VIDA: EXPLOITING TEMPORAL DEPENDENCIES FOR AUTOREGRESSIVE VIDEO DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive video distillation supervises predictions that also serve as context for subsequent generation. Causal self-rollout exposes the student to its own history, but a single re-noising level in Distribution Matching Distillation (DMD) couples the preservation of that context with the correction of generated content. We present **ViDA** (Video Distillation for Autoregressive generation), which incorporates temporal dependencies into DMD supervision through ordered score queries. Earlier chunks receive lower, nonzero corruption to retain context while remaining subject to correction. With both score estimators bidirectional and independent of the generator, the ordered query improves VBench Total by 0.75 over marginal-matched scalar queries, and by 0.64 and 1.40 over shuffled and reversed noise assignments. ViDA also replaces the separate fake-score network with a role-conditioned shared backbone, improving quality at a fixed update ratio. Combined with fewer fake updates, it outperforms the independent causal estimator at $r = 5$ at half the student post-training GPU-hours and 13.6% lower peak allocated memory. On five-second generation, the final 1.3B generator distilled from a 14B teacher reaches 85.62 VBench Total: 1.12 points above matched scalar querying within the same recipe and 1.38 above the locally evaluated official Self Forcing checkpoint. Evaluations on Cosmos and 60-second videos extend these comparisons across backbones and generation horizons. The autoregressive inference procedure remains unchanged.

1 INTRODUCTION

Autoregressive (AR) video generation makes each prediction part of the context for future predictions. Few-step distillation makes this streaming generation process increasingly practical (Yin et al., 2025; Huang et al., 2025a; Zhu et al., 2026; Zhao et al., 2026; Feng et al., 2026), but must train a student whose outputs remain useful as generation continues. Self Forcing applies Distribution Matching Distillation (DMD) (Yin et al., 2024b) to causal self-rollouts, exposing the student to its own generated history during training (Huang et al., 2025a). This aligns the sampled trajectories with inference. The supervision applied to those trajectories introduces a further design question: *how should distillation correct a prediction that also provides context for subsequent predictions?*

DMD queries a generated sample by adding noise and comparing a frozen *real score* representing the target distribution with a trainable *fake score* tracking the student distribution. Their difference provides the distillation signal. In an AR rollout, an earlier chunk serves two functions: it is an imperfect prediction to refine and context for subsequent chunks. A single noise level across the rollout couples these functions. Stronger corruption obscures both the content being evaluated and the prefix information supporting later predictions; weaker corruption retains more of the current predictions throughout the rollout. We call this the *context–correction dilemma*: generated history must remain informative as context while remaining subject to correction. This motivates allocating corruption according to the temporal dependencies within the rollout.

We introduce **ViDA**, whose *temporally ordered query* assigns sampled noise levels non-decreasingly along generation order. Lower prefix corruption retains contextual signal, while nonzero noise keeps every chunk subject to correction. The query acts on completed rollouts and leaves the deployed autoregressive sampler unchanged.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

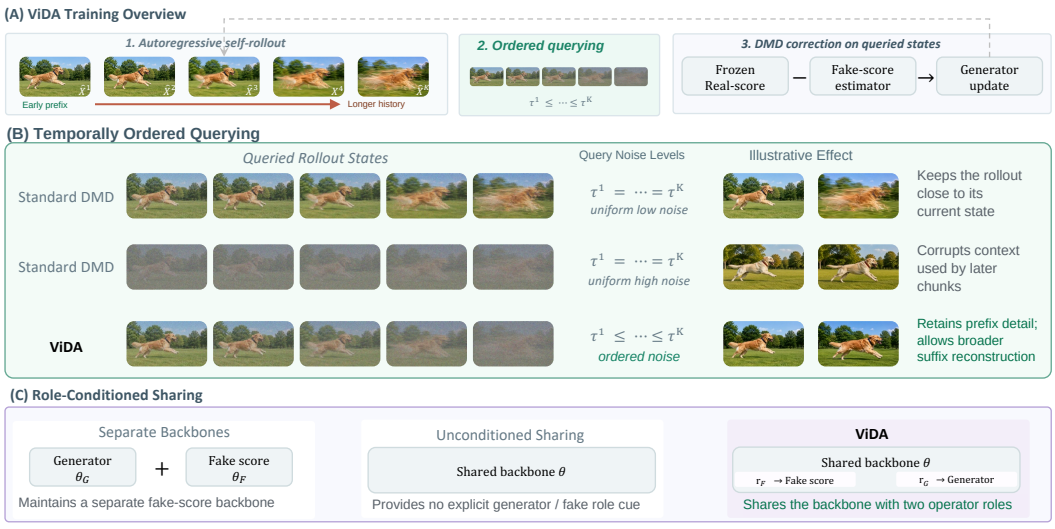


Figure 1: **ViDA adapts distribution matching to autoregressive self-rollouts.** (A) A causal self-rollout is queried by real and fake scores to update the generator. (B) Temporally ordered querying assigns lower but nonzero corruption to earlier chunks and no less corruption to later chunks, preserving relatively informative prefixes while keeping all chunks subject to correction. (C) Role-conditioned parameter sharing realizes the generator and fake-score estimator with one causal backbone and output head, with explicit conditioning to distinguish the two roles.

Controlled experiments test the query without parameter sharing or causal score estimators. With an adapted bidirectional teacher and independent bidirectional fake, ViDA achieves 84.51 VBench Total, versus 83.76 for marginal-matched scalar querying, 83.86 for shuffled assignment, and 83.11 for reversal (Table 1). Each comparison averages three training runs. Dynamic Degree and Motion Smoothness also improve over matched scalar. Query gains persist in shared causal training, with a 14B teacher, and on Cosmos; additional 60-second evaluations assess longer generation.

ViDA also reduces DMD training resources through *role-conditioned sharing*. Generation and fake-score estimation use one causal backbone and output head with explicit role conditioning: *diffusion time specifies the state; role specifies the operator*. At $r = 2$, sharing improves VBench Total from 83.75 to 85.05 over an independent causal fake with the same initialization. Both objectives update the shared backbone. Compared with the stronger independent causal fake at $r = 5$, it improves Total by 0.48 at half the student post-training GPU-hours and 13.6% lower peak allocated memory (Table 2).

The final 1.3B generator distilled from a 14B teacher achieves 85.62 VBench Total: 1.12 above matched scalar within the same recipe and 1.38 above the locally evaluated official Self Forcing checkpoint, with unchanged autoregressive sampling.

Contributions.

- **AR-structured distillation supervision.** We formulate the dual role of generated history in DMD and introduce temporally ordered queries that retain prefix context while keeping the rollout subject to correction. Matched-noise controls isolate the benefit of temporal assignment, with transfer across teacher sizes and backbones.
- **Efficient DMD through role-conditioned sharing.** We replace the separate fake-score network with a shared causal backbone and output head. Matched training comparisons show improved quality and lower memory use, with a better quality–cost tradeoff at fewer fake updates.

2 Related Work

Autoregressive video diffusion and distillation. Modern video diffusion backbones commonly denoise fixed clips with bidirectional temporal attention (Wan Team et al., 2025; HaCohen et al., 2024; Jin et al., 2024). Autoregressive alternatives instead emit frames or chunks sequentially, enabling streaming and reuse of cached history (Deng et al., 2024; Sand.ai et al., 2025; Chen et al., 2025). CausVid converts a pretrained bidirectional model into a causal few-step generator through ODE initialization and asymmetric DMD (Yin et al., 2025). It samples per-chunk diffusion times while constructing diffusion-forced student predictions, but after concatenating the prediction it draws one scalar time for the subsequent holistic DMD query. Self Forcing closes the exposure gap by obtaining the DMD sample from a causal self-rollout, while its completed rollout is likewise re-noised at one sampled time for distribution matching (Huang et al., 2025a). Causal Forcing retains self-rollout DMD but replaces its initialization with a causally aligned ODE; Causal Forcing++ further replaces that costly ODE with causal consistency distillation (Zhu et al., 2026; Zhao et al., 2026). One-Forcing targets a more aggressive one-step regime with an auxiliary adversarial objective (Feng et al., 2026). Diffusion Forcing uses token-specific noise to train sequence generators, and Diagonal Distillation coordinates denoising progress across video chunks (Chen et al., 2024; Liu et al., 2026). Rolling Diffusion, FIFO-Diffusion, and AR-Diffusion organize video generation with increasing or non-decreasing temporal noise levels (Ruhe et al., 2024; Kim et al., 2024; Sun et al., 2025); Rolling Forcing progressively denoises chunks within a rolling generation window (Liu et al., 2025). ViDA instead designs the *post-rollout distribution-matching query*: it assigns corruption levels to an already generated rollout to balance correction of its predictions with the context used to evaluate later chunks. This changes the distillation supervision while retaining the autoregressive sampler.

Long-horizon causal generation. LongLive and memory-based approaches modify cached history (Yang et al., 2025; Wu et al., 2025); BAgger learns recovery trajectories and Self-Forcing++ guides segments of long self-rollouts (Po et al., 2025; Cui et al., 2025). BlockVid combines blockwise generation, sparse caching, and noise scheduling (Zhang et al., 2025). HiAR conditions each block on equally noisy history through hierarchical denoising (Zou et al., 2026). These methods change generator training or rollout construction. ViDA changes the subsequent score query applied to the completed rollout.

Distribution matching and fake-score estimation. DMD estimates a reverse-KL correction from the difference between a frozen real score and a trainable fake score (Yin et al., 2024b). DMD2 stabilizes fake-score tracking and supports multi-step sampling (Yin et al., 2024a). To avoid a separate fake network, FSF-DMD uses a generator-induced pseudo-velocity for flow-map generators (Kim et al., 2026), while TwinFlow trains a self-adversarial flow without a fixed pretrained teacher (Cheng et al., 2025). ViDA retains an external frozen real-score teacher and the fake-denoising objective, and implements generation and fake-score estimation through explicit roles in one causal backbone. Table 2 evaluates this shared design against independent estimators at matched and increased update cadence.

3 Method

ViDA changes the post-rollout query and fake-score parameterization while retaining autoregressive inference. Ordered queries allocate corruption temporally; role conditioning shares the causal backbone between generation and fake-score estimation.

3.1 Post-Rollout Distribution Matching

Let $\hat{\mathbf{x}} = (\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^K)$ be a video generated by the causal student G_θ under text condition $\mathbf{c}$. We use the same few-step sampler and KV-cache procedure as at inference, with earlier generated chunks and their cached representations treated as detached history (Huang et al., 2025a). Distribution matching (Yin et al., 2024b) evaluates the completed rollout after re-noising it. A scalar query applies one noise level to every chunk:

$$\tilde{\mathbf{x}}^k = \alpha_\tau \, \mathrm{sg}[\hat{\mathbf{x}}^k] + \sigma_\tau \boldsymbol{\epsilon}^k, \qquad k = 1, \ldots, K, \tag{1}$$

Here $\boldsymbol{\epsilon}^k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and sg denotes stop-gradient. The difference between real- and fake-score predictions supplies the DMD correction.

A generated chunk is both a prediction and context for later chunks. Re-noising controls how much content the score models can retain versus reconstruct. A shared scalar couples prefix preservation with later-chunk correction; ViDA addresses this dilemma through the distribution of query states.

3.2 Temporally Ordered Querying

Constructing the query. We use generation order as a structural prior for allocating corruption: earlier chunks retain more contextual information through lower but nonzero noise, while later chunks, conditioned on longer self-generated histories, are queried at no lower noise. For each query, we draw a shared raw-time band (L, H), sample K values within it, and assign their order statistics to the chunks:

$$(L, H) \sim p_{\mathrm{band}}, \qquad r_k \overset{\mathrm{iid}}{\sim} \mathcal{U}(L, H), \qquad \tau_k = w\big(r_{(k)}\big), \tag{2}$$

Here $r_{(1)} \leq \cdots \leq r_{(K)}$, and w is the monotone scheduler mapping with non-endpoint clamping. The shared band controls the range of corruption within a query, while sorting gives it a temporal direction. The positive lower clamp keeps every generated chunk open to correction. Absolute noise levels and their contrast vary across queries; the exact sampler is given in Appendix A.4. We denote this query distribution by $\mathcal{Q}_{\mathrm{AR}}$. For $\boldsymbol{\tau} \sim \mathcal{Q}_{\mathrm{AR}}$, the structured state is

$$\tilde{\mathbf{x}}^k = \alpha_{\tau_k} \, \mathrm{sg}[\hat{\mathbf{x}}^k] + \sigma_{\tau_k} \boldsymbol{\epsilon}^k. \tag{3}$$

Preparing the real-score model. The pretrained teacher uses a video-level timestep condition. Before student post-training, we adapt its timestep embedding, projection, and modulation parameters to read one noise level per chunk, keeping the remaining pretrained weights fixed. Both the original and adapted teachers retain bidirectional attention over the video. Per-chunk timestep conditioning specifies the corruption state of each chunk without changing that attention pattern. Under the non-decreasing profile, earlier chunks have assigned noise levels $\tau_j \leq \tau_k$ for every $j < k$, preserving relatively informative prefix evidence within the teacher's full-video query.

For the 1.3B controlled study, adaptation uses the standard flow-matching denoising objective on video–text pairs from OpenVidHD-0.4M (Nan et al., 2025), with a uniform mixture of non-decreasing, shuffled, reversed, and constant query profiles. The adapted teacher is then frozen and reused across the adapted-teacher query comparisons. The final system uses a 14B teacher. Adaptation details are in Appendix A.12.

Matched evaluation and generator update. During a generator update, the real and fake models receive the same corrupted video, timestep vector, and text condition. Table 1 isolates temporal query assignment with an independent bidirectional fake that preserves the full timestep vector. The shared implementation below uses a causal fake, whose visible context differs from the teacher's. Its correction is a first-order distillation surrogate; matching query tensors alone guarantees neither an exact joint-KL gradient nor a vanishing correction at the target distribution. Let $\boldsymbol{\mu}^r_\phi$ and $\boldsymbol{\mu}^f_\psi$ denote the real and fake clean predictions, with ψ the fake-estimator parameters. Following Self Forcing (Huang et al., 2025a), we use the first-order clean-prediction DMD update:

$$\begin{aligned} Z &= \mathrm{mean}\left|\hat{\mathbf{x}} - \boldsymbol{\mu}^r_\phi\right|, & \mathbf{d}^k &= \frac{\boldsymbol{\mu}^{f,k}_\psi - \boldsymbol{\mu}^{r,k}_\phi}{Z}, \\ \mathcal{L}_{\mathrm{ViDA}} &= \frac{1}{2K} \mathbb{E} \sum_{k=1}^{K} \left\| \hat{\mathbf{x}}^k - \mathrm{sg}\big[\hat{\mathbf{x}}^k - \mathbf{d}^k\big] \right\|_2^2 . \end{aligned} \tag{4}$$

The real and fake predictions are evaluated without gradient tracking for this update; gradients pass through the student output in the loss. Fake-estimator updates use a detached rollout and an independently sampled query from the same distribution $\mathcal{Q}_{\mathrm{AR}}$, with the denoising target retained. This trains the fake estimator on the structured query states used by the DMD correction. The chunk-wise prediction-to-score conversion and scheduler preconditioning are given in Appendix A.11.

3.3 Role-Conditioned Sharing of the Causal Backbone

The generator and causal fake estimator use the same architecture, timestep and text interfaces, and velocity parameterization. We share their backbone and output head, replacing the separate training-only fake network with role-specific conditioning.

One backbone, two roles. We write the shared model as

$$F_\theta(\mathbf{z}, \boldsymbol{\tau}, \mathbf{c}, \rho), \qquad \rho \in \{\rho_{\mathrm{g}}, \rho_{\mathrm{f}}\}. \tag{5}$$

The generation role produces the few-step autoregressive rollout; the fake role estimates denoising velocities on its re-noised samples. With overlapping latent and conditioning inputs, the timestep specifies the corruption state and the role specifies the requested operator.

For the fake role, a learned token replaces the final text-conditioning slot and a residual linear map modulates the timestep embedding before time projection. The generation role uses a learned additive offset in the timestep embedding. These signals use existing conditioning pathways and add 2.37M parameters (0.167% of the backbone); all blocks and the output head remain shared. Appendix A.9 gives the exact parameterization.

Training the shared model. The generation role produces the causal self-rollout and is trained with the first-order DMD objective in Eq. 4. For each fake update, the generation role runs again under the current shared weights to produce a fresh rollout $\hat{\mathbf{x}}_{\mathrm{fresh}}$. On iterations that update both roles, this rollout follows the generator optimizer step. We detach it as $\bar{\mathbf{x}} = \mathrm{sg}[\hat{\mathbf{x}}_{\mathrm{fresh}}]$, draw an independent query from $\mathcal{Q}_{\mathrm{AR}}$, and re-noise $\bar{\mathbf{x}}$ as in Eq. 3. The fake role retains the velocity denoising objective

$$\mathcal{L}_{\mathrm{fake}}(\theta) = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left\|v_\theta(\tilde{\mathbf{x}}, \boldsymbol{\tau}, \mathbf{c}, \rho_{\mathrm{f}})^k - (\boldsymbol{\epsilon}^k - \bar{\mathbf{x}}^k)\right\|_2^2\right], \tag{6}$$

where $v_\theta(\cdot, \rho_{\mathrm{f}})$ denotes the fake-role velocity prediction. Two AdamW optimizers maintain separate moment states and update the same backbone under their respective objectives, along with their role-specific conditioning parameters. Updates from either objective change the shared backbone used by both roles. Our sharing comparison evaluates this coupled training design end to end. The rollout target is detached during fake updates, and the score predictions are detached during generator updates, so neither update differentiates through the other role's prediction.

The fake optimizer steps at every outer iteration, and the generator optimizer additionally steps every r iterations. Here r is the number of fake-estimator updates per generator update. Our default Wan configuration uses $r = 2$, giving 600 fake updates and 300 generator updates. Optimizer parameter groups and learning rates are specified in Appendix A.9.

Deployment. Only the generation role is used at inference. The generation-role timestep offset remains active; the fake-role parameters and teacher are unused. The denoising schedule, number of model evaluations, output head, and autoregressive KV-cache procedure remain unchanged.

4 Experiments

We first isolate the post-rollout query with independent bidirectional scores, then evaluate role-conditioned sharing and update cadence. The complete system is evaluated primarily on five-second VBench. Cosmos transfer and matched 60-second evaluation test the scope of the query gains.

Table 1: **Query design with independent bidirectional scores.** A 1.3B student receives 300 generator updates with an independent 1.3B bidirectional fake at $r = 5$. All rows use complete VBench under the protocol in Appendix A.1. The first row uses the original teacher; the remaining rows share the same adapted teacher. The first two rows use one training run each; the remaining rows average three. Per-run statistics appear in Appendix A.6.

Query	Teacher	Total↑	Quality↑	Semantic↑
Reference scalar	Original	83.57	84.56	79.61
Reference scalar	Adapted	83.55	84.68	79.03
Matched scalar	Adapted	83.76	85.00	78.80
Shuffled	Adapted	83.86	85.04	79.17
Reversed	Adapted	83.11	84.10	79.15
ViDA (non-decreasing)	Adapted	**84.51**	**85.72**	**79.66**

4.1 Experimental Setup

Wan controlled studies use a 1.3B causal student, a 1.3B bidirectional real teacher, and 1.3B fake backbones. Every configuration receives 300 generator updates. The ratio r counts fake-objective updates per generator update. Table 1 fixes an independent bidirectional fake at $r = 5$; Table 2 varies fake design and cadence under the non-decreasing query. Teacher adaptation changes timestep conditioning while retaining bidirectional attention. The generator and causal fakes start from the ODE-initialized AR model, before DMD post-training; the final Wan recipe uses this same initialization. The shared five-query study uses the same adapted teacher and causal initialization at $r = 2$. Both objectives update the backbone in shared training. All Wan controlled runs use 64 H100 GPUs and global batch size 64.

The Wan short-video evaluations cover all 946 VBench prompts and 16 dimensions, using Self Forcing's released prompt rewrites for generation and the unchanged official evaluator with the original annotations (Huang et al., 2025a; 2024). Sampling follows the dimension-specific protocol in Appendix A.1. We report checkpoint scores, averaged across independent training runs where available; repetition counts, individual scores, and across-run variability are collected in Appendix A.6. The sampler, adaptation, and evaluation details appear in Appendix A.1.

The final Wan system distills a 1.3B student from a 14B teacher. Main-system comparisons distinguish literature-reported values from local checkpoint evaluations and include a matched query-policy pair within the final recipe. Cosmos and long-video studies use the protocols in their respective appendices.

4.2 Temporal Assignment of DMD Queries

We first evaluate the query independently of parameter sharing. Table 1 fixes an independent bidirectional fake estimator at $r = 5$. Reference scalar queries compare the original and adapted teachers. With the adapted teacher fixed, matched scalar, shuffled, reversed, and non-decreasing queries test noise selection and temporal assignment. Both score estimators remain bidirectional throughout.

Query gains under a fixed adapted teacher. The reference scalar checkpoints score 83.57 and 83.55 with the original and adapted teachers. Holding the adapted teacher fixed, ViDA exceeds matched scalar, shuffled, and reversed queries by 0.75, 0.64, and 1.40 Total, respectively. Shuffling randomly assigns a shared-band noise multiset, while reversal changes its temporal direction. Their comparison with ViDA shows that heterogeneous noise alone does not explain the benefit: temporal assignment matters even with independent bidirectional scores. Across training runs, the Total SD is 0.37 for shuffling and 0.04 for ViDA. All three ViDA Total scores exceed every observed matched-scalar, shuffled, and reversed score (Appendix A.6). In the matched scalar/ViDA comparison, three-run mean normalized Dynamic Degree rises from 81.94 to 86.11 and Motion Smoothness from 90.69 to 92.93 (Appendix A.7).

Unordered controls in shared causal training. A complementary study with the role-shared causal fake at $r = 2$ also ranks ViDA first in each of three runs: 85.05 Total versus 84.32 for shuf-

Table 2: **Fake design: quality and measured training resources.** All rows use the non-decreasing query, the same adapted 1.3B teacher, and 300 generator updates. Cost is measured on 64 H100 GPUs; memory is peak PyTorch allocation on rank 0 after warmup. Quality uses the complete VBench protocol in Appendix A.1. For VBench scores, independent bidirectional $r = 2$ and independent causal $r = 5$ each use one training run; the other rows average three. Run statistics and measurement details are in Appendices A.6 and A.10.

Fake design	r	Total↑	Quality↑	Semantic↑	GPU-h↓	GiB↓
Independent bidirectional	2	83.45	84.58	78.92	118	77.8
Independent bidirectional	5	84.51	85.72	**79.66**	244	77.8
Independent causal	2	83.75	85.10	78.32	111	77.7
Independent causal	5	84.57	85.86	79.42	220	77.7
Role-shared causal	2	**85.05**	**86.41**	79.62	**110**	**67.1**

fling and 83.73 for reversal. Appendix A.5 reports all five query families, prompt-cluster intervals, and individual runs.

4.3 Role-Conditioned Sharing and Training Efficiency

Table 2 fixes the adapted 1.3B bidirectional teacher and non-decreasing query. The independent bidirectional fake starts from Wan; the independent causal fake and shared model use the same AR-adapted initialization as the generator. The two causal designs therefore compare sharing at fixed architecture and initialization, while comparisons with the bidirectional fake also change initialization.

Quality at fixed and increased fake-update cadence. At $r = 2$, role-conditioned sharing improves Total by 1.30 over the independent causal fake, with both Quality and Semantic increasing. More fake updates improve the independent estimators: the bidirectional and causal designs reach 84.51 and 84.57 at $r = 5$, respectively. Sharing reaches 85.05 at $r = 2$. This evaluates the complete shared training design: fake-denoising updates also update the generator's shared backbone. Per-run results are in Appendix A.6.

Distinguishing the roles. Removing all role-specific signals while retaining the shared causal backbone, head, and physical diffusion-time conditioning led to collapsed generation quality or diversity in the evaluated run, despite finite losses. This observation motivates explicit role conditioning; no VBench score is assigned to that run.

Measured quality–cost tradeoff. Against the independent causal fake at $r = 5$, sharing at $r = 2$ improves Total by 0.48 while halving training cost from 220 to 110 H100 GPU-hours. Peak allocated memory falls from 77.7 to 67.1 GiB (13.6%). The cost reduction combines sharing with fewer fake updates: at matched $r = 2$, costs are nearly identical (111 versus 110 GPU-hours), while the memory saving remains. These measurements cover student post-training with a frozen teacher; Appendix A.10 specifies the measurement scope. Sharing adds no model evaluation at inference.

4.4 Comparison with Prior Work

We evaluate the complete ViDA recipe, combining temporally ordered querying and role-conditioned sharing to distill a 1.3B generator from a 14B teacher. Table 3 compares it with bidirectional, autoregressive, and distilled autoregressive video generators.

Table 3: **Five-second VBench comparison.** Self Forcing scores and throughput are evaluated locally from its official checkpoint. Other baseline scores and throughput are reported by Zhu et al. (2026) (Table 1); CausVid denotes the Wan-based implementation. The final block compares two ViDA query policies under the same final training recipe and complete VBench protocol (Appendix A.1; one trained checkpoint per policy). Params is deployed model size and Teacher is the distillation real-score model (NR: not specified for the reported implementation). ViDA throughput is measured on one H100 including VAE decoding.

Method	Params	Teacher	VBench			FPS↑
			Total↑	Quality↑	Semantic↑	
Bidirectional Video Diffusion						
LTX-Video (HaCohen et al., 2024)	1.9B	–	79.83	81.88	71.62	8.98
Wan2.1-1.3B (Wan Team et al., 2025)	1.3B	–	83.37	84.30	79.65	0.78
Autoregressive Video Diffusion						
MAGI-1 (Sand.ai et al., 2025)	4.5B	–	78.88	81.67	67.72	0.19
NOVA (Deng et al., 2024)	0.6B	–	80.31	80.66	78.92	0.88
Pyramid Flow (Jin et al., 2024)	2B	–	80.75	83.41	70.11	6.70
SkyReels-V2 (Chen et al., 2025)	1.3B	–	81.97	83.96	74.01	0.49
Distilled Autoregressive Video Diffusion						
CausVid (Wan) (Yin et al., 2025)	1.3B	NR	81.33	83.98	70.72	17.0
Causal Forcing (Zhu et al., 2026)	1.3B	Wan-14B	84.04	84.59	81.84	17.0
Self Forcing (Huang et al., 2025a)	1.3B	Wan-14B	84.24	85.04	81.07	17.0
Final-recipe query comparison						
ViDA: matched scalar	1.3B	Wan-14B	84.50	85.42	80.80	17.0
ViDA (Ours)	1.3B	Wan-14B	**85.62**	**86.53**	**82.00**	17.0

The final-recipe comparison holds the adapted bidirectional 14B teacher, role-shared causal student/fake, initialization, update cadence, and training budget fixed. Replacing the marginal-matched scalar policy with ViDA's non-decreasing policy raises Total by 1.12, Quality by 1.11, and Semantic by 1.20. This extends the controlled query evidence to the deployed recipe; Appendix A.1 specifies the paired policies.

The complete ViDA system exceeds the locally evaluated official Self Forcing checkpoint by 1.38 Total, 1.49 Quality, and 0.93 Semantic. That comparison measures complete systems; the final two rows isolate the query policy within ViDA. ViDA retains 17.0 FPS measured throughput without additional inference-time model calls.

Figure 2 shows selected state transformations, motion, subject composition, and plant growth.

4.5 Transfer and Temporal Behavior

Generalization to Cosmos. With a common AR Cosmos-Predict2.5-2B initialization, adapted bidirectional 14B teacher, and shared causal fake (NVIDIA et al., 2025), non-decreasing querying improves Total from 78.17 to 80.00 over matched scalar, with gains in both Quality and Semantic (Table 4). Both variants receive 400 generator updates; Appendix A.8 gives the model-specific protocol.

Human preference on native five-second videos. Three non-author raters evaluated the final checkpoints on 128 common prompts, with method identities hidden and presentation order randomized. ViDA receives 191 wins against 125 losses versus matched scalar, and 182 wins against 149 losses versus the official Self Forcing checkpoint (Figure 3). Including ties in the denominator, net preference is $+17.19$ and $+8.59$ percentage points, respectively. These descriptive preferences complement the automated final-recipe comparison. Appendix A.3 gives the protocol and both evaluation dimensions.

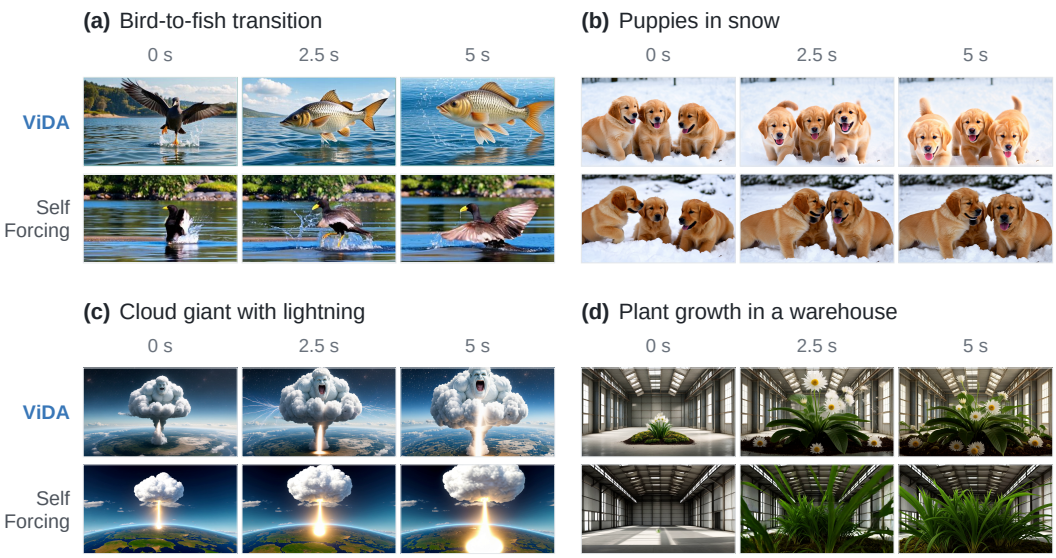


Figure 2: **Selected five-second comparisons with Self Forcing.** Each pair uses the same prompt and shows frames at matched timestamps. ViDA uses the final Table 3 checkpoint; Self Forcing uses its official released checkpoint.

Table 4: **Query transfer to Cosmos.**

Query	Total	Quality	Semantic
Matched scalar	78.17	79.53	72.76
ViDA	**80.00**	**81.20**	**75.21**

Three-run means; common 2B initialization, adapted 14B teacher, shared fake, and 400 generator updates.

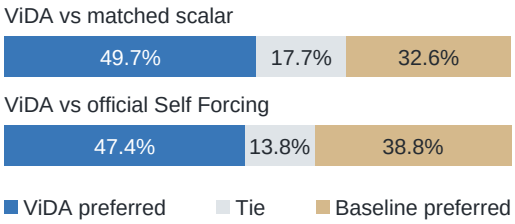


Figure 3: **Overall human preference.** Final Wan checkpoints; 128 prompts and three raters per comparison. Fractions include ties.

Matched evaluation over 60 seconds. We extend the final-recipe scalar and ND checkpoints using the same cache and attention-sink settings. On 128 MovieGen prompts, ViDA obtains higher mean scores on all six supported custom-input VBench-Long dimensions (Huang et al., 2025b) (Table 5). Aesthetic and imaging quality increase by 1.61 and 1.35 points, with positive differences in motion smoothness and dynamic degree. These whole-video measurements complement the five-second results; they do not measure when within a rollout the differences emerge.

Interpretation. The controls establish empirical utility of temporal assignment, without separately identifying position-wise noise marginals, context availability, or score-estimation accuracy. The sharing study likewise measures the coupled training design rather than tracking error alone.

5 Conclusion

ViDA structures DMD queries so earlier predictions retain contextual signal while the entire rollout remains subject to correction. Matched-noise controls, including shuffled and reversed assignments, support temporal assignment with independent and shared score estimators; query gains also transfer across teacher sizes and backbones. Role-conditioned sharing improves quality at fixed update cadence and enables a better quality–cost tradeoff with fewer fake updates. Both contributions operate during training, leaving autoregressive inference unchanged.

AI USE STATEMENT

In this work, we used generative AI tools for editing only, grammar, spelling, and word choice. The authors reviewed and revised the suggestions and take responsibility for the paper's methods, experiments, results, and conclusions.

REPRODUCIBILITY STATEMENT

Section 3 specifies the structured query and role-conditioned training design. Appendix A.1 documents the training and evaluation protocols; the remaining appendix sections provide the query sampler and matched controls, run-wise aggregate results and uncertainty estimates, temporal evaluation, role conditioning and alternating optimization, and cost accounting used in this work. We will publicly release the training and inference code and the distilled model checkpoints to support reproducibility.

REFERENCES

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, 2024.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, et al. SkyReels-V2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025.

Zhenglin Cheng, Peng Sun, Jianguo Li, and Tao Lin. TwinFlow: Realizing one-step generation on large models with self-adversarial flows. *arXiv preprint arXiv:2512.05150*, 2025.

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.

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. *arXiv preprint arXiv:2412.14169*, 2024.

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

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*, 2024.

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*, 2025a.

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*, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b. doi: 10.1109/TPAMI.2025.3633890.

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. *arXiv preprint arXiv:2410.05954*, 2024.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. FIFO-Diffusion: Generating infinite videos from text without training. In *Advances in Neural Information Processing Systems*, 2024.

Youngjoong Kim, Deokyeong Lee, and Jaesik Park. Distribution matching distillation without fake score network. *arXiv preprint arXiv:2605.19256*, 2026.

Jinxiu Liu, Xuanming Liu, Kangfu Mei, Yandong Wen, Ming-Hsuan Yang, and Weiyang Liu. Streaming autoregressive video generation via diagonal distillation. *arXiv preprint arXiv:2603.09488*, 2026.

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.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. OpenVid-1M: A large-scale high-quality dataset for text-to-video generation. In *International Conference on Learning Representations*, 2025.

NVIDIA et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025. URL https://arxiv.org/abs/2511.00062.

Ryan Po, Eric Ryan Chan, Changan Chen, and Gordon Wetzstein. BAgger: Backwards aggregation for mitigating drift in autoregressive video diffusion models. *arXiv preprint arXiv:2512.12080*, 2025.

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.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. *arXiv preprint arXiv:2402.09470*, 2024.

Sand.ai, Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, W. Q. Zhang, Weifeng Luo, et al. MAGI-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.

Mingzhen Sun, Weining Wang, Gen Li, Jiawei Liu, Jiahui Sun, Wanquan Feng, Shanshan Lao, Siyu Zhou, Qian He, and Jing Liu. AR-Diffusion: Asynchronous video generation with auto-regressive diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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

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*, 2024.

Xiaofei Wu, Guozhen Zhang, Zhiyong Xu, Yuan Zhou, Qinglin Lu, and Xuming He. Pack and force your memory: Long-form and consistent video generation. *arXiv preprint arXiv:2510.01784*, 2025.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. LongLive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025.

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

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*, 2024b.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Zeyu Zhang, Shuning Chang, Yuanyu He, Yizeng Han, Jiasheng Tang, Fan Wang, and Bohan Zhuang. BlockVid: Block diffusion for high-quality and consistent minute-long video generation. *arXiv preprint arXiv:2511.22973*, 2025.

Min Zhao, Hongzhou Zhu, Kaiwen Zheng, Zihan Zhou, Bokai Yan, Xinyuan Li, Xiao Yang, Chongxuan Li, and Jun Zhu. Causal forcing++: Scalable few-step autoregressive diffusion distillation for real-time interactive video generation. *arXiv preprint arXiv:2605.15141*, 2026.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. In *International Conference on Machine Learning*, 2026.

Kai Zou, Dian Zheng, Hongbo Liu, Tiankai Hang, Bin Liu, and Nenghai Yu. HiAR: Efficient autoregressive long video generation via hierarchical denoising. *arXiv preprint arXiv:2603.08703*, 2026.

A Additional Method and Evaluation Details

A.1 Experimental Protocol

Wan short-video generation. Each sample contains 81 RGB frames at 832×480 and 16 fps. The corresponding 21 latent frames are divided into seven chunks of three latent frames and generated causally. Every chunk uses four denoising steps at $[1000, 750, 500, 250]$ under the shift-5 scheduler. Training and inference use the same chunking, sampler, and KV-cache update. The student is initialized from the ODE-initialized Self Forcing AR-adapted Wan checkpoint, before DMD post-training, in both the controlled studies and the final recipe. During post-training, the real-score CFG scale is 3.0; the deployed student is sampled without CFG. Student DMD uses the filtered and LLM-expanded VidProM prompts released by Self Forcing (Wang & Yang, 2024; Huang et al., 2025a), without additional filtering.

Controlled query and fake-design studies. Table 1 uses an independent bidirectional fake and $r = 5$: 1,500 fake-objective updates for 300 generator updates. Matched scalar, shuffled, non-decreasing, and reversed queries use the same frozen adapted teacher. The two reference scalar rows differ only in the original versus adapted teacher. Table 2 uses non-decreasing queries throughout. Bidirectional fake backbones initialize from Wan-1.3B; causal independent and shared backbones use the same Self Forcing AR-adapted initialization as the generator. The independent bidirectional fake preserves the full per-frame timestep vector, repeating each chunk's time across its three latent frames, and is trained with all parameters. In each query arm, fake-denoising updates draw independently from that arm's query distribution.

The five-arm shared-query study fixes teacher, initialization, role module, $r = 2$, 600 outer iterations (300 generator updates), optimizer, and evaluation. All Wan controlled runs use 64 H100 GPUs with one sample per GPU, BF16, and gradient checkpointing. Terminal checkpoints are compared at the specified generator-update budget.

Evaluation. We use VBench 0.1.5 in standard mode (Huang et al., 2024). Complete short-video evaluation uses all 946 prompts in the official list and all 16 dimensions. Wan short-video VBench samples are generated from Self Forcing's released `all_dimension_extended.txt`, whose rewrites use Qwen2.5-7B-Instruct (Huang et al., 2025a). The rewritten texts map to the original prompt IDs; scoring retains the original prompt/dimension metadata and unchanged official evaluator. The same generation prompts and scoring pipeline are used for all local Wan VBench comparisons, including the released Self Forcing checkpoint. Each locally evaluated Wan model generates five videos per prompt, except for Temporal Flickering, where 25 candidates per prompt undergo the official static filter. Dimension scores, normalization, and the Quality, Semantic, and Total aggregates follow the unchanged official pipeline. Comparisons within each study use identical prompts and matched rendering indices. Training-run counts and variability appear in Appendix A.6; the Cosmos and longer-video evaluations follow the separately specified protocols below.

Final-recipe query comparison. The final two rows of Table 3 use the same frozen adapted bidirectional Wan-14B teacher, ODE-initialized AR-adapted 1.3B student, role-conditioned causal generator/fake backbone, optimizer settings, update cadence ($r = 2$), 300 generator updates and 600 fake updates, and checkpoint rule. Both use the complete five-second VBench protocol described above, with paired rendering seeds; each row evaluates one independently trained checkpoint.

The non-decreasing policy uses the deployed ViDA sampler. The matched scalar policy samples that same noise vector, selects a chunk index uniformly, and broadcasts its noise level over the rollout. This preserves the pooled one-chunk noise marginal, rather than each temporal position's marginal. The corresponding policy is used for both DMD queries and fake-denoising updates, so the comparison evaluates the complete re-noising policy within the shared recipe. The original and adapted teachers are bidirectional; the shared student/fake is causal in both rows. The locally evaluated Self Forcing row instead uses the downloaded official checkpoint, without retraining, and is a separate system-level comparison.

Inference efficiency. Locally evaluated rows in Table 3 use the same 946-prompt, 16-dimension VBench pipeline. Self Forcing throughput is also measured locally. Other baseline rows retain

their reported evaluation settings. ViDA inference is measured on one H100 GPU. Let T be the full generation wall-clock time in seconds, including VAE decoding, and N the number of output frames. Throughput is N/T; the five-second ViDA configuration has $N = 81$ frames.

A.2 Complementary Metric Protocol

Final-recipe VBench-Long comparison. Table 5 evaluates the two final-recipe checkpoints from Table 3 on 128 MovieGen Video Bench prompts, using the same prompt texts and paired rendering seeds. Each checkpoint generates one 957-frame video per prompt at 16 fps and 832×480. Both use the same attention-sink extension: the first three latent frames remain pinned, with an identical sliding-window cache budget. The attention window contains at most 21 latent frames: three pinned sink frames, up to 15 non-sink history frames, and three current frames. Once the window is full, advancing to a new chunk evicts the oldest non-sink chunk. After each chunk is denoised, its KV entries are recomputed at $t = 0$. We use the official VBench-Long `long_custom_input` pipeline, including its scene splitting, dimension-specific clip processing, slow–fast consistency evaluation, and aggregation (Huang et al., 2025b). We report the six supported custom-input dimension outputs multiplied by 100, without applying the short-video benchmark's normalization again. These are checkpoint-level comparisons over the same prompt set.

Table 5: **Final-recipe query comparison over 60 seconds.** The same 128 MovieGen prompts, one video per prompt and checkpoint, and identical extension settings. Official dimension outputs are multiplied by 100; higher is better. Appendix A.2 gives the protocol.

Query	Subject consistency	Background consistency	Motion smoothness	Dynamic degree	Aesthetic quality	Imaging quality
Matched scalar	96.79	96.22	97.83	53.85	54.64	65.43
ViDA ND	**96.86**	**96.40**	**98.35**	**54.66**	**56.25**	**66.78**

A.3 Human Evaluation of the Final Checkpoints

We compare the final ViDA checkpoint against matched scalar and the official Self Forcing checkpoint evaluated locally in Table 3. Each comparison uses native five-second videos for the first 128 prompts of MovieGen Video Bench (Polyak et al., 2024); the two comparisons reuse the same ViDA videos. Three non-author raters recruited from other teams, none of whom participated in this project, judge every pair. Method identities are hidden, left–right placement and presentation order are randomized, and raters may replay the paired videos and choose a tie.

The primary question is overall preference, considering prompt adherence, visual quality, and motion together. The secondary question concerns temporal coherence, including consistency of appearance and motion over the clip. These dimensions were specified before the results were collected. Each question has 384 votes per comparison (128 prompts times three raters). Table 6 reports the complete counts. Net preference is $100(W - L)/(W + T + L)$, where W, T, L are ViDA wins, ties, and losses.

Table 6: **Human preferences for the final five-second checkpoints.** Each row covers 128 prompts and three raters; the denominator includes ties. Net preference and its 95% prompt-bootstrap confidence interval are in percentage points, conditional on the fixed checkpoints and rater panel.

Opponent	Dimension	Wins	Ties	Losses	Net	95% CI
Matched scalar	Overall	191	68	125	+17.19	[5.73, 28.65]
Matched scalar	Temporal coherence	177	83	124	+13.80	[1.82, 25.78]
Self Forcing	Overall	182	53	149	+8.59	[3.91, 13.28]
Self Forcing	Temporal coherence	175	71	138	+9.64	[2.60, 16.68]

We report descriptive preferences for these fixed checkpoints and this panel. Votes on the two dimensions and comparisons share prompts, raters, and ViDA videos; they are not 1,536 independent observations. The study concerns native five-second generation, not the 60-second extension.

A.4 QUERY SAMPLER

The default sampler is the full non-decreasing profile in Eq. (2); no scalar or bridge mixture is used. Its exact fixed construction is

$$\begin{aligned} u_L, u_H, u_1, \ldots, u_K &\overset{\text{iid}}{\sim} \mathcal{U}(0,1), \\ L &= 600u_L^4, \\ H &= \max\big(1000 - 750u_H^2, L + 100\big), \\ r_k &= \lfloor L + u_k(H - L) \rfloor, \\ \tau_k &= \text{clip}_{[20,980]}\left[1000\frac{5(r_{(k)}/1000)}{1 + 4(r_{(k)}/1000)}\right]. \end{aligned} \tag{7}$$

where $r_{(1)} \leq \cdots \leq r_{(K)}$. During a generator update, the real and fake scores receive the same realized vector. A fake-estimator update draws independently from the same query family. Guidance is fixed at 3.0 in all five matched arms.

Raw timesteps are truncated to integers before the shift. The actual corruption coefficient σ_{τ_k} is obtained from the nearest entry of the native shifted scheduler grid; $\alpha_{\tau_k} = 1 - \sigma_{\tau_k}$. Noising and velocity-to-clean conversion use this same coefficient.

The constants have distinct structural roles. The fourth-power lower-edge draw keeps substantial probability on relatively light prefix corruption, while the squared upper-edge draw keeps substantial probability on high noise levels. The minimum raw-time span of 100 prevents nearly scalar bands. The shift-5 transform is the native Wan scheduler mapping, and clipping avoids degenerate clean and pure-noise endpoints. These choices define the query family; the five-arm experiment changes its cross-chunk structure while matching the pooled one-chunk marginal.

The reference scalar policy with either teacher draws one uniform integer raw time in $[0, 999]$, applies shift 5 and the same effective clamp, and broadcasts it over chunks. Matched scalar instead selects one entry uniformly from a sampled ViDA effective-time vector and broadcasts it. Reversed queries reverse that vector, retaining the multiset, discretization, and clamp. Shuffling randomly permutes the sampled vector, preserving the same within-query multiset distribution.

Matched counterfactual families. The five-arm comparison removes query structure in controlled stages. Given a realized ViDA vector, matched scalar uniformly selects one entry and broadcasts it across all chunks. Its pooled one-chunk marginal and expected corruption budget therefore match ViDA by construction. The independent vector samples one value per chunk independently from this same pooled marginal, removing the shared-band correlation. Shuffling draws the usual shared-band multiset but randomly assigns its entries. Reversal assigns the same realized multiset in reverse generation order. ViDA sorts it in non-decreasing order. The real and fake sides use the same family in every arm.

A.5 SHARED-CAUSAL QUERY CONTROLS

Table 7: **Ordered queries improve AR distillation under matched-noise controls.** All five families use a fixed adapted 1.3B bidirectional teacher, role-shared causal fake, $r = 2$, and 300 generator updates. They share the pooled one-chunk noise marginal. Scores average three independent training runs on complete VBench under the protocol in Appendix A.1. Reversed queries and ViDA additionally share the exact timestep multiset at every update. Δ Total is ViDA minus each alternative; intervals are paired prompt-cluster percentile bootstrap intervals (100,000 resamples). Differences are computed before rounding.

Query family	Total ↑	Quality ↑	Semantic ↑	Δ Total	Paired 95% CI
Matched scalar	84.07	85.39	78.78	+0.99	[+0.24, +1.75]
Independent vector	84.27	85.53	79.21	+0.78	[+0.23, +1.33]
Shuffled	84.32	85.58	79.30	+0.73	[+0.18, +1.29]
Reversed	83.73	84.92	78.95	+1.32	[+0.57, +2.04]
ViDA (non-decreasing)	**85.05**	**86.41**	**79.62**	–	–

Reversed queries and ViDA consume the same random stream and use the same realized timestep multiset step by step, changing only its mapping to causal chunks. Shuffling preserves the within-query multiset distribution but consumes an additional random permutation, so it is a secondary distribution-matched rather than a step-for-step control.

Table 8 reports the three independent training runs. Run indices identify matched training seeds across query families; the indices are local to this experiment. ViDA improves over every alternative in all three runs (12 of 12 matched Total comparisons). In particular, its gains over reversal are 1.31, 1.35, and 1.31 Total. Independent and shuffled queries exchange their relative ranking across runs, while ViDA ranks first in each run.

Table 8: **Query gains persist across independent training runs.** Complete VBench Total for each trained model under the protocol in Appendix A.1. SD is the sample standard deviation across the three training runs, computed from the displayed scores.

Query	Run 1	Run 2	Run 3	SD
Matched scalar	84.34	84.06	83.81	0.27
Independent vector	84.41	84.33	84.06	0.18
Shuffled	84.73	84.13	84.11	0.35
Reversed	83.89	83.81	83.48	0.22
ViDA (non-decreasing)	**85.20**	**85.16**	**84.79**	0.23

Uncertainty. The intervals in Table 7 use paired cluster bootstrap with the prompt as the cluster (100,000 resamples). Each resample draws N prompt IDs with replacement from the N unique prompts, using the same multiplicities across methods, the three evaluated training runs per method, and all dimensions sharing a prompt. Clusters retain their videos and static-filter outcomes; runs and videos are not separately resampled. We recompute each run's dimension scores and Total with the official aggregation and fixed normalization, average Totals across runs, and compute ViDA minus each alternative. The 95% intervals are the 2.5th and 97.5th percentiles of these differences. They quantify prompt-sampling uncertainty conditional on the evaluated models and videos; training-run variability is reported separately in Table 8.

A.6 Training Repetitions and Individual Runs

Configuration names follow Tables 1 and 2. Query rows use an independent bidirectional fake at $r = 5$; fake-design rows use the non-decreasing query.

Tables 1 and 2 report means for matched scalar, shuffled, non-decreasing, and reversed queries, independent causal $r = 2$, and role-shared causal $r = 2$, each with three independent training runs. The two reference scalar rows, independent bidirectional $r = 2$, and independent causal $r = 5$ each report one trained model. All five shared-query arms also have three training runs. Each trained model follows the dimension-specific sampling protocol in Appendix A.1; generation repetitions are distinct from independent training runs.

For the three-run configurations, Total scores (mean $\pm$ sample SD) are matched scalar: 83.76 ± 0.03, shuffled: 83.86 ± 0.37, non-decreasing: 84.51 ± 0.04, reversed: 83.11 ± 0.11, independent causal $r = 2$: 83.75 ± 0.25, and role-shared causal $r = 2$: 85.05 ± 0.23. Differences of means are computed before rounding the means for display.

Run labels are local identifiers for independently trained models. Table 2 reuses the non-decreasing query models from Table 1 as independent bidirectional $r = 5$ and the ViDA models from Table 7 as role-shared causal $r = 2$. Repeated use of a row does not represent a new training run. Table 9 collects the individual runs underlying Tables 1 and 2; Table 8 gives the shared-query Total scores. Its ViDA row is the same model group as role-shared causal $r = 2$ below.

Table 9: **Individual runs underlying the main query and fake-design studies.** Complete VBench under Appendix A.1. Independent and role-shared causal $r = 2$ use the same initialization; the latter reuses the shared-query ViDA models.

Configuration	Run	Total	Quality	Semantic
Reference scalar (original teacher)	Run1	83.57	84.56	79.61
Reference scalar (adapted teacher)	Run1	83.55	84.68	79.03
Matched scalar	Run1	83.79	85.05	78.75
Matched scalar	Run2	83.75	85.02	78.69
Matched scalar	Run3	83.73	84.92	78.97
Shuffled	Run1	84.09	85.33	79.13
Shuffled	Run2	84.06	85.40	78.73
Shuffled	Run3	83.44	84.39	79.65
ViDA (non-decreasing)	Run1	84.54	85.76	79.67
ViDA (non-decreasing)	Run2	84.52	85.73	79.68
ViDA (non-decreasing)	Run3	84.46	85.67	79.64
Reversed	Run1	83.20	84.17	79.33
Reversed	Run2	83.14	84.16	79.05
Reversed	Run3	82.99	83.97	79.07
Independent bidirectional ($r = 2$)	Run1	83.45	84.58	78.92
Independent causal ($r = 5$)	Run1	84.57	85.86	79.42
Independent causal ($r = 2$)	Run1	84.00	85.38	78.46
Independent causal ($r = 2$)	Run2	83.75	85.10	78.34
Independent causal ($r = 2$)	Run3	83.50	84.83	78.16
Role-shared causal ($r = 2$)	Run1	85.20	86.60	79.63
Role-shared causal ($r = 2$)	Run2	85.16	86.55	79.63
Role-shared causal ($r = 2$)	Run3	84.79	86.09	79.60

A.7 Motion in the Independent-Bidirectional Query Study

Table 10 reports normalized VBench dynamic degree and motion smoothness for the matched scalar and ND configurations of Table 1, averaged over the same three trained models. The mean gains are 4.17 and 2.24 points, respectively. Dynamic Degree varies across ND runs (sample SD 8.67), so its mean gain does not imply improvement in every run. These dimensions measure motion activity and smoothness, rather than task completion.

Table 10: **Motion scores for the controlled query comparison.** Normalized scores on a 0–100 scale; means use three training runs. Individual ND runs are included to show training variability.

Query	Run	Dynamic degree	Motion smoothness
Matched scalar	Mean	81.94	90.69
ViDA ND	Run1	88.89	94.02
ViDA ND	Run2	93.06	92.25
ViDA ND	Run3	76.39	92.53
ViDA ND	Mean	**86.11**	**92.93**

A.8 Query Generalization to Cosmos

We compare scalar and temporally ordered queries using the same autoregressive Cosmos-Predict2.5-2B initialization and adapted Cosmos-Predict2.5-14B teacher. The teacher retains bidirectional attention, is adapted with the same equal mixture of non-decreasing, shuffled, reversed, and constant profiles as Wan, and is frozen during distillation. Both variants use a role-shared causal generator/fake initialized from the same AR student model. They receive 400 generator updates using the same set of 6,505 training captions. DMD training uses prompts and generated rollouts. The matched scalar query uniformly selects one entry from a sampled non-decreasing noise vector and broadcasts it to all eight chunks; the ordered query assigns the vector's entries in non-decreasing chunk order. The real and fake score estimators use the corresponding query distribution. Each policy is trained in three runs.

Both variants use AdamW with $\beta_1 = 0$ and $\beta_2 = 0.999$, generator and fake-estimator learning rates of 2×10^{-6} and 4×10^{-7}, respectively, and five fake-estimator updates per generator update. Teacher guidance is set to 3.0. Inference uses chunk-causal attention with KV caching: the three latent frames within a chunk attend to one another, while future chunks are masked. There are eight chunks and four denoising steps per chunk. Each output contains 93 frames at 1280×704 resolution and 16 FPS. This comparison tests query transfer within the shared recipe. Generation uses the same Self Forcing prompt rewrites as the Wan evaluations (Appendix A.1). Both variants use the same rendering and scoring pipeline on the complete VBench suite. Video sampling and static filtering follow the Wan protocol in Appendix A.1.

A.9 Role Parameterization and Alternating Updates

Role conditioning adds 2,366,464 parameters (0.1668% of the 1,418,996,800-parameter backbone). Of these, 2,364,928 are fake-specific: a 4,096-parameter token that replaces the final text-conditioning slot and a 2,360,832-parameter residual linear map applied to the 1,536-dimensional timestep embedding before time projection. The generation role uses a 1,536-parameter additive offset in the same embedding space. The timestep signals reach the transformer blocks through the shared AdaLN conditioning pathway; the fake token uses the existing text-conditioning pathway. No role-specific transformer block or output head is used. The fake-role token, the weights and bias of its residual timestep map, and the generation-role timestep offset are all initialized to zero.

Two AdamW optimizers retain separate moment states over the shared trunk. The generator optimizer updates the trunk at 2×10^{-6} and the generation offset at 10^{-4}; the fake optimizer updates the same trunk at 4×10^{-7}, the fake token at 10^{-4}, and the fake modulation at 10^{-5}. Sharing therefore removes the duplicated fake-network weights and gradient buffer, but does not remove the second optimizer state. During deployment, the fake token and fake modulation are unused, while the generation-role offset remains active.

The fake optimizer steps at every outer iteration, and the generator optimizer also steps once every r iterations. On iterations that update both roles, the generator step precedes the fake step. The fake update uses a fresh, detached rollout generated with the updated shared weights. Consequently, $r = 2$ over 600 outer iterations gives 600 fake updates and 300 generator updates; $r = 5$ over 1,500 outer iterations gives 1,500 fake updates and 300 generator updates.

A.10 Training Efficiency Details

Table 2 measures student post-training with a frozen 1.3B bidirectional teacher on 64 H100 GPUs using DDP, one sample per GPU (global batch 64), and 21 latent frames per sample. Text embeddings are precomputed. H100 GPU-hours are measured elapsed training hours multiplied by 64. All configurations receive 300 generator updates: $r = 2$ entails 600 fake updates and $r = 5$ entails 1,500 fake updates, including fake updates in iterations that also update the generator.

Memory is the rank-0 peak from `torch.cuda.max_memory_allocated()` over multiple steady-state training cycles after warmup, with peak statistics reset before the measurement window. Values are in GiB (2^{30} bytes). This metric measures tensor allocations on the recorded rank, rather than reserved memory or total device occupancy. The shared design retains both AdamW optimizers' moment states, but removes duplicated fake weights and gradient storage (Section A.9). The measured 13.6% reduction compares the two causal designs under the same setup.

A.11 First-Order DMD Proxy

For a fixed timestep vector, the query uses the chunk-factorized Gaussian corruption kernel

$$q_{\boldsymbol{\tau}}(\tilde{\mathbf{x}} \mid \mathbf{x}) = \prod_{k=1}^{K} \mathcal{N}\big(\tilde{\mathbf{x}}^k; \alpha_{\tau_k}\mathbf{x}^k, \sigma_{\tau_k}^2 \mathbf{I}\big)\,. \tag{8}$$

Let $\boldsymbol{\mu}_\phi^r$ be the guided clean prediction of the frozen real-score model and $\boldsymbol{\mu}_\theta^f$ the clean prediction of the fake role. Both are evaluated at the same structured query $(\tilde{\mathbf{x}}, \boldsymbol{\tau}, \mathbf{c})$. We restate Eq. 4 for the shared model: its fake-role prediction is written as $\boldsymbol{\mu}_\theta^f$ rather than $\boldsymbol{\mu}_\psi^f$, and the same generator

objective as $\mathcal{L}_{\text{gen}}(\theta)$:

$$Z = \text{mean}\left|\hat{\mathbf{x}} - \boldsymbol{\mu}_\phi^r\right|, \qquad \mathbf{d}^k = \frac{\boldsymbol{\mu}_\theta^{f,k} - \boldsymbol{\mu}_\phi^{r,k}}{Z},$$
$$\mathcal{L}_{\text{gen}}(\theta) = \frac{1}{2K}\,\mathbb{E}\sum_{k=1}^{K}\left\|\hat{\mathbf{x}}^k - \text{sg}\left[\hat{\mathbf{x}}^k - \mathbf{d}^k\right]\right\|_2^2. \tag{9}$$

Z is computed separately for each video by averaging over all latent frames, channels, and spatial positions; all chunks of that video share this value. Native velocity predictions are converted to clean predictions before computing $\mathbf{d}^k$, using the coefficients associated with each τ_k. For the Gaussian corruption in Eq. (8), define the score-form outputs $\mathbf{s}^{a,k} = (\alpha_{\tau_k}\boldsymbol{\mu}^{a,k} - \tilde{\mathbf{x}}^k)/\sigma_{\tau_k}^2$ for $a \in \{r, f\}$. Their difference obeys

$$\mathbf{s}^{f,k} - \mathbf{s}^{r,k} = \frac{\alpha_{\tau_k}}{\sigma_{\tau_k}^2}\left(\boldsymbol{\mu}_\theta^{f,k} - \boldsymbol{\mu}_\phi^{r,k}\right). \tag{10}$$

Thus Eq. (9) applies the positive, chunk-wise preconditioner $\sigma_{\tau_k}^2/\alpha_{\tau_k}$ to the score-form difference before the shared robust normalization Z; the scheduler factor is retained separately for every chunk. Score evaluations are performed without gradient tracking during the generator update. During the fake-role update, the self-rollout is detached and gradients pass only through the prediction in Eq. (6). The two roles share parameters across alternating updates, but neither update introduces a second-order gradient path.

A.12 Real-Score Interface and Training Scope

The pretrained real-score model expects one timestep per video. For ViDA, we adapt only its timestep embedding, projection, and modulation tensors to read one timestep per three-latent-frame chunk.

Teacher for the 1.3B controlled study. The 17.20M trainable parameters constitute 1.21% of the 1.419B model; all 788 visual-backbone tensors remain frozen. The adapter is trained for 2,000 steps with the standard flow-matching denoising objective on video–text pairs from OpenVidHD-0.4M (Nan et al., 2025). The endpoint checkpoint is fixed rather than selected by downstream student quality.

The adaptation query distribution uniformly mixes non-decreasing, shuffled, reversed, and constant profiles. All five matched arms use the same frozen adapted checkpoint with one timestep per three-latent-frame chunk. Both original and adapted real-score models retain bidirectional attention across the full video; adapting time conditioning does not change visibility. No arm identity or arm-specific parameters are provided to the adapter; each arm supplies only its timestep vector through the shared interface. The mixture exposes the teacher to all four profiles; it does not impose equal denoising accuracy across profiles.

Teacher for the final system. The Wan-14B teacher likewise adapts its timestep embedding, projection, and modulation parameters, totaling 186.09M parameters (1.30% of the model), while its visual-backbone weights remain fixed. We use the checkpoint after 2,000 adaptation steps, using the same OpenVidHD-0.4M data, flow-matching objective, and equal four-profile mixture as the 1.3B teacher. We freeze it during student post-training. Score evaluation retains the pretrained bidirectional attention.

Real-score guidance is used only to form the training-time correction; neither the real-score model nor classifier-free guidance is used by the deployed student.