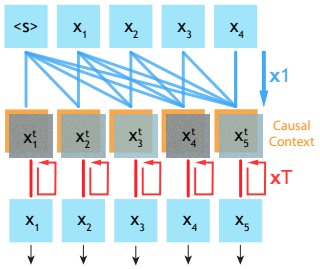


Figure 6. **Separable Causal Diffusion.** Once-per-frame causal reasoning produces a compact prior c_i, which the frame-wise diffuser reuses across T denoising steps to render x_i.

5.2 Frame-wise Diffusion Decoder

Motivated by the cross-frame temporal independence observed in deep layers (§4.2), we introduce a *lightweight frame-wise diffusion decoder* $\mathcal{D}_\theta$ that denoises noisy tokens corresponding to a video frame conditioned on the fixed contexts c_i from $\mathcal{E}_\phi$. In particular, $\mathcal{D}_\theta$ learns to predict velocity $\hat{v}_i^t$ for a frame at time i and denoising time step $t \in \{T, \dots, 1\}$

$$\hat{v}_i^t \;=\; \mathcal{D}_\theta\big(x_i^t,\, t,\, c_i\big), \tag{6}$$

The learned velocity $\hat{v}_i^t$ is used to iteratively denoise the corrupted video-frame tokens x_i^t (starting from Gaussian noise) into a clean latent x_i. The context c_i and noisy frame x_i^t are combined via frame-wise token concatenation along the sequence dimension, forming a joint input sequence that the decoder processes with bidirectional self-attention. The diffusion decoder uses bidirectional attention *within* each frame and does not propagate information *across* frames; all historical context information is provided through the learned c_i produced by the encoder $\mathcal{E}_\phi$.

5.3 Training and Inference Framework

Supervision. Our encoder and decoder are trained jointly in an end-to-end manner, with the next-frame prediction objective (Fig. 6). In particular, the encoder $\mathcal{E}_\phi$ takes in ground truth video frame tokens, processes them in parallel with causal attention, and generate a sequence of context tokens $\{c_i\}_{i=1}^N$ corresponding to next frames, following Teacher Forcing training paradigm. The decoder $\mathcal{D}_\theta$ then takes the noisy video-frame tokens $\{x_i^t\}_{i=1}^N$ together with $\{c_i\}_{i=1}^N$ to predict the corresponding velocities $\hat{v}_i^t$. The predicted velocities are supervised against the ground-truth conditional flow field (Equation 3) using the loss defined in Equation 4. Since our diffusion decoder operates independently for each frame, we can improve token utilization during training by repeating each video-frame latent and sampling multiple noise scales per frame in the training sequence, similar to prior work [41, 96].

Inference. Because $\mathcal{E}_\phi$ runs *once per frame* and $\mathcal{D}_\theta$ runs T times within the frame, the amortized per-frame time complexity is

$$\underbrace{\mathcal{O}(\mathcal{E}_\phi)}_{\text{once per frame}} + \underbrace{T \cdot \mathcal{O}(\mathcal{D}_\theta)}_{\text{per denoising step}},$$

with $\mathcal{O}(\mathcal{E}_\phi) \gg \mathcal{O}(\mathcal{D}_\theta)$ since $\mathcal{E}_\phi$ performs inter-frame causal attention with KV cache, whereas $\mathcal{D}_\theta$ operates on each video frame latent independently .

Moreover, prior AR video diffusion models require an extra pass to cache the current generated frame content after its generation [34, 90]. In contrast, as illustrated in Figure 6, our model follows a *next-frame denoising* paradigm, in line with the design in autoregressive language models, which eliminates the need for an extra model invocation for KV caching.

Corrupting causal context. To improve model robustness and context-following capability during inference, we inject noise to the historical context as done in prior work [11, 68]. Because our architecture separates temporal reasoning from frame-wise denoising, we can perturb their interface, the causal context c_i, as a means to inject corruption. We adopt a simple Gaussian corruption,

$$\tilde{c}_i = c_i + \eta\,\boldsymbol{\zeta}, \qquad \boldsymbol{\zeta} \sim \mathcal{N}(0, I). \tag{7}$$

Applying it during *training*, it acts as an augmentation to reduce exposure bias. At *inference*, it can also be used as a negative guidance signal. Compared with injecting noise to the frame tokens, corruption c_i does not require extra passes of the network and is thus very efficient. Empirically, we observe that modest noise corruption improves model robustness and context-following; full ablations are deferred to Appendix B.3.

6 Experiments

Setting. We evaluate our proposed Separable Causal Diffusion (SCD) architecture in two complementary settings: 1) training from scratch on low-resolution video datasets, and 2) fine-tuning a high-resolution pretrained text-to-video diffuser to our architecture. To study the effect of model capacity, we follow the DiT parameterization scheme (B/M/L) [57] and decompose the total transformer depth into a causal encoder and a diffusion decoder. We use the superscript ^{E} to denote variants with increased encoder depth and ^{D} to denote variants with increased decoder depth. Full hyperparameter specifications for all SCD variants are provided in Appendix D.2. We additionally benchmark a fully causal video diffusion baseline trained through teacher-forcing, denoted as Causal-DiT.

Table 1. Comparison & Ablation on **TECO–Minecraft 128×128**.

Model	Sec/F	144→156			36→264
		LPIPS↓	SSIM↑	PSNR↑	FVD↓
Latent FDM [53]	non-causal	0.429	0.349	13.4	167
FAR-M-Long [23]	2.2	0.251	0.448	16.9	39
Causal DiT-M	2.4	0.196	0.512	18.9	38.7
SCD-M	**0.52**	0.179	0.524	19.3	37.6
SCD-M^E	**0.52**	0.175	0.525	19.3	36.1
SCD-M^D	1.6	**0.168**	**0.535**	**19.5**	**34.9**

Table 2. Comparison & Ablation on **UCF-101 64×64**.

Model	Sec/F	4→12			
		LPIPS↓	SSIM↑	PSNR↑	FVD↓
RaMViD [33]	non-causal	0.090	0.639	21.37	396.7
MCVD-cp [73]	non-causal	0.088	0.658	21.82	468.1
FAR-B [23]	3.2	0.037	0.818	25.64	194.1
Causal DiT-B	3.9	0.038	0.827	25.85	187.6
SCD-B	**1.1**	0.038	0.824	25.78	174.7
SCD-B^E	**1.1**	0.037	**0.829**	25.98	171.1
SCD-B^D	2.8	**0.036**	**0.829**	**26.00**	**158.7**

6.1 Training from Scratch on Small Video Datasets

We evaluate pretraining performance on the widely used small-scale video generation benchmarks, TECO–Minecraft [82] and UCF-101 [69], as shown in Tables 1 and 2 (see also Appendix Table 7 for RealEstate10K results). We focus our comparison on diffusion-based methods, as non-diffusion baselines [4, 28, 63] yield substantially poorer visual quality on these datasets. On TECO–Minecraft, SCD-M attains the strongest overall generation quality—surpassing prior methods in LPIPS [95], SSIM, PSNR, and FVD [72]—while delivering more than a 4× reduction in inference latency, measured as wall-clock seconds per frame (Sec/F, lower is better) on a single H100 GPU. Similarly, on UCF-101 dataset, SCD-B attains better or on par performance on all metrics and delivers more than 2× inference speedup relative to existing approaches.

Furthermore, we observe that having more layers for the causal encoder yields modest latency overhead but consistently improves generation quality, as shown by SCD-M$^{\mathrm{E}}$ in Table 1 and SCD-B$^{\mathrm{E}}$ in Table 2 (see Appendix Table 8 for detailed model configurations). In addition, enlarging the diffusion decoder further boosts quality but incurs a substantial drop in inference speed. See SCD-M$^{\mathrm{D}}$ Table 1 and SCD-B$^{\mathrm{D}}$ in Table 2.

We further ablate the design choices of SCD and draw three conclusions. (1) Amortizing computation with a single once-per-frame encoder pass and multiple denoising passes significantly accelerates training (see Appendix Table 5 and Fig. 14). (2) When providing the causal context c_i and noisy frame x_i to the diffusion decoder, frame-wise concatenation consistently yields better performance than channel-wise

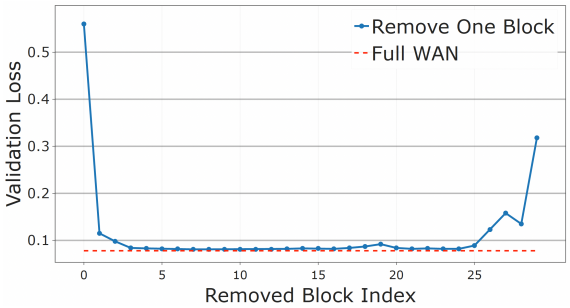


Figure 7. **The importance of transformer layers via leave-one-out inference.** We separately remove each layer in WAN2.1 T2V-1.3B and calculate the validation diffusion loss averaged across 5 noise levels. Results inform us which layers are important in finetuning.

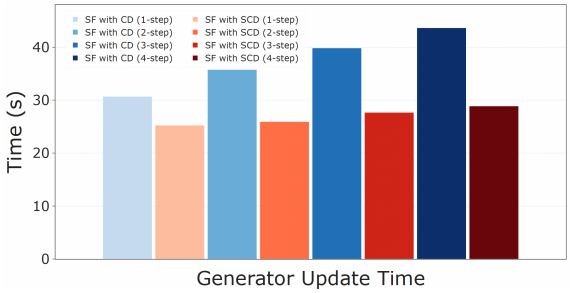


Figure 8. **Fine-tuning efficiency comparison.** Measured by per-iteration training time, SCD achieves superior training efficiency than causal diffusion baselines when performing frame-wise sequential rollout distribution matching training.

concatenation (Appendix Table 4). (3) Injecting noise into the context via Eq. 7 improves robustness and generation quality (Appendix Table 6). We refer readers to Appendix C and Appendix D for additional experimental details.

6.2 Fine-Tuning Pretrained T2V Diffusion Model

Architecture Adaptation. Recall from §5.3 that the causal encoder SCD performs next-frame prediction by taking the previously generated frame x_{i-1} at time $i-1$ as input to produce the context c_i at time i. In contrast, standard video diffusers require a noisy frame x_i^t at time i as input. Empirically, we find that this architectural mismatch prevents reliable transfer of the capabilities learned by a pretrained T2V model to our SCD architecture.

To bridge this gap, we align our causal encoder's input distribution with that of the pretrained video diffusion model. Specifically, during training, we feed a corrupted current frame x_i^t with high noise levels (top 20%) into the encoder, whereas during inference we use pure Gaussian noise as the input. This reparameterization preserves our decoupled design while matching the teacher's input distribution, enabling stable and effective fine-tuning.

Moreover, we observe that the learned feature distributions across different layers of a pretrained video diffusion model [79] differ substantially from the intended functionality of our SCD design. As a result, a straightforward layer decomposition—treating early layers as the causal encoder and the remaining late layers as the diffusion decoder —introduces a large domain gap that undermines the pretrained model's knowledge. To identify which layers are essential for generation quality, we conduct a leave-one-out analysis shown in Fig. 7. The results indicate that the earliest and latest layers contribute most to generation performance, whereas removing middle layers has much smaller negative impact. Motivated by this finding, we designate the first 25 layers of the pretrained 30-layer video diffusion model [79] as the causal encoder, and combine its first 5 and last 5 layers to form the diffusion decoder, resulting in a total of 35 layers in our SCD architecture (see Appendix Tables 9 and 10 for detailed training hyperparameters).

Training and Distillation. We apply the above architecture adaptation techniques to fine-tune our SCD model using a conditional flow-matching loss and a teacher-forcing training strategy. To further obtain a few-step diffusion decoder, we adopt a self-forcing–style distillation approach [34, 88, 89] and perform a full self-rollout of both the encoder and decoder, aligning the distribution of the generated samples with that of the pretrained bidirectional video diffusion teacher model. Additional training details are provided in Appendix E.

Results and analysis. Table 3 reports text-to-video results on VBench [35]. Throughput and latency are measured on one H100 80 GB GPU. Similar to the first-frame enhancement strategy of [85], we allocate extra compute to the initial frame and explicitly charge this overhead in throughput accounting. Despite the architecture mismatch that typically penalizes quality when moving from a decoder-only architecture to a decoupled encoder–decoder, SCD (1.6B) achieves strong performance while being **$\sim$1.3$\times$ faster** than the frame-wise Self Forcing baseline (11.1 vs. 8.9 FPS) with **$\sim$35% lower latency** (0.29 vs. 0.45 s). The total VBench score remains competitive (84.03 vs. 84.26), with similar quality and slightly lower semantic alignment, which we mainly attribute to the inevitable architectural mismatch. Compared to other AR baselines, SCD attains the highest throughput by a large margin (e.g., 11.1 vs. 6.7 FPS for Pyramid Flow), and is *$>$10$\times$ faster* than a strong non-causal diffusion model (Wan 2.1 at 0.78 FPS) while producing comparable overall scores. Figure 9 shows qualitative *I2V* examples from our finetuned model, where high visual quality and temporal consistency are observed with substantially lower inference cost (see Appendix Fig. 15 for temporal consistency visualization). Beyond inference efficiency, SCD enjoys high efficiency in rollout distribution matching training. As demonstrated in Figure 8, SCD achieves 20% higher

Table 3. **Text-to-Video quantitative comparison on VBench.** Models have similar parameter sizes and resolutions. Throughput (FPS) ↑ and latency (s) ↓ measured with batch size 1 on **1×H100 80 GB**. Higher is better for Total/Quality/Semantic scores ↑.

Model	#Params	Resolution	Throughput (FPS) ↑	Latency (s) ↓	Evaluation scores ↑		
					Total Score	Quality Score	Semantic Score
Bidirectional diffusion models							
LTX-Video [27]	1.9B	768×512	8.98	13.5	80.00	82.30	70.79
Wan2.1 [79]	1.3B	832×480	0.78	103	84.26	85.30	80.09
Frame-wise autoregressive models							
NOVA [16]	0.6B	768×480	0.88	4.1	80.12	80.39	79.05
Pyramid Flow [36]	2B	640×384	6.7	2.5	81.72	84.74	69.62
Self Forcing [34]	1.3B	832×480	<u>8.9</u>	<u>0.45</u>	**84.26**	**85.25**	**80.30**
SCD (Ours)	1.6B	832×480	**11.1**	**0.29**	<u>84.03</u>	<u>85.14</u>	<u>79.60</u>

Figure 9. **Image-to-Video qualitative samples from SCD fine-tuned from WAN 1.3B.** SCD preserves layout and motion while reducing per-frame compute; full video samples and baseline comparison are provided in the supplementary material.

training efficiency than Self Forcing in single-step rollout training, with marginal overhead in multi-step rollouts, indicating that SCD is more suitable for rollout training than full causal models.

7 Conclusion

Through probing and finetuning experiments, we identify two regularities in causal video diffusion models. Firstly, middle-layer features of the denoiser exhibit strong consistency across denoising steps. Secondly, cross-frame attention naturally becomes sparse with depth. Building on these insights, we design Separable Causal Diffusion (SCD), a novel architecture that decouples temporal reasoning from iterative denoising. Across synthetic and real video benchmarks, Separable Causal Diffusion leads to substantial computational speedups while preserving generation quality. Future work includes exploring the scaling law of next-frame denoising encoder against that of language models, exploiting the efficiency of SCD in roll-out based training frameworks, and integrating pre-trained reasoners and denoisers that lie in different latent spaces.

Limitations Our decoupling assumes that temporal reasoning can be amortized across denoising steps and that deeper layers are predominantly intra-frame—in densely pretrained models, both claims are approximations. (i) Step-wise invariance weakens near the end of the trajectory: the similarity between middle-layer features in the last 10 denoising steps and the first 40 drops to about 0.8 (Fig. 2), indicating that a single causal pass cannot fully substitute the evolving mid-layer dynamics. (ii) Deep layers retain a small but non-zero cross-frame attention mass (Fig. 4). These residual couplings plausibly account for the slight quality gap relative to fully causal-attention baselines at high resolution (Table 3). Closing this gap may require more complicated architectural design to restore the missing dependencies while preserving the efficiency gain of separable temporal reasoning.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. ViViT: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6836–6846, 2021. 2

[2] Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block Diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations (ICLR)*, 2025. 14

[3] Marianne Arriola, Yair Schiff, Hao Phung, Aaron Gokaslan, and Volodymyr Kuleshov. Encoder-decoder diffusion language models for efficient training and inference. *arXiv preprint arXiv:2510.22852*, 2025. 3

[4] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. FitVid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021. 6

[5] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 1

[6] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, Yuanzhen Li, Michael Rubinstein, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. Lumiere: A space-time diffusion model for video generation. In *ACM SIGGRAPH Asia 2024 Conference Papers*. ACM, 2024. 1, 2

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning (ICML)*, pages 813–824. PMLR, 2021. 2

[8] Yue Cao, Yuxin Cheng, Shusheng Yang, Siming Zhu, and Chenfei Wu. MAGI-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025. 2

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017. 2

[10] Boyuan Chen, Yilun Du, Diego Martí Monsó, Max Simchowitz, Vincent Sitzmann, and Russ Tedrake. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 24081–24125, 2024. 2, 3, 14

[11] Guoxiong Chen, Zerun Liang, Yidong Han, and Yiyang Zhang. SkyReels-V2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025. 2, 6

[12] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. VideoCrafter1: Open diffusion models for high-quality video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7310–7320, 2024. Tech Report updated to CVPR 2024 acceptance as VideoCrafter2 base. 2

[13] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformers for photorealistic text-to-image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024. 1

[14] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025. 17

[15] Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-Forcing++: Towards minute-scale high-quality video generation. *arXiv preprint arXiv:2510.02283*, 2025. 2, 14

[16] Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. In *International Conference on Learning Representations (ICLR)*, 2025. 2, 8

[17] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, pages 12606–12633. PMLR, 2024. 1

[18] Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. In *International Conference on Learning Representations (ICLR)*, 2025. 2

[19] Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. In *International Conference on Learning Representations (ICLR)*, 2025. 2

[20] FastVideo Team. FastVideo CausalWan2.2-I2V-A14B-Preview-Diffusers. `https://huggingface.co/FastVideo/CausalWan2.2-I2V-A14B-Preview-Diffusers`, 2025. Hugging Face Model Card. 14

[21] Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, Xunsong Li, Yifu Li, Shanchuan Lin, Zhijie Lin, Jiawei Liu, Shu Liu, Xiaonan Nie, Zhiwu Qing, Yuxi Ren, Limin Sun, Zhi Tian, Rui Wang, Sen Wang, Guoqiang Wei, Guohong Wu, Jie Wu, Ruiqi Xia, Fei Xiao, Xuefeng Xiao, Jiangqiao Yan, Ceyuan Yang, Jianchao Yang, Runkai Yang, Tao Yang, Yihang Yang, Ziyu Ye, Xuejiao Zeng, Yan Zeng, Heng Zhang, Yang Zhao, Xiaozheng Zheng, Peihao Zhu, Jiaxin Zou, and Feilong Zuo. SeeDance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025. 2

[22] Google DeepMind. Veo: A text-to-video generation system.

Technical report, Google DeepMind, 2024. Technical Report. 2

[23] Yuchao Gu, Weijia Mao, and Mike Zheng Shou. Long-context autoregressive video modeling with next-frame prediction. *arXiv preprint arXiv:2503.19325*, 2025. 1, 2, 3, 6, 17, 18

[24] Xiaoliu Guan, Lielin Jiang, Hanqi Chen, Xu Zhang, Jiaxing Yan, Guanzhong Wang, Yi Liu, Zetao Zhang, and Yu Wu. Forecasting when to forecast: Accelerating diffusion models with confidence-gated Taylor. *Knowledge-Based Systems*, 330:114635, 2025. 3

[25] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. In *International Conference on Learning Representations (ICLR)*, 2024. 1

[26] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Fei-Fei Li, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *European Conference on Computer Vision (ECCV)*, pages 393–411. Springer, 2024. 2

[27] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. LTX-Video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2025. 8

[28] Curtis Hawthorne, Andrew Jaegle, Cătălina Cangea, Sebastian Borgeaud, Charlie Nash, Mateusz Malinowski, Sander Dieleman, Oriol Vinyals, Matthew Botvinick, Ian Simon, Hannah Sheahan, Neil Zeghidour, Jean-Baptiste Alayrac, Joao Carreira, and Jesse Engel. General-purpose, long-context autoregressive modeling with Perceiver AR. In *International Conference on Machine Learning (ICML)*, pages 8535–8558. PMLR, 2022. 6

[29] Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. DiffusionBERT: Improving generative masked language models with diffusion models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4521–4534, 2023. 1

[30] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6840–6851, 2020. 1

[31] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 1

[32] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8633–8646, 2022. 1

[33] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *Transactions on Machine Learning Research (TMLR)*, 2022. 6

[34] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 1, 2, 3, 6, 7, 8, 14, 18

[35] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21807–21818, 2024. 7

[36] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *International Conference on Learning Representations (ICLR)*, 2025. 8

[37] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, 2014. 2

[38] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, Krishna Somandepalli, Hassan Akbari, Yair Alon, Yong Cheng, Joshua V. Dillon, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, Mikhail Sirotenko, Kihyuk Sohn, Xuan Yang, Hartwig Adam, Ming-Hsuan Yang, Irfan Essa, Huisheng Wang, David A. Ross, Bryan Seybold, and Lu Jiang. VideoPoet: A large language model for zero-shot video generation. In *International Conference on Machine Learning (ICML)*, pages 25105–25124. PMLR, 2024. 1, 2

[39] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations (ICLR)*, 2021. 1

[40] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for end-to-end tuning with latent diffusion transformers. *arXiv preprint arXiv:2504.10483*, 2025. 17

[41] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2, 5

[42] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7083–7093, 2019. 2

[43] Shanchuan Lin, Ceyuan Yang, Hao He, Jianwen Jiang, Yuxi Ren, Xin Xia, Yang Zhao, Xuefeng Xiao, and Lu Jiang. Autoregressive adversarial post-training for real-time interactive video generation. *arXiv preprint arXiv:2506.09350*, 2025. 14

[44] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. 1

[45] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D. Plumbley. Audi-

oLDM: Text-to-audio generation with latent diffusion models. In *International Conference on Machine Learning (ICML)*, pages 21450–21474. PMLR, 2023. 1

[46] Haozhe Liu, Zijian Zhou, Shikun Liu, Mengmeng Xu, Yanping Xie, Xiao Han, Juan Camilo Perez, Ding Liu, Kumara Kahatapitiya, Menglin Jia, Jui-Chieh Wu, Sen He, Tao Xiang, Jürgen Schmidhuber, and Juan-Manuel Perez-Rua. MarDini: Masked autoregressive diffusion for video generation at scale. *Transactions on Machine Learning Research (TMLR)*, 2025. 2

[47] Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161*, 2025. 14

[48] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3202–3211, 2022. 2

[49] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 18

[50] Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q. Weinberger. Latent diffusion for language generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 56998–57025, 2023. 1

[51] Xinyin Ma, Gongfan Fang, and Xinchao Wang. DeepCache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15762–15772, 2024. 3

[52] Sicheng Mo, Thao Nguyen, Xun Huang, Siddharth Srinivasan Iyer, Yijun Li, Yuchen Liu, Abhishek Tandon, Eli Shechtman, Krishna Kumar Singh, Yong Jae Lee, Bolei Zhou, and Yuheng Li. X-Fusion: Introducing new modality to frozen large language models. *arXiv preprint arXiv:2504.20996*, 2025. 2

[53] Haomiao Ni, Changhao Shi, Kai Li, Sharon X. Huang, and Martin Renqiang Min. Conditional image-to-video generation with latent flow diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18444–18455, 2023. 6

[54] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. LLaDA: Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025. 1

[55] OpenAI. DALL·E 3 system card, 2023. OpenAI System Card. 1

[56] Yuta Oshima, Shohei Taniguchi, Masahiro Suzuki, and Yutaka Matsuo. SSM meets video diffusion models: Efficient long-term video generation with selective state spaces. *arXiv preprint arXiv:2403.07711*, 2024. 2

[57] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, 2023. 6, 18

[58] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024. 1

[59] 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. 2

[60] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[61] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

[62] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 36479–36494, 2022. 1

[63] Vaibhav Saxena, Jimmy Ba, and Danijar Hafner. Clockwork variational autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 29004–29016, 2021. 6

[64] Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. FORA: Fast-forward caching in diffusion transformer acceleration. *arXiv preprint arXiv:2407.01425*, 2024. 3

[65] Weijia Shi, Xiaochuang Han, Chunting Zhou, Weixin Liang, Xi Victoria Lin, Luke Zettlemoyer, and Lili Yu. LMFusion: Adapting pretrained language models for multimodal generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[66] Joonghyuk Shin, Zhengqi Li, Richard Zhang, Jun-Yan Zhu, Jaesik Park, Eli Shechtman, and Xun Huang. MotionStream: Real-time video generation with interactive motion controls. *arXiv preprint arXiv:2511.01266*, 2025. 14

[67] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 2

[68] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-Guided video diffusion. *arXiv preprint arXiv:2502.06764*, 2025. 3, 6

[69] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human action classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6, 17

[70] Shengbang Tong, David Fan, Jiachen Li, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. MetaMorph: Multimodal understanding and generation via instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 2

[71] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, 2018. 2

[72] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation. In *International Conference on Learning Representations (ICLR) Workshops*, 2019. 6

[73] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 23371–23385, 2022. 6, 17

[74] Wan-AI. Wan 2.1 T2V-1.3B: Open-source text-to-video model. https://huggingface.co/Wan-AI/Wan2.1-T2V-1.3B, 2025. Hugging Face Model Card. 3

[75] Jingjing Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Zhijie Ji, Yingya Gu, Hang Chen, Chaoyue Wu, Xinyu Wen, and Xinyu Liu. ModelScope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 1

[76] Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. DDT: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025. 3

[77] Wenhao Wang and Yi Yang. VidProM: A million-scale real prompt-gallery dataset for text-to-video diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024. 18

[78] WanTeam. Wan 2.1 T2V-1.3B: Open-source text-to-video model. *arXiv preprint arXiv:2503.00123*, 2025. 3, 18

[79] WanTeam, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, and Jinkai Wang. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1, 2, 3, 7, 8, 18

[80] Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse Video-Gen: Accelerating video diffusion transformers with spatial-temporal sparsity. In *International Conference on Machine Learning (ICML)*, 2025. 2

[81] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video generation using VQ-VAE and transformers. In *arXiv preprint arXiv:2104.10157*, 2021. 1

[82] Wilson Yan, Danijar Hafner, Stephen James, and Pieter Abbeel. Temporally consistent transformers for video generation. In *International Conference on Machine Learning (ICML)*, pages 39062–39098. PMLR, 2023. 6, 17

[83] Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. LongLive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025. 2, 14

[84] Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, Jianfei Chen, Song Han, Kurt Keutzer, and Ion Stoica. Sparse VideoGen2: Accelerate video generation with sparse attention via semantic-aware permutation. *arXiv preprint arXiv:2505.18875*, 2025. 2

[85] Yongqi Yang, Huayang Huang, Xu Peng, Xiaobin Hu, Donghao Luo, Jiangning Zhang, Chengjie Wang, and Yu Wu. Towards one-step causal video generation via adversarial self-distillation. *arXiv preprint arXiv:2511.01419*, 2025. 7

[86] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations (ICLR)*, 2025. 1

[87] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15703–15712, 2025. 17

[88] Tianwei Yin, Michaël Gharbi, Taesung Park, Eli Shechtman, Richard Zhang, William T. Freeman, and Frédo Durand. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 47455–47487, 2024. 7

[89] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo 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 (CVPR)*, pages 6613–6623, 2024. 7

[90] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22963–22974, 2025. 1, 2, 3, 6

[91] Hu Yu, Biao Gong, Hangjie Yuan, DanDan Zheng, Weilong Chai, Jingdong Chen, Kecheng Zheng, and Feng Zhao. VideoMAR: Autoregressive video generation with continuous tokens. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 2

[92] Hui Zhang, Tingwei Gao, Jie Shao, and Zuxuan Wu. BlockDance: Reuse structurally similar spatio-temporal features to accelerate diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12891–12900, 2025. 3

[93] Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. In *International Conference on Machine Learning (ICML)*. PMLR, 2025. 2

[94] Peiyuan Zhang, Haofeng Huang, Yongqi Chen, Will Lin, Zhengzhong Liu, Ion Stoica, Eric Xing, and Hao Zhang. VSA: Faster video diffusion with trainable sparse attention. *arXiv preprint arXiv:2505.13389*, 2025. 2

[95] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 6

[96] Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T. Freeman,

and Hao Tan. Test-time training done right. *arXiv preprint arXiv:2505.23884*, 2025. 5

[97] Zhicheng Zhang, Junyao Hu, Wentao Cheng, Danda Paudel, and Jufeng Yang. ExtDM: Distribution extrapolation diffusion model for video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19310–19320, 2024. 17

[98] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *International Conference on Learning Representations (ICLR)*, 2025. 2

[99] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics (SIGGRAPH)*, 37(4), 2018. 17

A Additional Analysis of Uncovering Causal Separability

A.1 Redundancy across Denoising Steps

Feature similarity across denoising steps and depth. We extend the analysis in Fig. 2 to multiple depths of an autoregressively fine-tuned WAN-2.1 T2V-1.3B model. For a fixed set of prompts/seeds, we roll out the model for 50 denoising steps and, at every transformer block ℓ and step s, record the hidden features. We then compute a step–step *mean-squared-error (MSE) distance* matrix $\mathbf{S}_\ell \in \mathbb{R}^{T\times T}$ with entries $[\mathbf{S}_\ell]_{s,s'} = \|f_{\ell,s} - f_{\ell,s'}\|_2^2$, where $f_{\ell,s}$ denotes the layer-ℓ features at step s. Representative matrices for $\ell \in \{10, 15, 20, 25, 28, 29\}$ are shown in Fig. 10a and Fig. 10b

Layers 10–25 exhibit pronounced step-wise invariance: their similarity maps contain broad, near-uniform high-value bands (also visible at the 15th layer in the main paper), indicating that middle/early denoiser blocks repeatedly recompute almost the same features across the diffusion trajectory. In contrast, the last two layers ($\ell = 28, 29$) display markedly lower and more step-dependent similarity, consistent with these blocks performing step-specific, intra-frame rendering.

To further verify the redundancy observation, we fine-tune the baseline with a skip-layer design, in which the majority of denoising steps skip the middle-layers computation.

To further verify the redundancy finding, we fine-tune the causal baseline with a *skip-layer* schedule in which most denoising steps bypass the middle of the network. Concretely, only the first five denoising steps run the full denoiser; all subsequent steps traverse just a short *prefix* of 5 early layers and a *suffix* of 10 late layers, skipping the middle chunk. We retain the prefix because, as shown in Fig. 7 in the main text, early layers are particularly important during fine-tuning. The outcome (Fig. 3 in the main text) is that the skipped model produces high-quality videos and recovers the visual fidelity of the fully causal baseline, validating that most cross-step computation in early/middle layers is redundant and can be shared. This experiment is designed to test the observation in the simplest setting. In later SCD fine-tuning, we adopt a more aggressive recipe that also works: the first 25 layers run *once per frame* (amortized across steps), and only the final 5+5 layers participate in per-step denoising under our SCD design.

A.2 Evidence on Other Models

Beyond the WAN-2.1 (1.3B) model, we also evaluate an open-source, open-weight **Self-Forcing** (SF) 1.3B model [34]—a few-step, *block-autoregressive* student distilled from a bidirectional teacher that predicts three latent frames per block. We chose this model deliberately for two reasons. **(i) Block-autoregression** is a widely used generative pattern in contemporary video systems, and even in emerging language diffusion models [2], so validating our analysis on a block-AR student makes the conclusions relevant to a broad class of architectures [43, 66, 83]. **(ii) Self-forcing** is the prevailing post-training recipe for large autoregressive video models, bridging the train–test gap and distilling many-step teachers into efficient few-step samplers [15, 20, 34, 47]. By observing our claims on both a many-step WAN and a few-step, block-autoregressive, self-forced student, we cover complementary ends of the design space; the aligned observations across these regimes would substantially strengthen the generality of our claims.

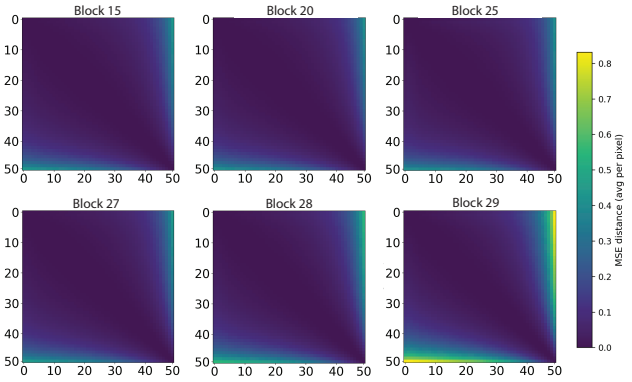
Observations. We see the same patterns as in §4. Early and middle *layers* produce very similar features across denoising steps. PCA views change little with the step index and already capture global structure in the *first block*. Deeper layers are temporally sparse and focus on intra-frame rendering. These results indicate that the separability trends hold for both a 50-step WAN and a distilled 4-step *block-autoregressive* self-forcing model.

Evidence on Diffusion Forcing with 3D UNet. To further validate the generality of our observations beyond Transformer-based architectures and teacher-forcing training, we analyze a 3D UNet trained with Diffusion Forcing [10] on Minecraft. Despite the substantially different backbone (UNet vs. Transformer) and training objective (Diffusion Forcing vs. Teacher Forcing), we observe the same qualitative trends as shown in Fig. 13. Mid-layer representations stabilize early in denoising, exhibiting high cross-step cosine similarity and strong PCA subspace alignment. Moreover, deep denoiser layers attend sparsely to the context frames while focusing primarily on intra-frame structure. We also demonstrate this phenomenon is consistent across denoising timesteps (e.g., t=50 vs. t=99). These results provide strong evidence that the causal separability we observe is a fundamental property of causal video diffusion rather than model-specific artifacts.

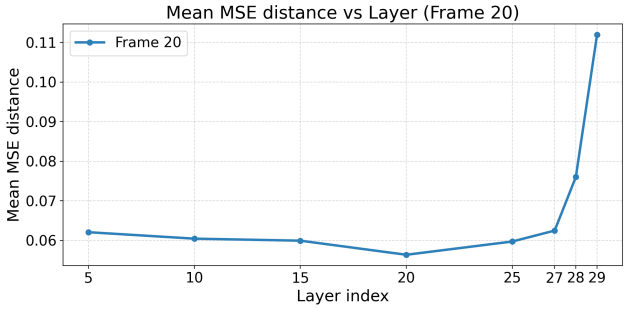
B Ablation Studies

B.1 Encoder-Decoder Interface

We compare two ways of providing the context latent c_t to the frame-wise diffusion decoder $\mathcal{D}_\theta$: (i) *Channel Concatenation* we concatenate c_t with the noisy frame tokens along the channel dimension, project back to the standard channel dimension with a linear layer, and feed it to the decoder as input; (ii) *Frame Concatenation* we prepend c_t as a prefix frame for the noisy frame, and then feed them into the decoder. This effectively positions c_t as the context frame of the current frame, performing self-attention together as a



(a) Step–step *MSE-distance* matrices for multiple layers.



(b) Mean MSE distance versus block index. Each value here represents the average value across the entire corresponding matrix on the left.

Figure 10. **Redundant computation across denoising steps.** (a) Step–step feature *MSE-distance* matrices of a fine-tuned AR WAN-2.1 T2V-1.3B model at several layer depths (layers 5, 10, 15, 20, 25, 28, 29). Middle layers (10–25) show broad, *low-distance* bands across all 50 denoising steps, indicating that their features are mostly invariant along the diffusion trajectory, whereas the last few layers exhibit more step-dependent distances. (b) A complementary per-layer summary: the average MSE distance across all pairs of denoising steps remains small in the early and middle layers, but increases dramatically in late layers. Together, these views support our claim that early/middle blocks perform largely redundant computation across denoising steps, while the deepest blocks remain step-specific for intra-frame rendering, motivating our design that amortizes the first 25 layers once per frame and reuses them across all denoising steps.

Table 4. **Ablations on the encoder-decoder interface.** Sequence fusion outperforms channel fusion; temporal RoPE is slightly better than identical (non-causal) RoPE. For simplicity of ablation, metrics are reported at 400k training steps, with the unified "144 context frames, 156 generated frames" evaluation setup.

Encoder-Decoder Interface	FVD ↓	LPIPS ↓
Channel dim.	25.4	0.231
Frame dim. with temporal RoPE	**24.8**	**0.219**
Frame dim. with identical RoPE	25.1	0.223

Table 5. **Amortized multi-sample decoding in training.** Encoder depth 8, decoder depth 4, as in our SCD-B and SCD-M models. For simplicity of ablation, metrics are reported at 400k training steps, with the unified "144 context frames, 156 generated frames" evaluation setup.

K	BP/clean batch	BP/noisy batch	noisy batch/s	FVD
1	$8+4\times1=12$	$8/1+4=12$	*22.0*	*23.9*
2	$8+4\times2=16$	$8/2+4=8$	*39.2*	*23.3*
4	$8+4\times4=24$	$8/4+4=6$	*63.0*	***23.1***

whole sequence. We also ablate the positional embedding applied to the context frame, either embedding it as "the last temporal frame" or "the current frame". As shown in Tab. 4, sequence fusion with positional embedding as a historical frame outperforms the alternatives.

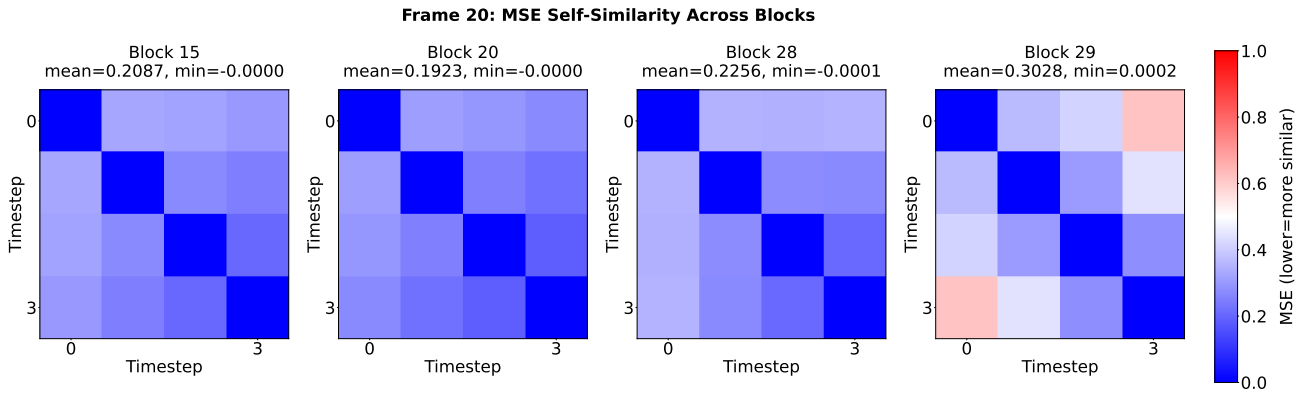
B.2 Training Noisy Batches: Amortized Multi-Sample Decoding

Because $\mathcal{E}_\phi$ consumes only clean history, it is noise-agnostic. We therefore perform one efficiency trick in training: for each batch of clean videos, we encode once per frame to obtain c_t, then draw K i.i.d. noise/timestep pairs for the current frame and run $\mathcal{D}_\theta$ K times, saving the amortized cost for learning each noisy batch. As Tab. 5 demonstrates, throughput of noisy batch increases with K, which also improves the training speed.

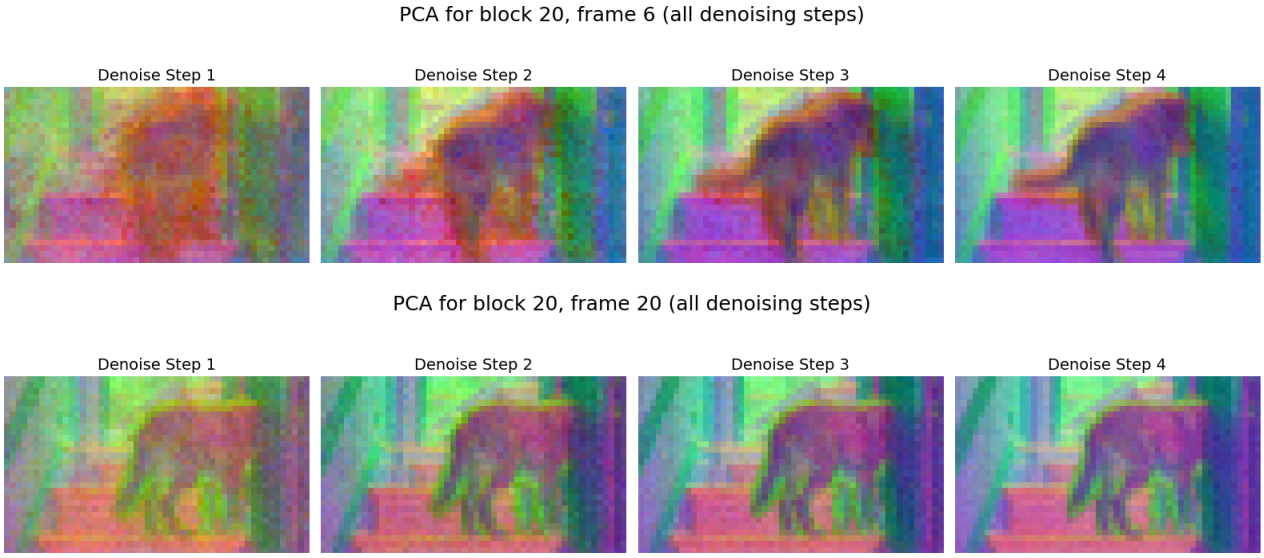
Training-time comparison. The main-table ablation in Sec. B.2 compares different K at matched optimization steps, which slightly favors larger K because each step processes more independently noised targets. To control for this, Fig. 14 re-plots LPIPS against *wall-clock training time*. Despite the heavier per-step compute, curves with $K{=}2$ and especially $K{=}4$ reach lower LPIPS than $K{=}1$ at the same elapsed time. This shows that amortized multi-sample decoding does more than just see more noise per step: reusing the once-per-frame encoder output across multiple noisy decoder calls yields a better optimization trajectory even under a fixed compute budget.

B.3 Noisy Context Latent: Context Corruption and CFG

We perturb only the context c_i by adding Gaussian noise $\tilde{c}_t = c_t + \eta_t\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$, where c_t is normalized to unit variance and η_t^2 is the noise variance. Table 6 reports two complementary ablations on TECO–Minecraft at 400k steps. On the *left*, we vary the training-time corruption strength $\eta_t \in \{0, 0.05, 0.10, 0.20, 0.50\}$ and evaluate without any inference-time classifier-free guidance (CFG) on c_t: moderate noise around $\eta_t = 0.05$ improves FVD while stronger corruption eventually hurts. On the *right*, we fix training-



(a) **Self-forcing: step–step feature similarity** across multiple layers (analogous to Fig. 2(a) in §4). Early/middle layers show broad high-similarity (small MSE) bands over denoising steps.



(b) **Self-forcing: PCA of activations** (combined view; analogous to Fig. 2(b) in §4). Principal components remain stable across denoising steps and across *blocks* in the block-autoregressive rollout.

Figure 11. **Evidence on another model family (self-forcing, 4-step, block-autoregressive).** We replicate the §4 analyses on a few-step self-forcing student. (a) Early/middle layers exhibit high step–step feature similarity. (b) PCA views confirm that principal directions stabilize early and change little across denoising steps and *blocks*.

time corruption at η_t=0.05 and ablate CFG using a negative branch with the same perturbation $\tilde{c}_t = c_t + 0.05\,\epsilon$ and guidance scale $\eta_{\text{cfg}} \in \{0.0, 1.0, 1.5, 2.0\}$; η_{cfg}=1.5 gives the best trade-off. Overall, modest training-time corruption plus a small CFG weight on the corrupted causal prior ($\eta_t \approx 0.05$, $\eta_{\text{cfg}} \approx 1.5$) yields the strongest long-horizon visual quality. Because these perturbations act only on the causal interface, they do not require re-caching any context frames.

Table 6. **Training-time causal corruption and test-time CFG with corruption (Minecraft, 400k steps).** Left: vary η_t at training and evaluate with no CFG on c_t (η_{cfg}=0). Right: fix η_t=0.05 at training and sweep inference-time CFG scale η_{cfg} on a corrupted causal prior $\tilde{c}_t = c_t + 0.05\,\epsilon$.

Training corruption			**Inference CFG with corruption**		
Noise η_t	FVD ↓	LPIPS ↓	η_{cfg}	FVD ↓	LPIPS ↓
0.00	24.8	0.199	0.0	24.2	0.254
0.05	**23.8**	0.195	1.0	23.1	0.223
0.10	24.5	0.195	**1.5**	**22.3**	**0.219**
0.20	25.1	**0.191**	2.0	22.7	0.221
0.50	27.6	0.199	(η_t=0.05)		

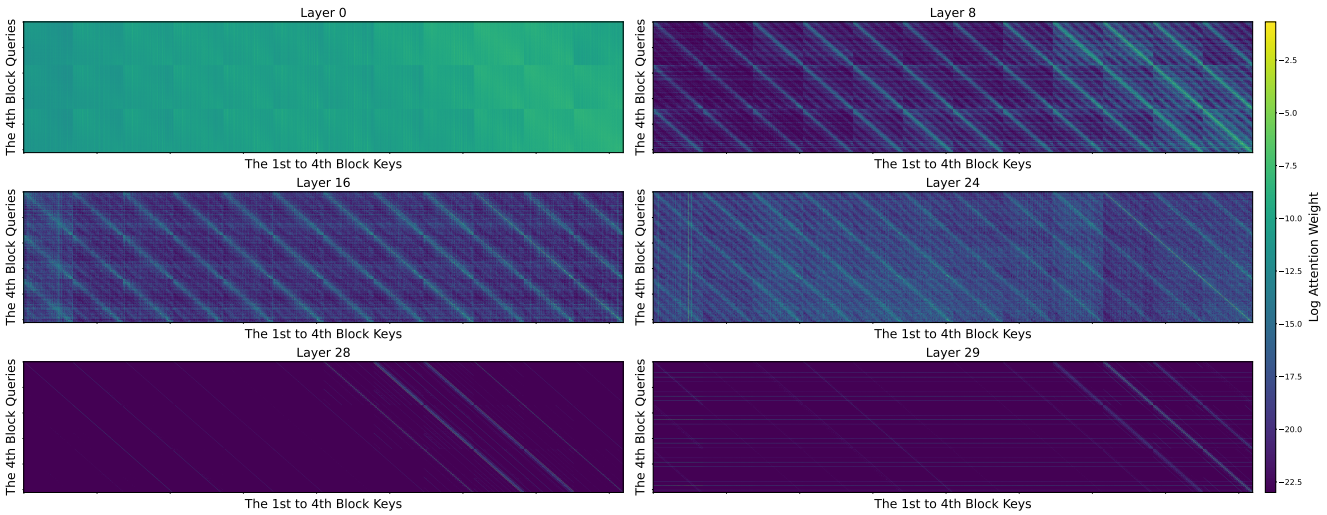


Figure 12. **Evidence on another model family (self-forcing, 4-step, block-autoregressive).** We replicate the §4 analyses on a few-step self-forcing student. As with the multi-step models, its deeper layers place minimal attention on past blocks, again revealing strong temporal sparsity.

C Additional Experimental Setup

C.1 Datasets

TECO-Minecraft[82]. We adopt the long-context, action-conditioned prediction setup popularized by TECO [82]. Each video contains 300 frames, with 128×128 resolution, with per-frame action annotations. We evaluate on 256 video clips; for long-horizon quality (FVD), each clip supplies 36 ground-truth context frames followed by 264 generated frames; for frame-wise metrics (LPIPS, SSIM, PSNR), each clip supplies 144 observed frames followed by 156 generated frames. This set-up exactly aligns with TECO.

UCF-101[69]. We use the UCF-101 dataset to demonstrate the models' capability in real-world motion. This dataset comprises ∼13K unconstrained, unconditional action videos. Following MCVD [73]/ExtDM [97] and FAR [23], we randomly sample 256 videos and, for each, draw 100 stochastic trajectories. Pixel metrics (LPIPS/SSIM/PSNR) are computed best-of-100 per video, and FVD is averaged over all 100 trajectories.

RealEstate10K[99]. We also perform unconditional generation experiments on an auxiliary benchmark, RealEstate10K, a dataset consisting of real-world indoor scenes. While this dataset is predominantly used in 3D tasks, we use it because it is a relatively small real-world dataset, where pretraining is feasible for our experiments. We use a resolution of 256×256. Since our model is orthogonal to camera-pose conditioning techniques, we simply perform unconditional prediction tasks with 16 context frames and 48 generated frames. Results are shown in Tab 7.

Table 7. **Unconditional generation on RealEstate10K (256^2, 16→48).** Numbers are measured at 400k training steps with 50 denoising steps.

Model	Sec/F	16→48			
		LPIPS↓	SSIM↑	PSNR↑	FVD↓
Causal DiT-B	1.07	0.172	0.594	19.35	101.64
SCD-B	0.44	0.142	0.616	19.67	102.83
SCD-B^E	0.45	<u>0.139</u>	<u>0.622</u>	<u>19.95</u>	<u>101.61</u>
SCD-B^D	1.03	**0.135**	**0.623**	**20.01**	**85.12**

Tokenizer. Following the common practice, we compress video frames into video latent, and apply diffusion models in the corresponding latent space. To compress a video, we adopt a series of VAE and DCAE models [14]. For Minecraft and UCF, we use the DCAE trained in FAR [23] ; for RealEstate10K, we adopt the E2E-VAE tokenizer from [40] finetuned from VA-VAE [87].

D Model and Training Details

D.1 Design Details of Separable Causal Diffusion

Let $\mathcal{E}_\phi$ denote the causal reasoning encoder and $\mathcal{D}_\theta$ the frame-wise diffusion decoder. The encoder runs once per video frame outside the denoising loop to produce a latent c_t, summarizing the temporal context. Then, the decoder denoises each frame with multiple diffusion steps, conditioned on c_t:

$$c_t = \mathcal{E}_\phi(x_{<t}, a_{\le t}) \quad , \qquad \hat{v}_t^{(s)} = \mathcal{D}_\theta\big(x_t^{(s)},\, s,\, c_t\big).$$

Here $x_t^{(s)}$ are noisy frame latents at step s, and $\hat{v}_t^{(s)}$ is the predicted velocity/score used by the sampler. $\mathcal{E}_\phi$ uses

Table 8. **Model variants and depth split.** Depth is split into ℓ causal blocks and m diffusion blocks. BP/frame $= \ell + S \cdot m$ with S=50.

Model	#Blocks	Hidden	#Heads	Params	BP / frame
DiT-B	12	768	12	131M	600
SCD-B	8+4	768	12	132M	208
SCD-B^E	12+4	768	12	174M	212
SCD-B^D	8+12	768	12	217M	608
FAR-M	12	1024	16	230M	600
SCD-M	8+4	1024	16	230M	208
SCD-M^E	12+4	1024	16	306M	212
SCD-M^D	8+12	1024	16	383M	608

frame-wise causal attention and KV caches; $\mathcal{D}_\theta$ performs intra-frame bidirectional attention. The amortized per-frame cost is therefore $O(\mathcal{E}_\phi) \; + \; S \cdot O(\mathcal{D}_\theta)$.

In our experiments, we choose $O(\mathcal{E}_\phi) \gg O(\mathcal{D}_\theta)$ for two reasons: 1) Empirically, we observe that a large portion of layer features are shareable across the denoising process (Fig.10). 2) The encoder's cost is amortized across multiple denoising steps, so it can be made larger without significantly sacrificing efficiency.

D.2 Architectures and Model Variants

We follow the Diffusion Transformer (DiT) structure [57] to implement the SCD neural network. We follow DiT's width/head configuration for hidden size and MLP. To compare with FAR [23], the SOTA model on Minecraft, we also adopt its FAR-M parametrization. Table 8 enumerates the variants used in our experiments and reports BP/frame under S=50.

D.3 Algorithmic Pipeline

We summarize the end-to-end training and inference procedures of Separable Causal Diffusion (SCD) in Algorithms 1 and 2.

Algorithm 1 SCD Training

Require: Videos $\mathbf{x}_{1:N}$ with controls $\mathbf{a}_{1:N}$, where N is the number of frames; temporal reasoning module $\mathcal{E}_\phi$; frame diffusion module $\mathcal{D}_\theta$; diffusion loss $\mathcal{L}$; noisy multi-batch size K.

repeat
  Choose target frame $i \in \{1, \dots, N\}$
  $c_i \leftarrow \mathcal{E}_\phi(\mathbf{x}_{<i}, \mathbf{a}_{\le i})$
  $\mathcal{L}_{\text{step}} \leftarrow 0$
  for $k = 1$ to K **do**
    Sample $t \sim \mathcal{U}[0,1]$, $\epsilon \sim \mathcal{N}(0, I)$
    $x_i^t \leftarrow (1-t)\, x_i + t\, \epsilon$
    $\hat{u} \leftarrow \mathcal{D}_\theta(x_i^t, t, c_i)$
    $\mathcal{L}_{\text{step}} \leftarrow \mathcal{L}_{\text{step}} + \mathcal{L}(\hat{u},\, x_i,\, \epsilon,\, t)$
  Take a gradient step on $\nabla_{\theta,\phi} \left(\mathcal{L}_{\text{step}}/K\right)$
until converged

Algorithm 2 SCD Generation by Roll-out over Frames

Require: Controls $\mathbf{a}_{1:N}$, where N is the number of frames to be generated; temporal reasoning module $\mathcal{E}_\phi$; frame diffusion module $\mathcal{D}_\theta$; sampler with T denoising steps and schedule $\{t_1, \dots, t_T\}$.

$\hat{\mathbf{x}} \leftarrow [\,]$ % generated frames buffer
for $i = 1, \dots, N$ **do**
  $c_i \leftarrow \mathcal{E}_\phi(\hat{\mathbf{x}}_{<i}, \mathbf{a}_{\le i})$ % AR context: previously generated frames
  Initialize $z^T \sim \mathcal{N}(0, I)$
  for $t = T, T-1, \dots, 1$ **do**
    $\hat{u} \leftarrow \mathcal{D}_\theta(z^t, t, c_i)$
    $z^{t-1} \leftarrow \textsc{Sampler}(z^t, \hat{u}, t)$
  $\hat{x}_i \leftarrow z^0$; append $\hat{x}_i$ to $\hat{\mathbf{x}}$
return $\hat{\mathbf{x}}_{1:N}$

E SCD Fine-tuning Details

This section describes fine-tuning details omitted from the main text. We first specify the teacher, data, and architecture adaptations used to convert a pretrained bidirectional video diffusion model into our Separable Causal Diffusion (SCD), which contains a causal encoder + a frame-wise diffusion decoder. We then detail the self-forcing rollout/distillation protocol used for post-training, and finally discuss capacity splits between encoder and decoder that highlight the flexibility of SCD.

Bidirectional Teacher. Unless otherwise noted, we fine-tune from a high-quality, bidirectional T2V checkpoint of WAN 2.1 T2V-1.3B [78, 79], whose weights are transplanted into our decoupled backbone (§E.1). All training lies in the latent spaces derived from the original VAE of WAN 2.1.

Datasets. We use the text prompts from a 1M subset of VidProM [77] following the same filtering process in [34]. For fine-tuning with diffusion loss with our architecture, we use 70k synthetic data generated by WAN 2.1 T2V-14B with the above text prompts. For self-rollout training, we use the full 1M text prompts as conditions.

Training Specifications. We jointly train the encoder and decoder with the conditional flow-matching objective (Eq. (1) in the main text). For time step distribution, following WAN, we employ the timestep shifting $t'(k,t) = \frac{kt}{1+(k-1)t}$ and the forward interpolation is given as $x_t = (1-t')x + t'\epsilon$, $\epsilon \sim \mathcal{N}(0, I), t \in \mathcal{U}(0,1)$. We use the AdamW [49] optimizer for all experiments. Detailed hyperparameters can be found in the Table 9 and Table 10.

E.1 Architecture Adaptation for Decoupling

As described in the main paper, our decoupled backbone implements once-per-frame temporal reasoning in a causal encoder $\mathcal{E}_\phi$ and iterative rendering in a light frame-wise diffusion decoder $\mathcal{D}_\theta$, which differs from the teacher architecture. Therefore, to align the two architectures, in practice we make two adaptations when initializing from a bidirectional teacher:

(i) Input reparameterization for the encoder. Pretrained video diffusers consume a *noisy* current frame at each denoising step, while our encoder must operate on last generated frame instead of current frame. During fine-tuning, we therefore feed the encoder a corrupted current frame x_i^t at relatively *high* noise levels (e.g., top 20% of the diffusion/flow schedule), and at inference we replace it with pure Gaussian noise. This aligns the encoder's input distribution with the teacher while preserving the decoupled compute pattern (once per frame for $\mathcal{E}_\phi$, multi-step for $\mathcal{D}_\theta$).

(ii) Layer decomposition. Directly treating early layers as encoder and late layers as decoder often harms generation performance from finetuning. As discussed in the main text, the early layers play an important role in converting model input scale to an internal model scale. Guided by leave-one-out loss probing, we allocate the first 25 layers to $\mathcal{E}_\phi$ and build $\mathcal{D}_\theta$ by combining the first 5 and last 5 layers.

E.2 Hyperparameters

Table 9. **Fine-tuning hyperparameters.**

Resolution / Frames	832×480 / 81 frames
Batch size	64
LR / WD / Optimizer	2×10^{-5} / 0.01 / AdamW$(0.9, 0.99)$
EMA decay	0.99
Time sampler	$\frac{5t}{1+4t}, t \sim \mathcal{U}(0,1)$

Table 10. **Self-Forcing rollout training hyperparameters.**

Resolution / Frames	832×480 / 81 frames
Teacher	WAN 2.1 14B
Teacher CFG	3.0
Critic initialization	WAN 2.1 1.3B
Batch size	64
Student LR / WD / Optimizer	2×10^{-6} / 0.01 / AdamW$(0.0, 0.99)$
Critic LR / WD / Optimizer	4×10^{-7} / 0.01 / AdamW$(0.0, 0.99)$
Student EMA decay	0.99
Critic/student update ratio	5
Time sampler	$\frac{5t}{1+4t}, t \sim \mathcal{U}(0,1)$

Throughput and latency. We report wall-clock throughput (FPS) and per-frame latency with batch size 1 on 1×H100 80 GB, charging the initial frame's extra compute. SCD fine-tuned from a strong T2V teacher achieves ∼11.1 FPS with 0.29 s latency at 832×480 while retaining competitive VBench scores; the self-forcing baseline at the same scale reaches 8.9 FPS and 0.45 s latency.(Table 11).

Table 11. **Frame-Autoregressive Text-to-Video on VBench (832×480, batch 1)**. Reported throughput includes first-frame overhead.

Model	FPS ↑	Latency (s) ↓	Total ↑	Quality / Semantic ↑
Self Forcing	8.9	0.45	84.26	85.25 / 80.30
SCD (Ours)	**11.1**	**0.29**	84.03	85.14 / 79.60

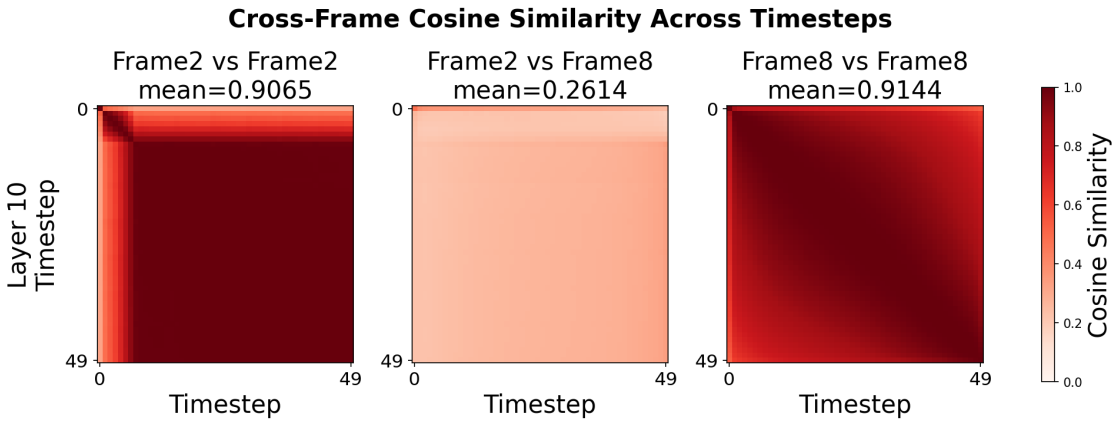
E.3 Flexibility of Separable Causal Diffusion

A practical benefit of SCD is that temporal reasoning capacity and per-frame rendering capacity can be traded *independently*. Let total depth be $\ell+m$ with encoder depth ℓ (causal reasoning, amortized once per frame) and decoder depth m (frame-wise denoising, repeated S steps in inference, where $S = 4$ in standard self-forcing settings). For fixed parameters:

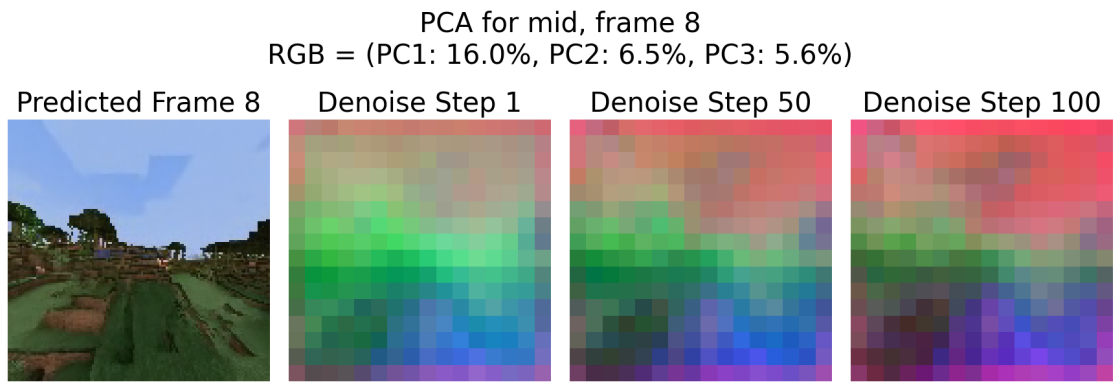
- **Encoder-heavier variants** slightly increase reasoning cost per frame but systematically improve motion/layout adherence and long-horizon stability; it only slightly reduce the throughput since the encoder runs once per frame.
- **Decoder-heavier variants** improve per-frame detail at the cost of $S\times m$ block passes per frame; quality gains can be notable when targeting high-fidelity or very few denoising steps, but latency rises accordingly.

F Additional Visualization

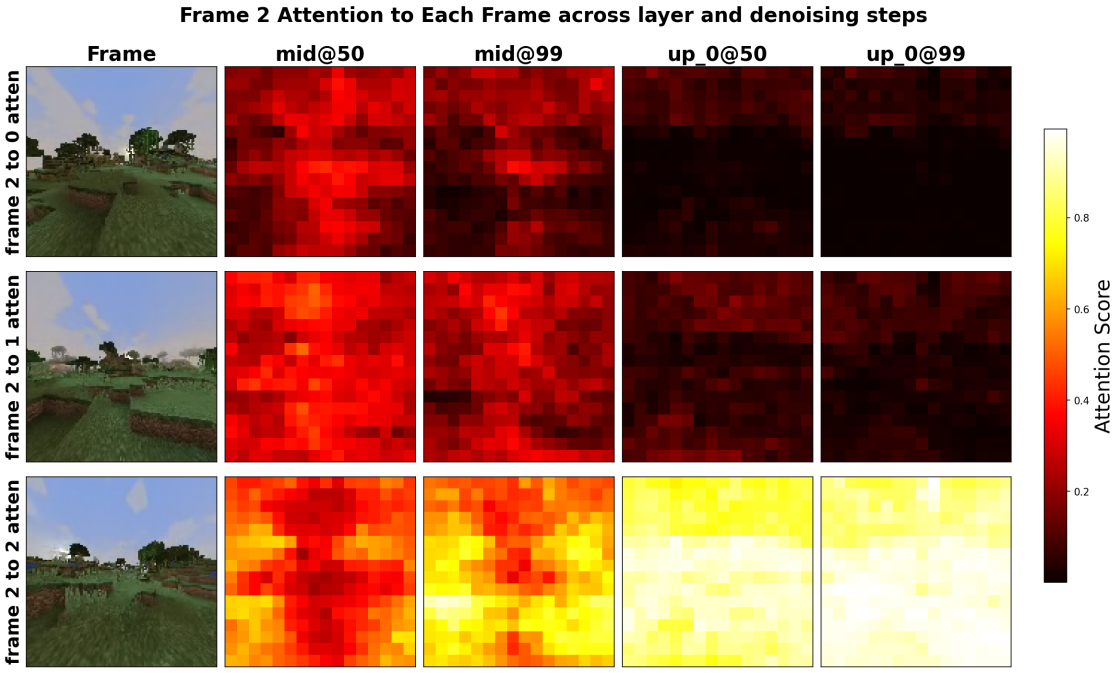
Temporal Consistency. Figure 15 qualitatively demonstrates the temporal consistency of Separable Causal Diffusion by showing frames generated across five seconds. Despite removing all decoder cross-frame attention and achieving 40% faster inference, SCD maintains near-identical temporal consistency to the baseline. Objects and backgrounds are well preserved across the generation horizon.



(a) Cross-step cosine similarity



(b) PCA subspace alignment



(c) Attention visualization

Figure 13. **Generality to a 3D UNet on Minecraft under Diffusion Forcing.** Despite the substantially different backbone (UNet vs. Transformer) and training objective (Diffusion Forcing vs. Teacher Forcing), we observe the same qualitative trends: mid-layer representations stabilize early in denoising, exhibiting (a) high cross-step cosine similarity and (b) strong PCA subspace alignment. Moreover, (c) deep denoiser layers attend sparsely to the context frames (rows 1–2) while focusing primarily on intra-frame structure (row 3). We demonstrate this phenomenon is consistent across denoising timesteps (e.g., t=50 vs. t=99).

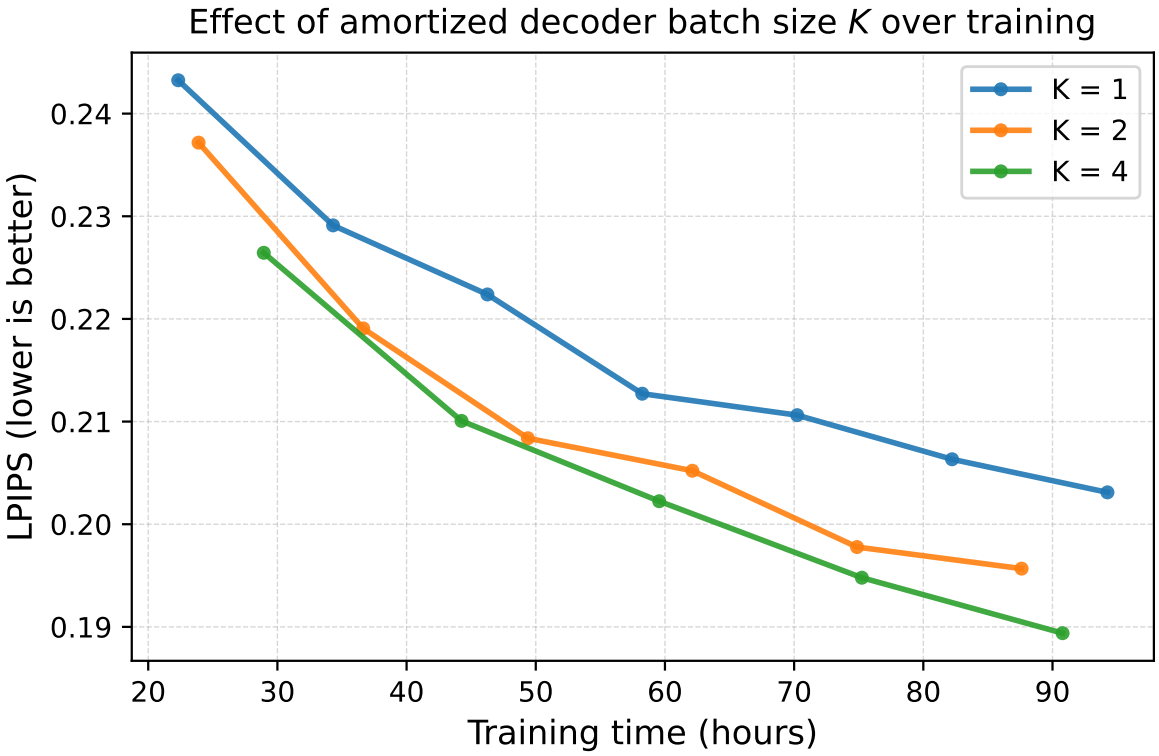


Figure 14. **Effect of amortized decoder batch size K at matched training time.** LPIPS on the validation set versus wall-clock training time (hours) for $K \in \{1, 2, 4\}$. Even when comparing at equal training time rather than equal optimization steps, larger K achieves lower LPIPS, indicating genuine gains from amortizing multiple noisy decoder samples per encoder pass.

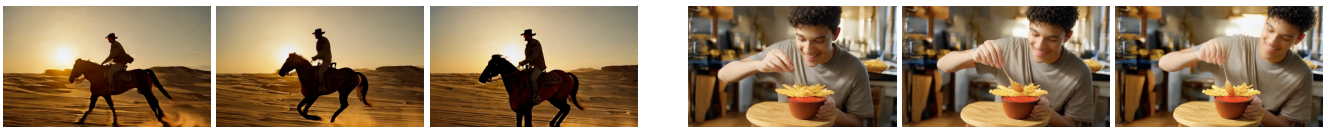


Figure 15. **Temporal consistency of SCD between the 0th and 84th generated frames.** Objects and backgrounds are preserved across the generation horizon, demonstrating that the encoder's context sufficiently captures inter-frame dependencies, making decoder cross-frame computation unnecessary.