LOCAL FORCING: LOCAL KV-CACHE CONSTRUCTION FOR LONG VIDEO GENERATION

Anonymous authors
Paper under double-blind review



Figure 1: **Local Forcing enables minute-level video rollouts.** **(a)** Two Local Forcing examples spanning 240 seconds, without specialized sink strategies, additional memory modules, or long-horizon DMD. **(b)** Full seven-dimension VBench Long Quality Score at 30/60 seconds; labels show gains in percentage points. Full results: Table 2.

ABSTRACT

One of the central challenges in causal video generation is error accumulation. We find that attending to history during key–value (KV) cache construction may exacerbate error accumulation, even though the same context is essential for denoising. This observation motivates **Local Forcing** (LF), which follows a simple principle: *generate with context; encode locally*. This principle is implemented through an attention-mask modification applied consistently across teacher forcing, consistency distillation, self-rollout distillation, and inference. Surprisingly, LF improves robustness to drift, enabling stable minute-level rollouts without specialized attention-sink strategies, additional memory modules, or long-horizon distribution matching distillation. The simplicity of LF facilitates integration with both training-aware and training-free approaches to long-video generation, yielding consistent improvements in overall video quality across the evaluated methods. These findings position local cache construction as a foundational design principle for scaling causal video generation far beyond its training horizon.

1 INTRODUCTION

Autoregressive video diffusion provides a practical foundation for interactive world models, where generation must respond to new actions as a scene unfolds (Team et al., 2026; Gao et al., 2026; Jiang

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 2: **Less appearance drift despite a training mismatch.** Even a checkpoint trained with full-context encoding better preserves colors and textures at 30 seconds when we restrict cache construction to the completed block. Only the encoding mask changes: weights remain frozen, and denoising retains historical attention. This improvement suggests that local encoding is a promising basis for training more stable long-video generators. The inference-only switch also increases temporal jitter during playback, motivating consistent local encoding during training and inference.

et al., 2026). Generating successive blocks from available history allows these models to extend a scene as new inputs arrive (Yin et al., 2025; Huang et al., 2025; Zhu et al., 2026). Each generated block becomes context for the next, helping the model keep the scene consistent. Yet errors in this context can also spread to later blocks and accumulate into visible drift over long rollouts (Wang et al., 2026; Li et al., 2026c).

To understand how these errors can reach later predictions, we examine how generated blocks become historical context. In common autoregressive diffusion frameworks, this context takes the form of attention keys and values (K/V) stored in a cache (Huang et al., 2025; Zhu et al., 2026; Zhuang et al., 2026). The transformer first denoises a new latent block using this cache, then re-encodes the completed block at timestep $t = 0$ to construct its cache entries. Both passes commonly use the same block-causal attention mask, so encoding reads history again and incorporates historical features into the hidden states used to construct new K/V. Cache construction thus introduces an additional route by which historical errors may affect future predictions, beyond their influence on the generated block itself.

We therefore ask: *does a completed block need to attend to history again to become useful context for the future?* Denoising uses context to infer content that is not yet known; encoding starts from a completed block whose content is already available. We probe this question in a frozen model by restricting KV-cache encoding attention to the completed block while retaining historical attention during denoising. Even though the checkpoint was trained with full-context encoding, this single mask change better preserves the scene's colors and textures at 30 seconds in the pilot (Figure 2). This improvement suggests that local cache construction provides a promising basis for training more stable long-video generators. However, switching the encoding rule only at inference also increases temporal jitter due to the discrepancy between training and inference.

We introduce **Local Forcing** (LF), which applies one rule throughout training and inference: *generate with context; encode locally*. LF retains historical context during denoising and restricts video self-attention during KV-cache construction to the current block of jointly denoised latent frames. Under full-context encoding, errors in historical features can enter each new block's K/V and be passed on again through subsequent cache updates. LF cuts this recursive encoding path, which can reduce error accumulation while retaining the influence of history through the generated block. Training with this rule lets the same transformer learn both to construct local K/V and to use them for subsequent generation, which can mitigate inference-only jitter without an extra loss or smoothing module. The same mask applies across teacher forcing (TF), consistency distillation (CD), distribution-matching distillation (DMD), and inference. LF preserves the existing objectives and KV-cache interface, adding no parameters or forward passes.

To evaluate whether local encoding improves long-video generation, we combine controlled studies of encoding context with long-rollout evaluations across training frameworks and inference methods. We begin with matched teacher-forcing experiments that vary the encoding context while holding historical access during denoising fixed. Local encoding improves video quality across three training seeds before any rollout distillation, demonstrating that the encoding rule itself can improve generation under otherwise matched conditions.

We then test whether this benefit carries over to few-step generation as rollouts extend beyond the distillation horizon. LF improves Quality Score at 30 and 60 seconds across all three evaluated forcing frameworks, with gains extending to minute-level generation. With short-horizon DMD and no additional memory or compression modules, Causal Forcing with LF improves all seven evaluated dimensions over minute-level rollouts, including dynamics and subject and background consistency. To assess whether this benefit complements specialized history management, we combine one shared LF-trained checkpoint with three long-video inference methods. Each combination improves Quality Score over the corresponding method's baseline without method-specific retraining. Together, these results support improving how history is encoded as a foundation for both the base generator and the methods that retain and reuse its context.

2 Related Work

2.1 From Video Diffusion to Autoregressive Generation

Video diffusion models typically denoise all frames in a clip jointly, using bidirectional attention to model temporal dependencies (Ho et al., 2022; Blattmann et al., 2023; Yang et al., 2025; Wan et al., 2025). This full-clip computation is less suited to streaming applications, which require new frames to be generated without access to future content. Autoregressive models offer a different formulation, generating visual tokens sequentially conditioned on the preceding sequence (Weissenborn et al., 2020; Yan et al., 2021; Kondratyuk et al., 2024). Autoregressive video diffusion brings these approaches together: it generates frames or blocks sequentially, with a diffusion model representing each conditional distribution (Weng et al., 2024; Gao et al., 2024; 2025; Sand.ai et al., 2025). Causal attention further allows historical keys and values to be cached and reused across generation steps (Yin et al., 2025; Huang et al., 2025). This supports efficient streaming, but errors in generated history can propagate to subsequent predictions.

2.2 Forcing and Learning from Generated Histories

Exposure bias in sequential prediction motivates training beyond ground-truth prefixes (Ross et al., 2011; Bengio et al., 2015). Diffusion Forcing unifies next-token and full-sequence prediction through independently noised sequence elements (Chen et al., 2024). Distillation reduces diffusion sampling cost (Salimans & Ho, 2022; Song et al., 2023; Yin et al., 2024a;b), enabling few-step causal generation in CausVid (Yin et al., 2025). Self Forcing trains on self-generated rollouts (Huang et al., 2025); subsequent work improves causal initialization (Zhu et al., 2026; Zhao et al., 2026) and longer-horizon supervision (Cui et al., 2026b; Guo et al., 2025; Chen et al., 2026). Future-aware representation targets and multi-chunk prediction further connect learning to subsequent generation (Yu et al., 2026; Xu et al., 2026). Local Forcing applies across these training procedures by restricting the context used to encode completed blocks, while retaining their objectives.

2.3 Long-Video Generation and Context Design

Generating long videos often involves coordinating shorter temporal segments (Harvey et al., 2022; Yin et al., 2023; Wang et al., 2023). Moving diffusion windows and noise rescheduling extend generation within a limited processing window (Ruhe et al., 2024; Kim et al., 2024; Qiu et al., 2024; Xie et al., 2025), while positional extrapolation supports temporal positions beyond training lengths (Zhao et al., 2025; Yesiltepe et al., 2026; Cui et al., 2026a). To carry information across these segments, memory mechanisms and compact context representations retain history under limited storage (Henschel et al., 2025; Zhang et al., 2025; Kim et al., 2026; Mao et al., 2026). Beyond retaining history, context policies determine how it is used: Rolling Forcing (Liu et al., 2026) excludes sink images but retains recent history during cache updates, while history guidance and context noise regulate temporal conditioning (Song et al., 2025; Yang et al., 2026). Sustaining quality over long rollouts further motivates recent approaches to mitigating drift (Guo et al., 2026; Li et al., 2026c; Yuan et al., 2026; Po et al., 2026), alongside inference strategies based on adaptive references and spectral interventions (Yi et al., 2026; Li et al., 2026a; Dalva & Yanardag, 2026; Li et al., 2026b). SGF (Zhuang et al., 2026) and LF address this challenge through complementary redesigns of KV-cache construction. SGF restores future-loss gradients through cache construction during

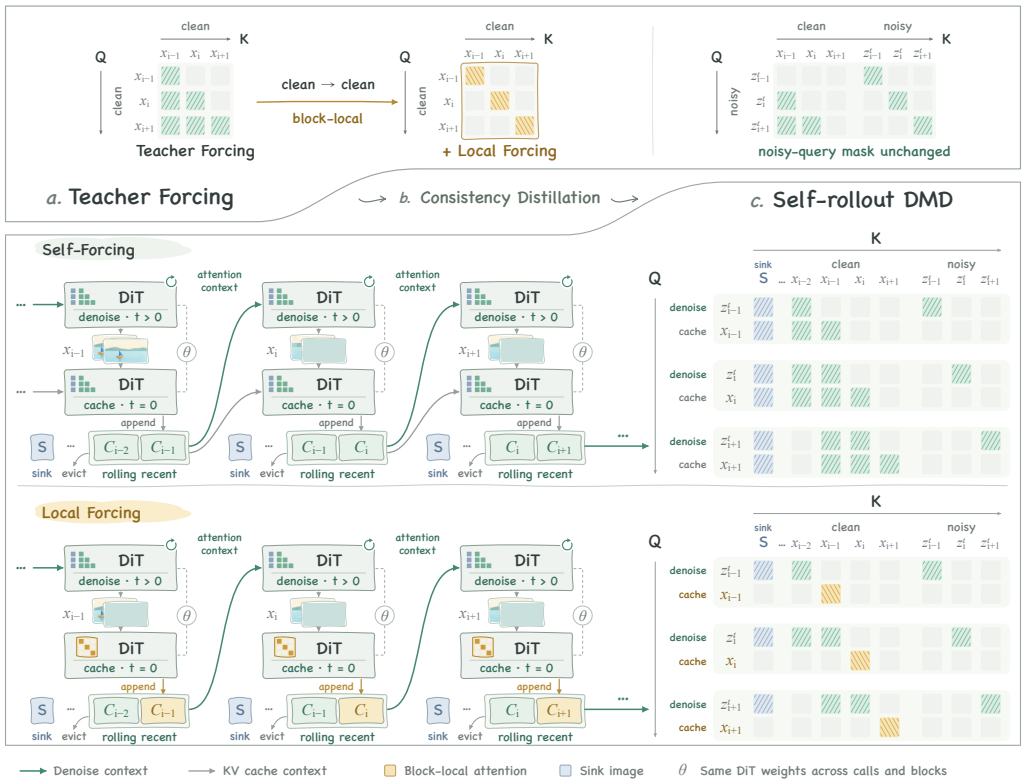


Figure 3: **Local Forcing in the training pipeline.** The same block-local encoding mask applies across teacher forcing, consistency distillation, and self-rollout DMD; denoising retains historical access and each stage retains its objective. Both passes share the same DiT.

DMD, whereas LF enforces block-local attention for cache construction across TF, CD, DMD, and inference.

3 METHOD

Local Forcing follows the principle *generate with context; encode locally*: denoising uses historical context, while KV-cache construction is restricted to the completed block (Figure 3).

3.1 DENOISING AND KV-CACHE CONSTRUCTION

We consider autoregressive video generation in blocks of jointly denoised latent frames. Let x_i denote the clean latent tokens of block i and z_i^t their noisy state at diffusion timestep t, with $t = 0$ the clean endpoint. For a transformer with L layers, let $K_i^{(\ell)}$ and $V_i^{(\ell)}$ denote the keys and values computed from the completed block at layer ℓ. Its KV cache is the collection $C_i = \{(K_i^{(\ell)}, V_i^{(\ell)})\}_{\ell=1}^{L}$, while $C_{<i}$ denotes the retained K/V from preceding blocks. This retained context may include the full history or selected sink and recent tokens.

Cached autoregressive generation alternates between denoising and cache construction (Huang et al., 2025). The diffusion transformer first denoises z_i^t over successive sampling steps, attending to both the current block and $C_{<i}$. Once x_i is obtained, the same transformer processes it at $t = 0$ in a clean-context forward pass. At each self-attention layer, the keys and values computed from the current block's hidden states are stored in C_i. These cached features subsequently provide historical context for generating later blocks.

3.2 Block-Local Attention for Cache Construction

Historical attention serves two roles in cached generation: it conditions denoising and influences the representations stored for future blocks. During cache construction, historical features enter the hidden states from which subsequent layers compute K/V. LF separates these roles by preserving historical context for denoising and restricting KV-cache construction to the completed block.

To express this distinction, consider a single self-attention layer and head. Let Q_i, K_i, and V_i denote the current block's query, key, and value matrices, and $K_{<i}$ and $V_{<i}$ the historical K/V in $C_{<i}$. The current-block matrices are computed afresh from hidden states in each pass, using z_i^t for denoising and x_i for cache construction. Let d denote the head dimension, $[\,;\,]$ concatenation along the key-token dimension, and M an additive mask with entries 0 for permitted query–key pairs and $-\infty$ otherwise. Both passes use scaled dot-product attention (Vaswani et al., 2017):

$$\mathrm{Attn}_i(M) = \mathrm{softmax}\left(\frac{Q_i[K_{<i}; K_i]^\top}{\sqrt{d}} + M\right)[V_{<i}; V_i]. \tag{1}$$

LF assigns separate masks to denoising (M_{denoise}) and cache construction (M_{cache}), excluding historical K/V only during the clean-context pass:

$$M_{\mathrm{denoise}} = [\underbrace{0}_{\text{history}} \;;\; \underbrace{0}_{\text{current block}}]; \qquad M_{\mathrm{cache}} = [\underbrace{-\infty}_{\text{history}} \;;\; \underbrace{0}_{\text{current block}}]. \tag{2}$$

Each entry denotes a constant submatrix over the indicated key group. Under M_{cache}, each token still attends to all K/V within its jointly denoised block, preserving within-block spatiotemporal interactions. The resulting K/V form C_i and serve as context for later blocks.

LF therefore preserves historical conditioning during generation while constructing each block's KV cache locally. The restriction is applied at every video self-attention layer: C_i is local to the supplied x_i, which itself was generated with history. This attention-mask change adds no parameters, losses, or forward passes, and leaves text cross-attention and positional transformations unchanged. Supplementary Section S10 provides the formal locality analysis and mask implementation details.

3.3 Integration Across Training and Inference

The distinction between generating a block and encoding completed content applies throughout training and inference. LF therefore uses the same local cache-construction rule in parallel training sequences and sequential rollouts, while preserving contextual denoising.

Teacher forcing. Ground-truth clean blocks and their noisy counterparts are processed together in one sequence. LF restricts each clean block to attend only to itself, making the clean-to-clean attention submatrix block diagonal. Each noisy block retains attention to preceding clean blocks and its own noisy tokens; its corresponding clean target, future clean blocks, and other noisy blocks remain excluded. The denoising objective and parallel training layout are preserved.

Consistency distillation. The causal student, its exponential-moving-average (EMA) target, and the causal teacher all use M_{cache} for clean-context K/V construction. The consistency objective and timestep pairing remain unchanged (Song et al., 2023; Zhu et al., 2026). LF can also be introduced from an existing distilled checkpoint, as in our framework comparisons (Supplementary Section S1).

Self-rollout distillation. During distribution-matching distillation (DMD), the generator sequentially denoises each block using retained history and constructs its K/V locally before generating the next block (Huang et al., 2025). The distribution-matching objective and score networks are unchanged. Local cache construction remains differentiable with respect to model parameters, preserving the gradient path from future-block losses used by SGF (Zhuang et al., 2026).

Inference and composition. Inference combines local cache construction with the existing context-selection policy for denoising. The unchanged K/V interface allows an LF-trained checkpoint to incorporate context-retention, compression, and positional-encoding methods (Yi et al., 2026; Yesiltepe et al., 2026; Kim et al., 2026) without method-specific retraining.

Table 1: **Teacher forcing on RF200.** Results over three training seeds (Supplementary Section S5). Shading marks the better score within each pair.

Method	Aesthetic Quality	Background Consistency	Imaging Quality	Motion Smoothness	Subject Consistency	Temporal Flickering	Dynamic Degree	Quality Score	Δ
10 seconds									
Teacher Forcing	52.18	91.66	61.87	98.21	87.72	**97.32**	57.17	79.62	—
+ Local Forcing	**53.05**	**92.32**	**64.24**	**98.25**	**89.11**	97.24	**59.17**	**80.59**	+0.97
20 seconds									
Teacher Forcing	48.02	88.70	56.51	97.62	80.89	**96.67**	61.67	76.81	—
+ Local Forcing	**50.21**	**90.13**	**61.17**	**97.78**	**84.00**	**96.67**	**63.33**	**78.71**	+1.90

4 Experiments

4.1 Experimental Setup

Models and training. For the main teacher-forcing (TF) experiments, we initialize the baseline and LF models from Wan2.1-T2V-1.3B (Wan et al., 2025) and train for 10,000 steps on Open-Sora-Plan (Lin et al., 2024), averaging results over three training seeds. For the main DMD experiments, we initialize from the consistency-distilled checkpoint of Causal Forcing (CF) (Zhu et al., 2026). We train CF and Self Gradient Forcing (SGF) (Zhuang et al., 2026), each with and without LF, for 500 DMD steps. All training runs use 8 GPUs. In the main experiments, the generation chunk size is fixed at three jointly denoised latent frames, with outputs at 832×480 resolution and 16 fps. Unless otherwise specified, LF denotes the block-local setting with a KV-cache attention chunk size of three latent frames. We apply this setting during training and inference, preserving each framework's denoising context, objective, sampler, and cache-retention policy. Protocols and additional Self Forcing training results with ODE initialization are provided in Supplementary Sections S1 and S3.

Evaluation. Following Rolling Forcing (Liu et al., 2026), we evaluate TF checkpoints at 10 and 20 seconds on its 200-prompt subset (RF200) sampled from MovieGen (Polyak et al., 2024). We follow the official VBench Long protocol (Huang et al., 2024; 2026) for CF and SGF at 30 and 60 seconds and use 128 randomly sampled MovieGen prompts (MovieGen-128) at 240 seconds.

4.2 Main Results

Teacher-forcing results. Local KV-cache construction improves generation in the teacher-forcing regime (Table 1). LF raises the three-seed mean Quality Score by 0.97 points at 10 seconds and 1.90 points at 20 seconds, with gains in each individual seed at both horizons. At the longer horizon, these gains accompany improvements of 4.66 points in Imaging Quality and 3.11 points in Subject Consistency. LF also consistently improves Quality Score at both horizons with frame-wise autoregressive generation and on Wan2.2-TI2V-5B (Supplementary Sections S7 and S8).

DMD results. Using the released Causal Forcing (CF) consistency-distilled checkpoint, we train CF and Self Gradient Forcing (SGF) with and without LF. LF improves Quality Score for both frameworks at 30 and 60 seconds (Table 2, upper groups). At 60 seconds, CF and SGF gain 1.53 and 1.31 Quality Score points, respectively. Dynamic Degree and Subject Consistency also improve together. Despite these gains, Imaging Quality decreases slightly. We attribute this in part to limited adaptation to local KV-cache construction within the short DMD training budget, while extending DMD training risks motion collapse. Applying LF from TF onward avoids this decrease in our matched full TF–CD–DMD experiment, where both Imaging Quality and overall Quality Score improve at both horizons (Supplementary Section S4). We report it in the supplement because our TF–CD checkpoints are less stable on long rollouts than CF's released CD checkpoint, likely reflecting subtle differences in TF/CD data and training recipes.

Compatibility with training-free methods. Because LF changes only the attention mask for KV-cache construction, it integrates directly with training-free long-video inference methods. We apply

Table 2: **VBench Long at 30 and 60 seconds.** Training-free inference methods share the CF baseline/LF-trained checkpoint pair. Shading marks the better result within each pair.

Method	Aesthetic Quality	Background Consistency	Imaging Quality	Motion Smoothness	Subject Consistency	Temporal Flickering	Dynamic Degree	Quality Score	Δ
30 seconds									
Training frameworks									
Causal Forcing (Zhu et al., 2026)	59.98	96.42	**67.74**	97.87	97.49	99.01	56.57	82.94	—
+ Local Forcing	**61.21**	**96.59**	67.55	**98.47**	**98.06**	**99.14**	**63.15**	**84.11**	+1.18
Self Gradient Forcing (Zhuang et al., 2026)	58.66	**95.21**	**69.95**	**98.23**	97.04	99.14	77.31	84.59	—
+ Local Forcing	**59.26**	95.05	69.01	97.70	**97.26**	**99.41**	**89.35**	**85.29**	+0.71
Training-free inference methods (shared CF checkpoints)									
Deep Forcing (Yi et al., 2026)	59.92	**95.32**	**67.95**	97.97	96.79	**99.33**	71.57	83.94	—
+ Local Forcing	**60.95**	95.19	67.34	**98.27**	**97.29**	99.03	**75.56**	**84.42**	+0.48
Infinity-RoPE (Yesiltepe et al., 2026)	60.69	**94.73**	**68.37**	97.44	96.38	98.99	**82.87**	84.38	—
+ Local Forcing	**61.18**	94.31	67.48	**98.03**	**96.73**	**99.09**	81.57	**84.55**	+0.17
MemRoPE (Kim et al., 2026)	59.16	94.88	**68.01**	97.87	96.03	**99.33**	**72.59**	83.64	—
+ Local Forcing	**60.46**	**95.03**	67.85	**98.22**	**97.18**	98.72	70.83	**83.85**	+0.21
60 seconds									
Training frameworks									
Causal Forcing (Zhu et al., 2026)	59.80	96.38	**67.57**	97.47	96.91	98.96	56.67	82.55	—
+ Local Forcing	**60.84**	**96.54**	67.42	**98.44**	**97.87**	**99.11**	**64.63**	**84.08**	+1.53
Self Gradient Forcing (Zhuang et al., 2026)	56.40	**94.98**	**69.09**	**98.20**	96.56	99.03	77.13	83.89	—
+ Local Forcing	**58.70**	94.95	68.89	97.65	**96.98**	**99.39**	**90.88**	**85.20**	+1.31
Training-free inference methods (shared CF checkpoints)									
Deep Forcing (Yi et al., 2026)	58.56	95.08	66.87	98.01	96.25	**99.22**	69.81	83.26	—
+ Local Forcing	**60.86**	**95.11**	**67.24**	**98.20**	**97.17**	98.99	**79.07**	**84.56**	+1.30
Infinity-RoPE (Yesiltepe et al., 2026)	60.29	**94.58**	**68.36**	97.31	95.98	98.78	**83.94**	84.14	—
+ Local Forcing	**60.86**	94.27	67.37	**98.01**	**96.67**	**99.01**	82.50	**84.49**	+0.35
MemRoPE (Kim et al., 2026)	57.55	94.52	66.84	97.74	95.28	**99.29**	**73.89**	83.02	—
+ Local Forcing	**59.87**	**94.89**	**67.86**	**98.19**	**96.93**	98.65	71.44	**83.68**	+0.67

Table 3: **Minute-level generation on MovieGen-128.** All methods use the same 128 randomly sampled prompts; training-free inference methods share the CF baseline/LF-trained checkpoint pair. Shading marks the better result within each pair.

Method	Aesthetic Quality	Background Consistency	Imaging Quality	Motion Smoothness	Subject Consistency	Temporal Flickering	Dynamic Degree	Quality Score	Δ
Training frameworks									
Causal Forcing (Zhu et al., 2026)	53.49	95.54	63.58	97.22	95.91	97.54	65.05	80.53	—
+ Local Forcing	**56.82**	**96.04**	**65.38**	**98.03**	**96.83**	**98.55**	**75.01**	**83.20**	+2.67
Self Gradient Forcing (Zhuang et al., 2026)	51.50	95.00	68.15	**97.80**	95.41	**97.87**	83.44	82.59	—
+ Local Forcing	**55.87**	**95.28**	**68.87**	96.97	**95.59**	96.86	**95.40**	**83.52**	+0.93
Training-free inference methods (shared CF checkpoints)									
Deep Forcing (Yi et al., 2026)	55.05	95.14	**66.94**	97.59	95.69	98.13	76.44	82.48	—
+ Local Forcing	**56.69**	**95.68**	66.27	**97.89**	**96.42**	**98.33**	**86.50**	**83.89**	+1.41
Infinity-RoPE (Yesiltepe et al., 2026)	56.80	95.12	**68.65**	97.00	95.55	97.97	87.70	83.47	—
+ Local Forcing	**57.59**	**95.17**	66.93	**97.56**	**95.67**	**98.12**	**91.60**	**84.02**	+0.55
MemRoPE (Kim et al., 2026)	54.47	95.08	**68.76**	97.63	95.34	97.96	81.95	82.97	—
+ Local Forcing	**56.46**	**95.63**	67.46	**97.72**	**96.40**	**98.04**	**83.42**	**83.58**	+0.61

Deep Forcing (Yi et al., 2026), Infinity-RoPE (Yesiltepe et al., 2026), and MemRoPE (Kim et al., 2026) to both the CF and CF+LF checkpoints, preserving each method's cache-management and positional-encoding settings without method-specific retraining. LF improves Quality Score for all three methods at both 30 and 60 seconds (Table 2, lower groups), with gains of 0.35–1.30 points at 60 seconds. Aesthetic Quality, Subject Consistency, and Motion Smoothness also improve across all three methods at both horizons; Supplementary Section S12 details the inference setup.

Minute-level generation. The benefits extend to minute-level MovieGen-128 rollouts (Table 3). LF improves all seven dimensions for CF, with gains of 2.67 points in Quality Score and 9.96 points in Dynamic Degree. SGF gains 0.93 Quality Score points and 11.95 Dynamic Degree points, with higher subject and background consistency. The three training-free methods also benefit: Quality Score increases by 0.55–1.41 points, and each improves six of seven dimensions, including Dynamic Degree and Subject Consistency, with a decrease in Imaging Quality.

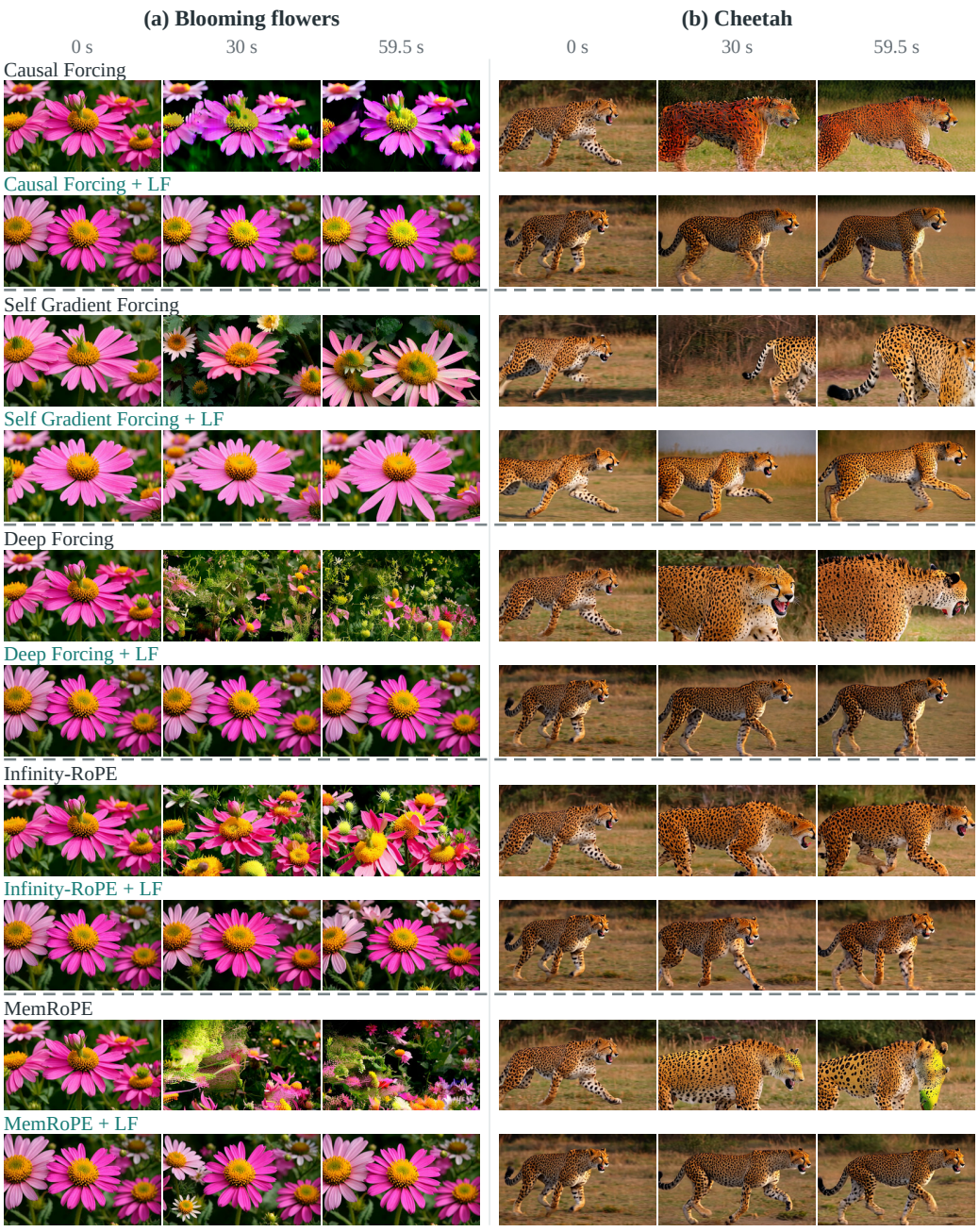


Figure 4: **Qualitative results on MovieGen-128.** LF better preserves flower structure and cheetah appearance across the five methods.

Across all five methods and three horizons, Dynamic Degree and Subject Consistency increase together in 11 of 15 comparisons. Figure 4 illustrates more stable appearance across the evaluated methods, with additional qualitative comparisons in Supplementary Section S14. Supplementary Section S2 reports blinded, model-based Astra preferences from sampled frames of 30-second prefixes and minute-level rollouts. Although CF primarily targets real-time interactive generation, LF unlocks its potential for long-video generation through a single change to the KV-cache attention mask, without long-horizon DMD or additional memory modules.

4.3 Ablation Studies

How local should cache construction be? We progressively restrict cache-construction attention to distinguish the effect of sink exclusion from the benefit of removing recent history. **Full context** attends to retained history and the current block. **Sink-isolated** excludes the seven sink frames while retaining recent context, following the cache-context choice in Rolling Forcing (Liu et al., 2026). **Block-local** restricts attention to the current three-frame latent block, and **Frame-local** further restricts it to a single latent frame. We train each variant with three seeds, matching initialization, data order, and training budget within each seed. Denoising context and generation block size are fixed.

Quality Score increases monotonically with locality at both horizons in all three seeds (Table 4). Sink exclusion improves mean Quality by 0.56/0.72 points at 10/20 seconds; restricting attention further to the jointly denoised block adds 0.41/1.19 points. This additional gain supports local cache construction beyond sink exclusion alone. Frame-local attention achieves the highest Quality Score, with its gain over block-local attention driven primarily by higher Dynamic Degree (S.Table 6). Its slightly lower mean Subject Consistency at both horizons indicates a trade-off between dynamics and subject consistency in this comparison. We therefore adopt block-local cache construction as the default across training stages, preserving attention within each jointly denoised block. Figure 5 compares variants at 5 seconds; per-seed results appear in Supplementary Section S5.

Table 4: **Ablation of cache-construction locality.** Equal-weight means over three training seeds on RF200, with identical denoising context. Per-seed metrics are in S.Table 6.

	10 seconds			20 seconds		
Cache construction	Quality	Aesthetic Quality	Subject Consistency	Quality	Aesthetic Quality	Subject Consistency
Full context	79.620	52.177	87.715	76.809	48.023	80.895
Sink-isolated	80.178	52.648	88.214	77.524	49.074	82.125
Block-local	80.585	53.051	**89.112**	78.712	50.206	**84.000**
Frame-local	**80.970**	**53.155**	88.822	**79.310**	**50.221**	83.532



Figure 5: **Qualitative cache-locality ablation.** Four TF variants at 5 seconds (Seed 1), using the same prompt and sampling settings. The leftmost panel is the full-context model's initial frame.

Training efficiency. Local cache construction also reduces teacher-forcing training time. On eight GPUs, median step time decreases from 5.12 s with full context to 4.13 s with block-local attention and 4.00 s with frame-local attention, corresponding to 1.24× and 1.28× speedups. Matched configurations, data streams, and timing protocols are detailed in Supplementary Section S9.

5 Conclusion

We presented **Local Forcing**, an attention-mask modification that separates contextual denoising from block-local KV-cache construction. The same rule applies across teacher forcing, consistency distillation, self-rollout distillation, and inference without additional parameters, losses, or forward passes. Experiments demonstrate improvements in overall video quality across the evaluated training frameworks and inference methods, together with a 1.24× speedup in teacher-forcing training. In Causal Forcing, LF improves both dynamics and consistency over minute-level rollouts with short-horizon DMD and relative RoPE, without additional memory or compression modules. These findings support local cache construction as a foundation for long autoregressive video rollouts.

AI USE STATEMENT

AI tools assisted literature discovery, coding and analysis, and manuscript and figure preparation. The authors are responsible for the final content and conclusions.

REPRODUCIBILITY STATEMENT

Supplementary Section S1 specifies training and evaluation protocols. Supplementary Sections S5–S12 provide ablation settings, timing methodology, mask implementation, and inference configurations; the tables report all evaluated quality dimensions.

ETHICS STATEMENT

Local Forcing inherits the biases and misuse risks of its underlying video generators, including deceptive media and unauthorized depictions of individuals. Improved long-video generation calls for appropriate consent and clear disclosure of synthetic content when deployed.

REFERENCES

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NeurIPS*, 2015.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *NeurIPS*, 2024.

Shuo Chen, Cong Wei, Sun Sun, Tiancheng Shen, Ping Nie, Kai Zou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. In *ICML*, 2026.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Lol: Longer than longer, scaling video generation to hour. In *CVPR*, 2026a.

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. In *ICLR*, 2026b.

Yusuf Dalva and Pinar Yanardag. Adastate: Self-evolving anchors for streaming video generation. *arXiv preprint arXiv:2605.30349*, 2026.

Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, and Jun Xiao. Vid-gpt: Introducing gpt-style autoregressive generation in video diffusion models. *arXiv preprint arXiv:2406.10981*, 2024.

Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, Jun Xiao, and Long Chen. Ca2-vdm: Efficient autoregressive video diffusion model with causal generation and cache sharing. In *ICML*, 2025.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, et al. Infinite worlds with versatile interactions. *arXiv preprint arXiv:2607.07534*, 2026.

Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and Lu Jiang. Long context tuning for video generation. In *ICCV*, 2025.

Yuwei Guo, Ceyuan Yang, Hao He, Yang Zhao, Meng Wei, Zhenheng Yang, Weilin Huang, and Dahua Lin. End-to-end training for autoregressive video diffusion via self-resampling. In *ECCV*, 2026.

William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. In *NeurIPS*, 2022.

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In *CVPR*, 2025.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *NeurIPS*, 2022.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *NeurIPS*, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE TPAMI*, 2026.

Fan Jiang, Zhaoxu Sun, Mengchao Wang, Ziyu Zhu, Chiyu Wang, Yunpeng Zhang, Wenlin Liu, Yun Wang, Xue Zheng, Rui Sun, et al. ABot-World-0: Infinite interactive world rollout on a single desktop gpu. *arXiv preprint arXiv:2607.19191*, 2026.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. In *NeurIPS*, 2024.

Youngrae Kim, Qixin Hu, C.-C. Jay Kuo, and Peter A. Beerel. Memrope: Training-free infinite video generation via evolving memory tokens. In *ECCV*, 2026. To appear.

Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. In *ICML*, 2024.

Haodong Li, Shaoteng Liu, Zhe Lin, and Manmohan Chandraker. Rolling sink: Bridging limited-horizon training and open-ended testing in autoregressive video diffusion. *arXiv preprint arXiv:2602.07775*, 2026a.

Jiatong Li, Leo Liang, Linghe Kong, and Yulun Zhang. Freqforcing: Autoregressive long video generation via spectral self-anchoring. *arXiv preprint arXiv:2607.27110*, 2026b.

Wuyang Li, Wentao Pan, Po-Chien Luan, Yang Gao, and Alexandre Alahi. Stable video infinity: Infinite-length video generation with error recycling. In *ICLR*, 2026c.

Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *ICLR*, 2026.

Xiaofeng Mao, Shaohao Rui, Kaining Ying, Bo Zheng, Chuanhao Li, Mingmin Chi, and Kaipeng Zhang. Packforcing: Short video training suffices for long video sampling and long context inference. In *ECCV*, 2026.

MiniMax. MiniMax H3. https://github.com/MiniMax-AI/MiniMax-H3, 2026. Open-weight omni-modal audio-video generation model. Accessed: 2026-08-28.

Ryan Po, Eric Ryan Chan, Changan Chen, and Gordon Wetzstein. Bagger: Backwards aggregation for mitigating drift in autoregressive video diffusion models. In *CVPR*, 2026.

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.

Haonan Qiu, Menghan Xia, Yong Zhang, Yingqing He, Xintao Wang, Ying Shan, and Ziwei Liu. Freenoise: Tuning-free longer video diffusion via noise rescheduling. In *ICLR*, 2024.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. In *ICML*, 2024.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

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

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. In *ICML*, 2025.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, et al. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Team Wan, 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.

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

Jing Wang, Fengzhuo Zhang, Xiaoli Li, Vincent Y. F. Tan, Tianyu Pang, Chao Du, Aixin Sun, and Zhuoran Yang. Error analyses of auto-regressive video diffusion models: A unified framework. *JMLR*, 2026.

Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *ICLR*, 2020.

Wenming Weng, Ruoyu Feng, Yanhui Wang, Qi Dai, Chunyu Wang, Dacheng Yin, Zhiyuan Zhao, Kai Qiu, Jianmin Bao, Yuhui Yuan, et al. Art·v: Auto-regressive text-to-video generation with diffusion models. In *CVPR Workshops*, 2024.

Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models. In *CVPR Workshops*, 2025.

Gangwei Xu, Qihang Zhang, Jiaming Zhou, Xing Zhu, Yujun Shen, Xin Yang, and Yinghao Xu. Next forcing: Causal world modeling with multi-chunk prediction. *arXiv preprint arXiv:2606.11187*, 2026.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

Lingxiao Yang, Liu Liu, Moran Li, Han Feng, Wenjian Cao, Jiangning Zhang, and Ye Shi. In-context forcing: Uncovering context effects in autoregressive video diffusion. *arXiv preprint arXiv:2608.05237*, 2026.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025.

Hidir Yesiltepe, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. In *CVPR*, 2026.

Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression. In *ICML*, 2026.

Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. Nuwa-xl: Diffusion over diffusion for extremely long video generation. In *ACL*, 2023.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024a.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024b.

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 *CVPR*, 2025.

Yonghao Yu, Lang Huang, Runyi Li, Zerun Wang, and Toshihiko Yamasaki. Video-mirai: Autoregressive video diffusion models need foresight. *arXiv preprint arXiv:2606.03971*, 2026.

Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. In *NeurIPS*, 2025.

Min Zhao, Guande He, Yixiao Chen, Hongzhou Zhu, Chongxuan Li, and Jun Zhu. Riflex: A free lunch for length extrapolation in video diffusion transformers. In *ICML*, 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 *ICML*, 2026.

Junhao Zhuang, Shiyi Zhang, Yuxuan Bian, Yaowei Li, Yawen Luo, Yijun Liu, Weiyang Jin, Songchun Zhang, Xianglong He, Xuying Zhang, et al. Self gradient forcing: Native long video extrapolation. *arXiv preprint arXiv:2607.20368*, 2026.

SUPPLEMENTARY MATERIAL

S1 EXPERIMENTAL PROTOCOLS

Training and generation. The teacher-forcing comparison in Table 1 uses official Wan2.1-T2V-1.3B initialization and 10,000 training steps on Open-Sora-Plan. We evaluate the final live checkpoints from three training runs, denoted Seed 1, Seed 2, and Seed 3, and average their results with equal weight. Generation uses 50 UniPC steps, CFG 3, a 21-latent-frame attention window with relative positions, and seven retained sink frames. The four-way locality ablation uses the same settings and all three training seeds, with equal-weight averaging. CF and SGF use 500 DMD steps from CF consistency-distilled initialization, retaining their respective training and sampling procedures. The CF VBench Long and MovieGen-128 evaluations reuse the same checkpoint pair; SF's corresponding configuration is given in Supplementary Section S3. CF uses four denoising steps and a 21-latent-frame denoising window including seven sink frames, without additional memory or compression modules. The CF rollouts on MovieGen-128 use relative RoPE.

Prompts and rollout horizons. All benchmark evaluations use a fixed generation seed, with paired prompts and initial noise within each comparison. We evaluate RF200 in VBench's custom-input mode; the 10-second videos are prefixes of the corresponding 20-second rollouts. VBench Long evaluates the union of 326 official quality prompts, with the appropriate prompt subset for each metric; 30-second results use prefixes of the same 60-second rollouts. All 240-second benchmark comparisons use MovieGen-128, the same 128 prompts randomly sampled from MovieGen; their zero-based indices are supplied in `supplementary/moviegen_original_indices.json`. Scores across horizons cover nested prefixes, not equal-length tails.

Metrics and aggregation. We report Aesthetic Quality, Background Consistency, Imaging Quality, Motion Smoothness, Subject Consistency, Temporal Flickering, and Dynamic Degree. Quality Score for VBench Long and MovieGen-128 follows the official leaderboard min–max normalization, with Dynamic Degree weighted by 0.5 and each remaining dimension by 1, divided by 6.5. RF200 uses the same weights on raw metrics, without min–max normalization; its Quality Score is therefore a separate diagnostic. Temporal Flickering uses the official static-video filter, which can retain different numbers of videos in the two arms (24 versus 11 prompts for SGF MovieGen-128).

Cross-setting trends. Across the five methods and three horizons in Section 4.2, Dynamic Degree and Subject Consistency improve jointly in 11 of 15 comparisons. Including Self Forcing (S.Table 4) brings this to 12 of 18; the 30/60-second evaluations share checkpoints and nested prefixes.

Figures and interpretation. Figure 1(a) uses complete, unretouched frames from two 240-second Local Forcing rollouts at 16 fps. Frames are sampled at 0, 60, 120, 180, and 239.9375 seconds, with the last timestamp rounded to 4:00; the rows show different examples rather than a baseline comparison. It illustrates appearance preservation, while temporal behavior is evaluated separately through video metrics. Figure 1(b) uses the complete normalized seven-dimension Quality Score for CF, SGF, and the three inference methods from Table 2, at 30- and 60-second nested prefixes. The inference comparisons reuse the CF Base/LF-trained checkpoint pair without method-specific retraining; Self Forcing's benchmark results remain in S.Table 4. In Figure 3, Q and K denote queries and keys, each attention-matrix cell represents a generation block, and the DiT masks and cache slots are schematic. Its generated thumbnails are illustrative.

S2 BLINDED MODEL-BASED PREFERENCE ASSESSMENT

We use Astra (`gpt-6-astra`) to assess 128 matched baseline/LF pairs for each of CF, SF, SGF, Deep Forcing, Infinity-RoPE, and MemRoPE, totaling 768 pairs and 1,536 judgments. All six methods use MovieGen-128, with prompt indices listed in `supplementary/moviegen_original_indices.json`, with the same source videos as the benchmark comparisons. Each video is represented by 64 uniformly spaced, full-frame samples from 0 to 239.9375 seconds, arranged into eight paired contact sheets. Each pair is evaluated in two fresh contexts with anonymous A/B positions swapped and initial positions balanced within each method; method identities and

benchmark scores are hidden. The rubric considers prompt adherence, visual integrity, temporal coherence, and meaningful motion. A preference requires both presentations to favor the same underlying model; ties and unresolved judgments remain in the denominator of 128.

S.Table 1 reports preferences over the four-minute sampled sequence, with LF preferred in 89.1%–97.7% of pairs across the six methods. Each presentation also requests a secondary judgment on samples from the final minute, using earlier frames only as context; LF preference rates range from 85.2% to 96.1% (S.Table 2). These results measure model-based sampled-frame preference, not human preference or full-rate temporal smoothness.

S.Table 1: **Blinded Astra preferences over four-minute videos.** Counts (percentages) over all 128 paired prompts per method, including ties and unresolved judgments.

Method	LF preferred	Baseline preferred	Tie	Unresolved
Causal Forcing	125 (97.7%)	3 (2.3%)	0 (0.0%)	0 (0.0%)
Self Forcing	120 (93.8%)	5 (3.9%)	0 (0.0%)	3 (2.3%)
Self Gradient Forcing	114 (89.1%)	10 (7.8%)	0 (0.0%)	4 (3.1%)
Deep Forcing	119 (93.0%)	6 (4.7%)	0 (0.0%)	3 (2.3%)
Infinity-RoPE	116 (90.6%)	8 (6.2%)	0 (0.0%)	4 (3.1%)
MemRoPE	118 (92.2%)	9 (7.0%)	0 (0.0%)	1 (0.8%)

S.Table 2: **Astra preferences for the final minute (secondary assessment).** Counts (percentages) over all 128 paired prompts per method, including ties and unresolved judgments.

Method	LF preferred	Baseline preferred	Tie	Unresolved
Causal Forcing	123 (96.1%)	2 (1.6%)	1 (0.8%)	2 (1.6%)
Self Forcing	121 (94.5%)	3 (2.3%)	1 (0.8%)	3 (2.3%)
Self Gradient Forcing	109 (85.2%)	15 (11.7%)	0 (0.0%)	4 (3.1%)
Deep Forcing	118 (92.2%)	6 (4.7%)	0 (0.0%)	4 (3.1%)
Infinity-RoPE	117 (91.4%)	9 (7.0%)	0 (0.0%)	2 (1.6%)
MemRoPE	115 (89.8%)	9 (7.0%)	2 (1.6%)	2 (1.6%)

First-30-second assessment. We also assess the first 30 seconds for all six methods using the same judge and two-order consensus rule (S.Table 3). Each video is represented by 32 uniformly spaced frames from 0 to 29.9375 seconds, arranged into four paired contact sheets; all MovieGen-128 prompts are included. LF is preferred in 57.0%–81.2% of pairs across the six methods.

S.Table 3: **Blinded Astra preferences on the first 30 seconds.** Counts (percentages) over all 128 matched pairs per method, including ties and unresolved judgments.

Method	LF preferred	Baseline preferred	Tie	Unresolved
Causal Forcing	104 (81.2%)	11 (8.6%)	3 (2.3%)	10 (7.8%)
Self Forcing	73 (57.0%)	35 (27.3%)	2 (1.6%)	18 (14.1%)
Self Gradient Forcing	79 (61.7%)	31 (24.2%)	0 (0.0%)	18 (14.1%)
Deep Forcing	81 (63.3%)	31 (24.2%)	0 (0.0%)	16 (12.5%)
Infinity-RoPE	75 (58.6%)	41 (32.0%)	2 (1.6%)	10 (7.8%)
MemRoPE	82 (64.1%)	25 (19.5%)	2 (1.6%)	19 (14.8%)

S3 Additional Results with Self Forcing

We integrate Local Forcing into Self Forcing (Huang et al., 2025) using a matched pair trained for 1,500 DMD steps from SF ODE initialization. The baseline and Local Forcing variants retain SF's sampling and training recipe, with their respective cache-construction rules applied during training and inference. VBench Long and MovieGen-128 use the same checkpoint pair, a fixed generation

seed, and matched initial noise, following the evaluation protocols in Section 4.1. S.Table 4 retains all seven dimensions, normalized Quality Score, and its change at each horizon.

On VBench Long, Quality Score improves by 0.53 points at 30 seconds and 0.45 points at 60 seconds. At both horizons, Local Forcing improves Aesthetic Quality, Background and Subject Consistency, Motion Smoothness, and Temporal Flickering, while Dynamic Degree and Imaging Quality decrease. On MovieGen-128 at 240 seconds, all seven dimensions improve, with a Quality Score gain of 2.07 points. Dynamic Degree increases by 7.10 points and Aesthetic Quality by 2.15 points, alongside higher visual consistency and motion smoothness.

S.Table 4: **Local Forcing with Self Forcing.** The same checkpoints are evaluated on VBench Long at 30/60 seconds and MovieGen-128 at 240 seconds. Shading marks the better score in each pair.

Method	Aesthetic Quality	Background Consistency	Imaging Quality	Motion Smoothness	Subject Consistency	Temporal Flickering	Dynamic Degree	Quality Score	Δ
VBench Long: 30 seconds									
Self Forcing (Huang et al., 2025)	60.91	96.85	**71.09**	97.92	98.18	98.75	**55.56**	83.65	—
+ Local Forcing	**61.92**	**97.42**	70.71	**98.67**	**98.24**	**98.97**	53.15	**84.18**	+0.53
VBench Long: 60 seconds									
Self Forcing (Huang et al., 2025)	60.81	96.65	**71.13**	97.81	97.87	98.75	**56.16**	83.53	—
+ Local Forcing	**61.68**	**97.33**	70.23	**98.62**	**98.04**	**98.96**	53.06	**83.98**	+0.45
MovieGen-128: 240 seconds									
Self Forcing (Huang et al., 2025)	55.43	95.88	68.31	97.73	96.82	97.32	48.09	80.66	—
+ Local Forcing	**57.58**	**96.40**	**69.03**	**98.53**	**97.48**	**98.36**	**55.19**	**82.73**	+2.07

S4 Full TF–CD–DMD Training

We additionally compare complete training chains using Seed 2, starting both variants from the same official Wan2.1 initialization. Each undergoes 10,000 TF steps, 5,000 CD steps, and 1,000 DMD iterations (200 generator updates), with matched training data, per-rank data order, and training budgets. The baseline uses full-context cache construction throughout; LF uses block-local construction throughout training and inference. The final DMD EMA checkpoints are evaluated with four denoising steps, three-latent-frame blocks, a 21-latent-frame window, seven sink frames, and relative RoPE. Both variants use the same 326 VBench Long prompts and paired generation noise, following Supplementary Section S1.

We report this comparison separately because our locally trained CD checkpoints provide a weaker starting point for long rollouts than the released CF CD checkpoint used in the main experiments. Nevertheless, under the matched full-chain setup, LF improves Quality Score at both 30 and 60 seconds, by 0.30 and 0.79 points, respectively (S.Table 5). Imaging Quality improves by 2.57/3.25 points and Subject Consistency by 0.30 points at both horizons, supporting the use of local cache construction across the complete training pipeline.

S.Table 5: **Full TF–CD–DMD training (Seed 2).** VBench Long results from matched training chains starting at official Wan initialization. Shading marks the better score within each pair.

Method	Aesthetic Quality	Background Consistency	Imaging Quality	Motion Smoothness	Subject Consistency	Temporal Flickering	Dynamic Degree	Quality Score	Δ
30 seconds									
Full-chain baseline	57.77	**96.09**	66.91	98.35	97.33	**99.20**	**59.91**	82.96	—
+ Local Forcing	**58.13**	95.63	**69.48**	**98.49**	**97.63**	98.86	59.35	**83.26**	+0.30
60 seconds									
Full-chain baseline	54.89	**95.68**	63.83	**98.29**	96.68	**99.13**	62.55	81.98	—
+ Local Forcing	**55.40**	95.22	**67.08**	98.25	**96.98**	98.85	**67.59**	**82.77**	+0.79

S5 Full Cache-Locality Ablation

S.Table 6 reports all evaluated dimensions for each of the three seeds; Table 4 averages their unrounded scores equally. Within each seed, all four variants use official Wan2.1-T2V-1.3B initialization, 10,000 TF steps, and identical Open-Sora-Plan examples and per-rank order. Final live checkpoints are evaluated with 50 UniPC steps, CFG 3, a fixed generation seed, a 21-latent-frame attention window with relative positions, and seven retained sink frames, at 832×480 resolution and 16 fps. Each model generates 320 frames for every RF200 prompt without exclusions; the 10-second results use the first 160 frames.

Full-context encoding attends to all available tokens within the retained window. Sink-isolated encoding preserves clean-context attention for sink frames F_0–F_6 but prevents clean queries from F_7 onward from attending to them. This follows the global-context exclusion in Appendix A of Liu et al. (2026), retaining recent context during cache updates. Our matched TF setup does not reproduce Rolling Forcing's rolling schedule or context sizes. Block-local restricts clean queries to all spatial tokens in their three-latent-frame generation block; frame-local further restricts them to their own latent frame. These rules apply during training and inference, with denoising attention and generation block size unchanged. Full-context and Block-local means match Table 1. Figure 5 shows complete, unretouched Seed 1 frames for the Tokyo street prompt at 5 seconds, with matched generation seeds and sampling settings. The initial-frame reference is the full-context model's output, not a shared conditioning image.

S.Table 6: **Full RF200 cache-locality ablation by training seed.** All prompts are evaluated for every model and horizon; shading marks the highest score within each seed and horizon. Quality excludes Overall Consistency (text–video alignment; Supplementary Section S1).

Cache construction	Aesthetic Quality	Background Consistency	Imaging Quality	Motion Smoothness	Subject Consistency	Temporal Flickering	Dynamic Degree	Overall Consistency	Quality
Seed 1: 10 seconds									
Full context	50.782	90.772	59.082	98.068	85.766	97.141	63.000	23.429	78.940
Sink-isolated	52.418	91.732	61.718	98.010	87.635	97.081	60.500	23.690	79.822
Block-local	53.328	**92.458**	63.613	98.322	89.273	**97.373**	56.500	23.493	80.403
Frame-local	**53.604**	92.212	**64.179**	**98.394**	**89.715**	97.277	**67.000**	**23.715**	**81.366**
Seed 1: 20 seconds									
Full context	45.730	87.164	51.968	97.105	77.527	96.132	68.500	22.452	75.366
Sink-isolated	48.592	88.911	56.949	96.889	81.230	95.981	61.000	23.400	76.777
Block-local	50.415	**90.331**	60.320	97.881	84.358	**96.920**	60.000	23.468	78.496
Frame-local	**50.583**	89.850	**61.219**	**97.971**	**84.446**	96.798	**73.500**	**23.712**	**79.633**
Seed 2: 10 seconds									
Full context	53.626	92.954	64.980	98.650	90.613	**97.868**	47.000	23.451	80.337
Sink-isolated	53.711	92.826	64.767	**98.656**	90.734	97.843	52.000	23.664	80.698
Block-local	53.649	**93.117**	**65.448**	98.540	**91.039**	97.623	51.500	23.672	80.795
Frame-local	**53.763**	92.325	64.780	98.538	90.039	97.545	**58.500**	**23.713**	**80.960**
Seed 2: 20 seconds									
Full context	50.319	90.682	61.769	**98.421**	85.397	**97.682**	50.500	23.030	78.388
Sink-isolated	50.547	90.535	61.367	98.357	85.460	97.541	52.500	23.624	78.470
Block-local	**51.224**	**91.168**	**62.974**	98.242	**86.477**	97.265	54.500	23.522	79.169
Frame-local	50.913	90.204	61.922	98.280	85.003	97.287	**65.000**	**23.669**	**79.401**
Seed 3: 10 seconds									
Full context	52.122	91.251	61.539	**97.912**	86.767	**96.939**	61.500	23.361	79.582
Sink-isolated	51.816	91.068	62.398	97.809	86.272	96.728	68.000	23.505	80.014
Block-local	**52.177**	**91.386**	**63.671**	97.897	**87.023**	96.729	69.500	23.643	80.559
Frame-local	52.098	91.092	62.374	97.863	86.713	96.661	**74.000**	**23.690**	**80.585**
Seed 3: 20 seconds									
Full context	48.020	88.257	55.797	97.331	79.760	**96.210**	66.000	22.893	76.673
Sink-isolated	48.083	88.365	57.846	96.842	79.684	95.546	72.500	23.202	77.325
Block-local	48.980	**88.901**	**60.212**	97.218	**81.167**	95.838	75.500	23.246	78.472
Frame-local	**49.167**	88.657	58.975	**97.468**	81.148	96.151	**82.500**	**23.496**	**78.895**

S6 KV Cache Construction and Window Length

Local Forcing restricts cache construction without fixing the context window used for denoising. To test whether its benefit extends to a smaller context budget, we repeat the TF comparison with a 13-latent-frame window containing three sink frames, instead of the 21-frame window and seven sink frames used above. All three variants start from official Wan2.1-T2V-1.3B and undergo 10,000 TF steps with matched training data, order, and random seed. Training clips remain 21 latent frames long, with three-frame generation blocks; inference uses relative RoPE and the same RF200 sampling protocol. Sink-isolated construction excludes the three sink frames from subsequent cache updates, whereas block-local construction attends only within the current block.

S.Table 7 reports every evaluated dimension at 10 and 20 seconds. Quality improves progressively from full context to sink isolation to block-local construction at both horizons. Block-local construction improves Quality over full context by 0.51 and 1.28 points, and over sink isolation by 0.23 and 0.83 points, respectively. Together with the 21-frame results in S.Table 6, these results show that the benefit of local cache construction extends across the tested context-window configurations.

S.Table 7: **Cache construction with a shorter context window.** RF200 scores with a 13-latent-frame window and three sink frames. Shading marks the highest score at each horizon; Quality uses the same raw aggregate as S.Table 6.

Cache construction	Aesthetic Quality	Background Consistency	Imaging Quality	Motion Smoothness	Subject Consistency	Temporal Flickering	Dynamic Degree	Overall Consistency	Quality
10 seconds									
Full context	53.069	92.449	63.364	98.526	89.238	97.729	53.500	23.484	80.173
Sink-isolated	53.223	92.738	64.103	98.387	90.054	97.490	**54.000**	**23.869**	80.461
Block-local	**54.078**	**92.968**	**65.798**	**98.638**	**90.918**	**97.808**	48.500	23.820	**80.686**
20 seconds									
Full context	49.315	89.538	58.617	98.045	82.796	97.271	**58.000**	23.098	77.628
Sink-isolated	50.017	90.147	59.903	97.711	84.143	96.823	57.500	23.704	78.076
Block-local	**51.297**	**90.575**	**62.630**	**98.425**	**85.602**	**97.597**	53.500	**23.891**	**78.904**

S7 Frame-Wise Autoregressive Generation

We also test Local Forcing when each generation block contains a single latent frame, rather than the three frames used above. Both variants start from official Wan2.1-T2V-1.3B and undergo 10,000 teacher-forcing steps with matched training data, order, and random seed. The baseline constructs its cache using full context; + Local Forcing restricts cache construction to the current latent frame throughout training and inference, while retaining historical context for denoising. We evaluate the final live checkpoints on RF200 using the sampling settings in Supplementary Section S1; the 10-second clips are prefixes of the same 20-second rollouts. Local Forcing improves Quality by 3.90 and 4.18 points at 10 and 20 seconds, respectively, alongside higher Subject Consistency, Imaging Quality, and Dynamic Degree (S.Table 8). These results extend the benefit of local cache construction to frame-wise autoregressive generation.

S.Table 8: **Frame-wise teacher forcing on RF200.** Both variants generate one latent frame at a time. Quality uses the raw RF200 aggregate. Shading marks the better score within each pair.

Method	Aesthetic Quality	Background Consistency	Imaging Quality	Motion Smoothness	Subject Consistency	Temporal Flickering	Dynamic Degree	Quality	Δ
10 seconds									
Frame-wise TF	47.90	87.89	52.83	96.51	79.61	**95.43**	57.50	75.22	—
+ Local Forcing	**49.89**	**88.95**	**57.63**	**96.81**	**82.71**	95.04	**86.50**	**79.12**	+3.90
20 seconds									
Frame-wise TF	42.77	84.27	46.21	**95.58**	71.60	**94.44**	63.00	71.75	—
+ Local Forcing	**45.15**	**85.86**	**52.89**	95.19	**75.22**	93.50	**91.50**	**75.93**	+4.18

S8 WAN2.2 TEACHER FORCING

We further evaluate Local Forcing on Wan2.2-TI2V-5B. Both variants start from the official weights and undergo 10,000 TF steps on Open-Sora-Plan, using the first 31 latent frames of each cached training video, with matched data order and random seed. Generation uses an independent first latent frame followed by three-frame blocks; LF restricts cache construction to each latent frame during both training and inference, without changing the generation block size. The final live checkpoints generate 20-second videos at 1280×704 resolution and 24 fps, using 50 UniPC steps, CFG 3, relative RoPE, a 31-latent-frame attention window, and seven sink frames. Both models use the full RF200 prompt set, with the first 10 seconds evaluated separately.

S.Table 9 shows Quality gains of 0.62 and 1.64 points at 10 and 20 seconds, respectively. At 20 seconds, Dynamic Degree increases by 12.00 points, alongside gains of 1.95, 1.58, and 1.32 in Aesthetic Quality, Subject Consistency, and Background Consistency. These results extend the benefits of local cache construction to this larger backbone.

S.Table 9: **Wan2.2 teacher forcing on RF200.** Wan2.2-TI2V-5B after 10,000 TF steps, evaluated on the full prompt set. LF uses frame-local cache construction. Quality uses the raw RF200 aggregate. Shading marks the better score within each pair.

Method	Aesthetic Quality	Background Consistency	Imaging Quality	Motion Smoothness	Subject Consistency	Temporal Flickering	Dynamic Degree	Quality	Δ
10 seconds									
Wan2.2 TF	53.22	93.85	**66.47**	**99.22**	91.61	**98.57**	39.00	80.38	—
+ Local Forcing	**53.98**	**94.14**	65.93	99.15	**91.80**	98.50	**46.00**	**81.00**	+0.62
20 seconds									
Wan2.2 TF	48.18	90.42	**63.98**	**99.13**	85.24	**98.46**	37.00	77.52	—
+ Local Forcing	**50.13**	**91.73**	63.95	99.03	**86.82**	98.39	**49.00**	**79.16**	+1.64

S9 TEACHER-FORCING TRAINING EFFICIENCY

S.Table 10 reports wall-clock timings from the three TF runs used in the locality ablation. All three use eight GPUs, global batch eight, 21 latent frames at 832×480 resolution, BF16, gradient checkpointing, and Seed 1 for training. Their training-data streams match on all eight ranks, with identical dependency lockfiles and three-frame generation blocks.

Step times are medians over 471 common 20-step intervals between steps 200 and 9,800, extracted from rank-zero heartbeat timestamps. We exclude warmup and checkpoint-overlapping intervals from all three runs; full context saved every 1,000 steps, whereas local variants saved only at the end. Speedups use unrounded median times relative to full context.

S.Table 10: **Teacher-forcing training efficiency.** Median times over matched intervals from historical runs on 8 GPUs with global batch size 8, excluding warmup and checkpoint overhead.

Cache construction	Time / step (s) ↓	Speedup ↑	Time saved
Full context	5.12	1.00×	0.0%
Block-local	4.13	1.24×	19.5%
Frame-local	4.00	1.28×	22.0%

S10 CONDITIONAL LOCALITY OF CACHE CONSTRUCTION

Here C_i denotes the layer-wise K/V produced from the current block and $C_{<i}$ the retained K/V from preceding blocks, as in Section 3. Let c be the text condition, p_i the current block's spatiotemporal token positions, and θ the shared transformer parameters. When video self-attention is the

transformer's only cross-block operation, applying M_{cache} from Eq. 2 at every layer gives

$$\left.\frac{\partial \operatorname{vec}(C_i)}{\partial \operatorname{vec}(C_{<i})}\right|_{x_i,c,p_i,\theta} = 0. \tag{3}$$

Here vec stacks the entries of the layer-wise tensors into a vector. The current block's own keys and values remain active throughout this computation. The derivative holds x_i, c, p_i, and θ fixed; it does not assert independence of x_i from the history used to generate it. Token-wise layers, text cross-attention, and fixed positional operations preserve this locality. Gradients with respect to θ remain available, supporting differentiable cache recomputation.

Implementation of the mask. In our sequential implementation, the current block occupies a contiguous cache tail. The core K/V range selection can be expressed as

```
read_start = block_start if cache_pass else context_start
```

Here `block_start` is the first current-block token, `context_start` is the first token in the retained attention context, and `cache_pass` identifies clean-context cache construction. Both branches read through the current block's end; positional transformations and cache management are unchanged. Integration also routes the cache-pass flag to each layer and updates packed training's clean-to-clean mask; the one-line selection is specific to this contiguous-cache layout.

S11 Pilot Study: Inference-Only Cache Interventions

A matched pilot. Figure 2 uses the native-trained, 1,500-step Causal Forcing EMA checkpoint from a training-rule $\times$ inference-rule crossover. It compares the same checkpoint under native and block-local cache construction, with no further optimization between conditions. The prompt is "Five gray wolf pups frolicking and chasing each other around a remote gravel road, surrounded by grass. The pups run and leap, chasing each other, and nipping at each other, playing." Both conditions share the checkpoint, prompt, generation seed, and initial noise. Generation uses four denoising steps, 832×480 resolution, 16 fps, three latent frames per generation block, and a 21-latent-frame denoising window including seven retained sink frames, with the original absolute positional encoding. Only clean-context video attention changes: native attention or attention within the current three-latent block (Local Forcing). At 30 seconds, native encoding produces oversaturated grass and changes in fur color and texture; local encoding retains a more natural gray-fur appearance and grass color. The figure uses the complete decoded frames at indices 0 and 480 (30 seconds), with the native initial frame as a visual reference rather than a shared conditioning image.

Drift reduction and temporal jitter. A separate diagnostic pairs native and block-local 30-second inference on the first eight official Causal Forcing `demo.txt` prompts using its released chunkwise checkpoint. Several cases better preserve content, color, or background texture. Continuous-video inspection also shows increased block-boundary jitter, varying with checkpoint and prompt, so reduced drift alone does not make inference-only switching a satisfactory method. Applying the local rule during training can mitigate jitter without an extra loss or smoothing module; this is the setting used in the main benchmark tables.

S12 Composition with Training-Free Inference Methods

The lower group at each horizon in Table 2 reuses the CF Base and Local Forcing checkpoints from the training-framework comparison. The weights are fixed when adapting the inference pipelines; no additional training is performed for Deep Forcing, Infinity-RoPE, or MemRoPE. Both arms run each method's official implementation, with the Local Forcing arm restricting only clean-context video attention to the current three-latent-frame block. Deep Forcing retains its deep-sink and participative-compression operations (allocation 21, sink 10, budget 16, recent 4); Infinity-RoPE retains its block-relative positions (window 6, sink 1); MemRoPE retains its dual EMA memory and online positions (cache 12, sink 3, recent 4, EMA coefficients 0.01 and 0.1). These method-specific configurations differ in context budget, so cross-method rows do not isolate the effect of a single inference operator.

All six configurations generate the same 326 official quality prompts with a fixed generation seed and matching initial noise, using four denoising steps, 832×480 resolution, and 16 fps. The 30- and 60-second evaluations use the same 60-second source videos, with no prompt exclusions. Decoding preserves the official Wan VAE's state continuously across video chunks in both arms. Scoring follows the seven-dimension VBench Long protocol in the main text, with the official static filter for Temporal Flickering. The number of retained static videos at 30/60 seconds is 60/66 versus 60/62 for Deep Forcing, 57/58 versus 58/63 for Infinity-RoPE, and 57/57 versus 64/66 for MemRoPE (Base versus Local Forcing).

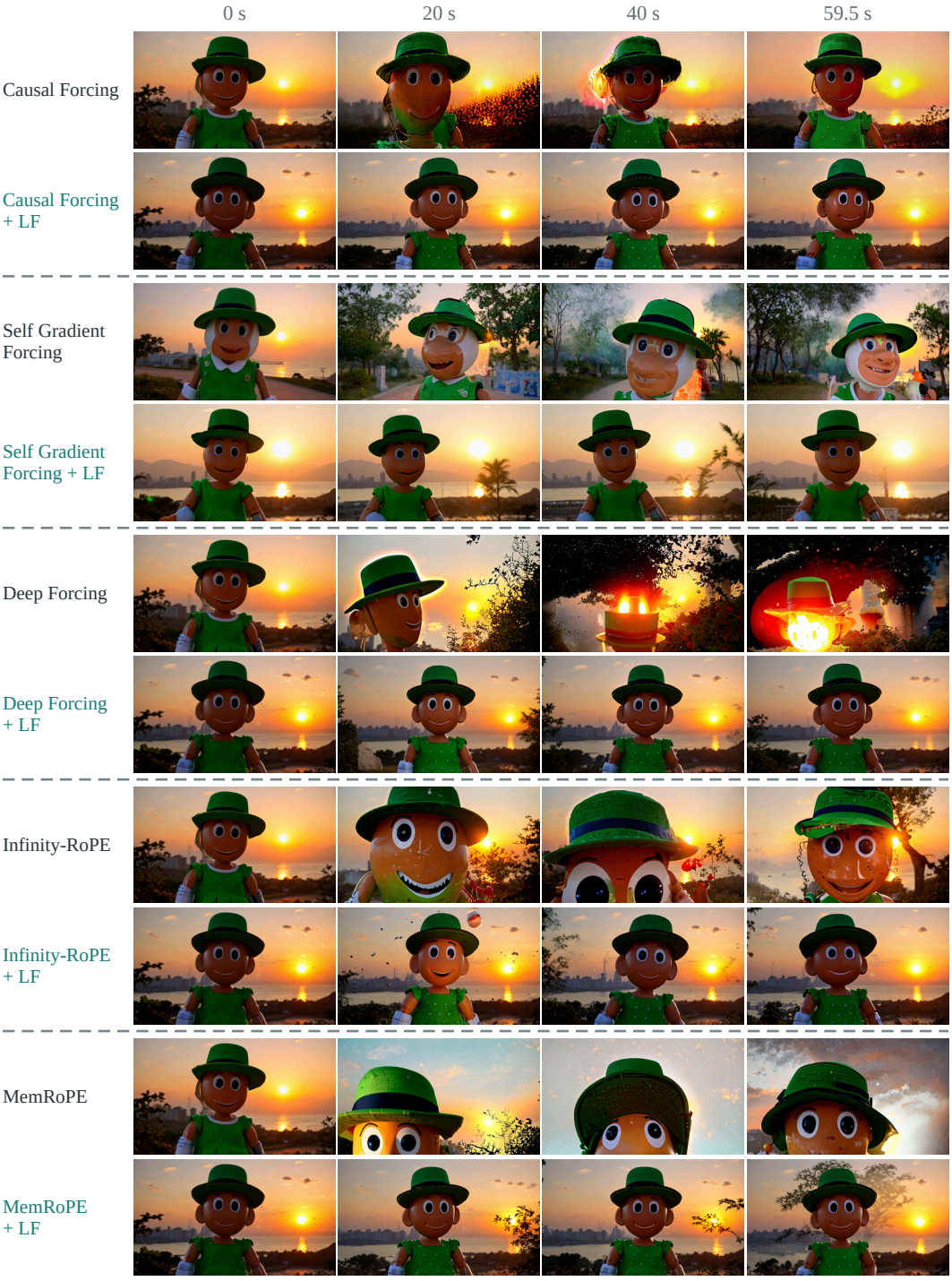
For MovieGen-128 at 240 seconds, all six configurations generate 3,840 frames per prompt with a fixed generation seed and matching initial noise, using the same checkpoints, sampling settings, and continuous decoding as above. They use the same MovieGen-128 prompts as the training-framework comparisons. Scoring and Quality Score normalization follow the same VBench Long implementation. The static filter retains videos from 39/27 prompts for Deep Forcing, 17/16 for Infinity-RoPE, and 36/29 for MemRoPE (Base/Local Forcing).

S13 LIMITATIONS

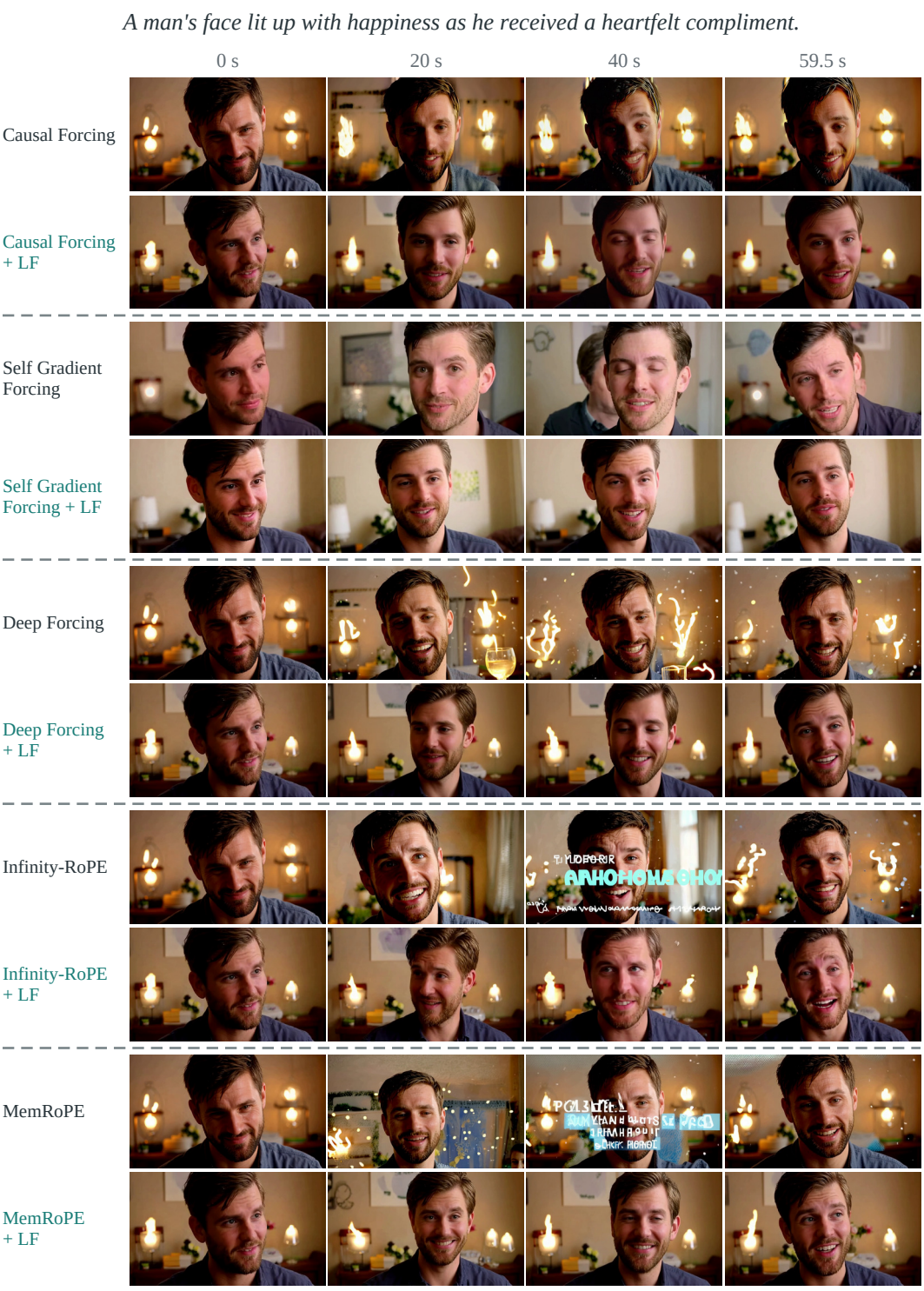
Local Forcing is not a training-free method: it requires a model trained or fine-tuned with local KV-cache construction, incurring additional training cost compared with methods applied directly to existing checkpoints at inference. In addition, our experiments cover backbones with up to 5 billion parameters. Whether the observed benefits extend to larger models, such as Wan2.2-A14B and MiniMax-H3 (MiniMax, 2026), remains to be established through further experiments.

S14 Additional Qualitative Comparisons

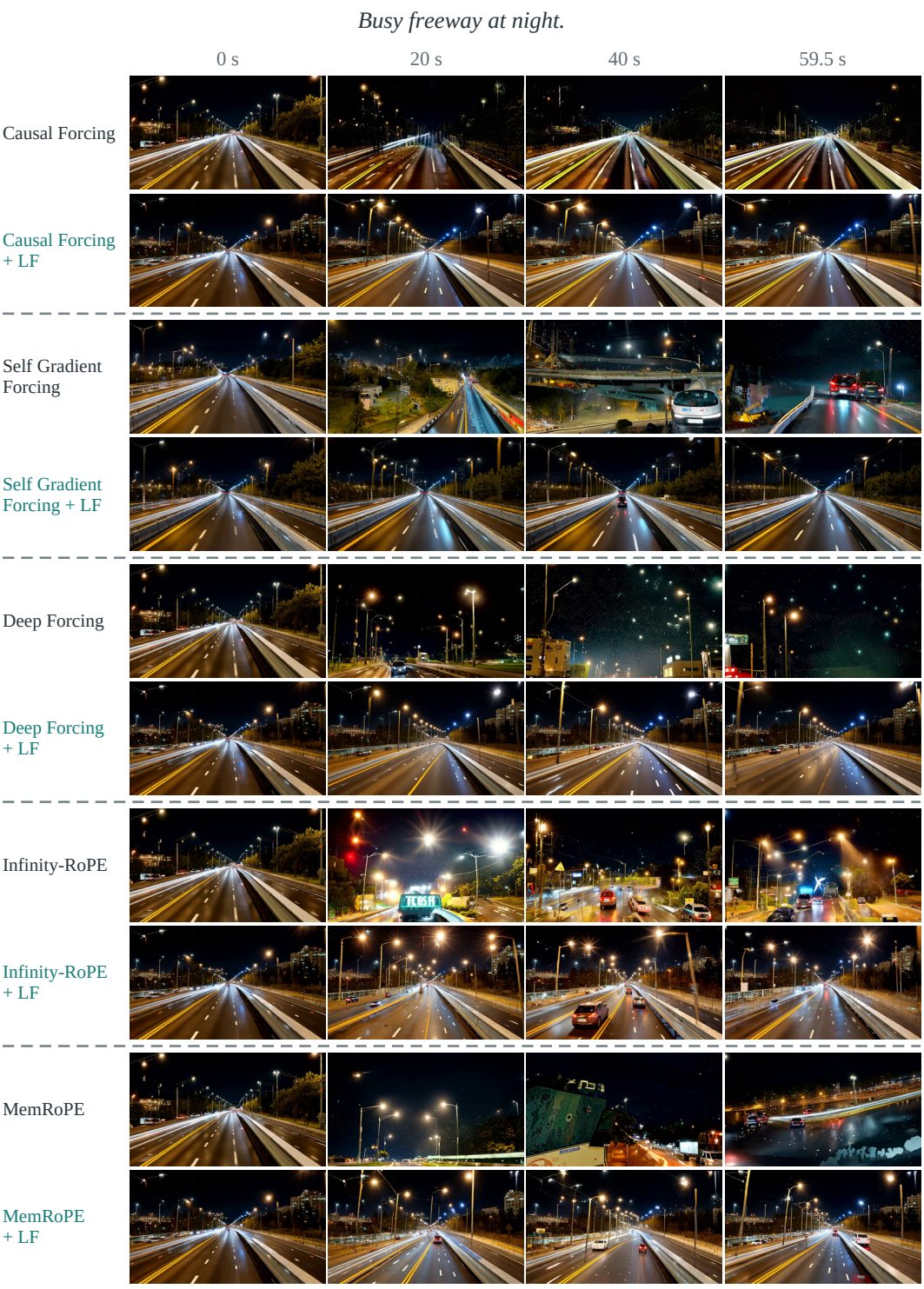
a toy robot wearing a green dress and a sun hat taking a pleasant stroll in Mumbai India during a beautiful sunset



S.Figure 1: Matched-time comparison of each method with and without LF.



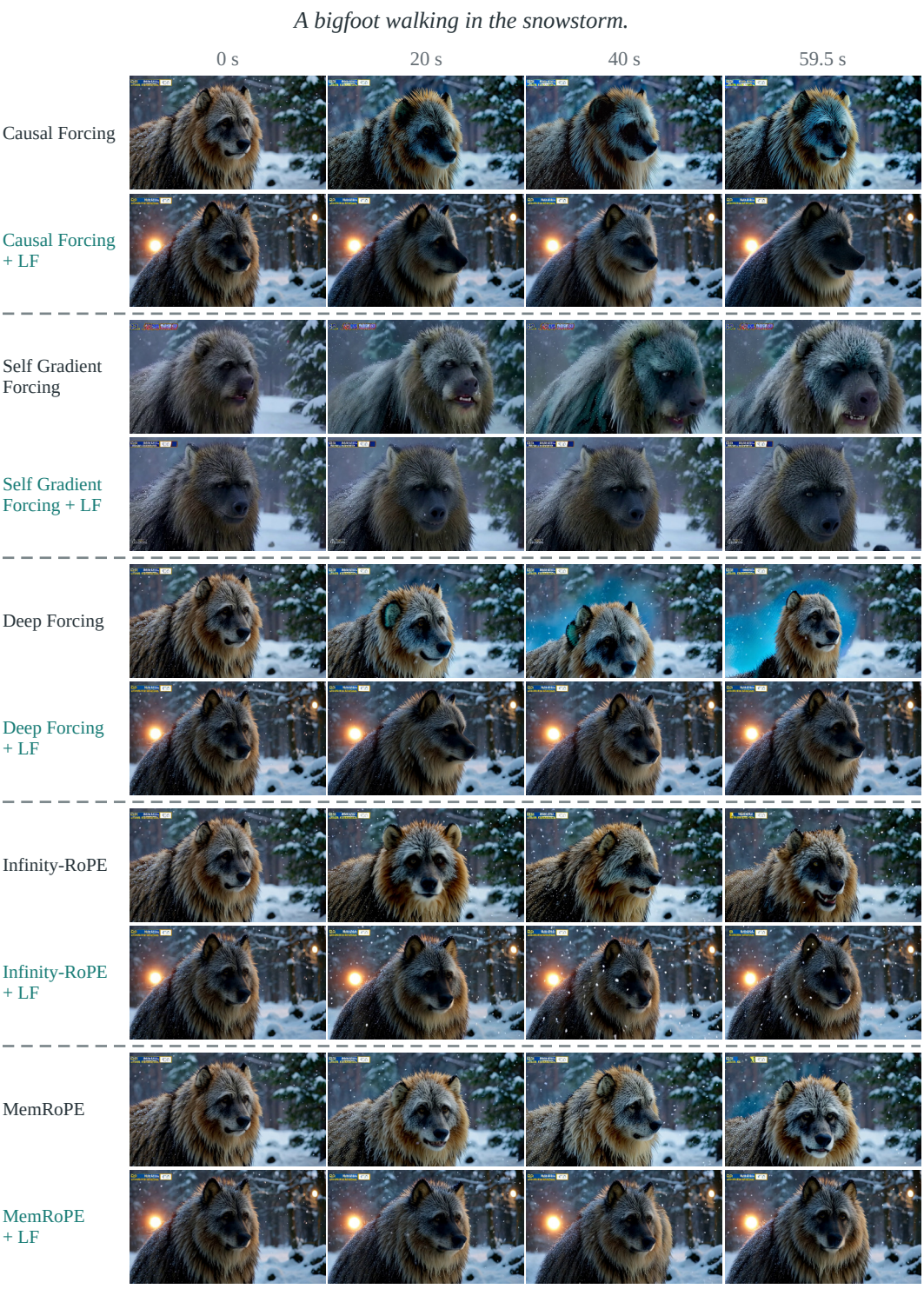
S.Figure 2: Matched-time comparison of each method with and without LF.



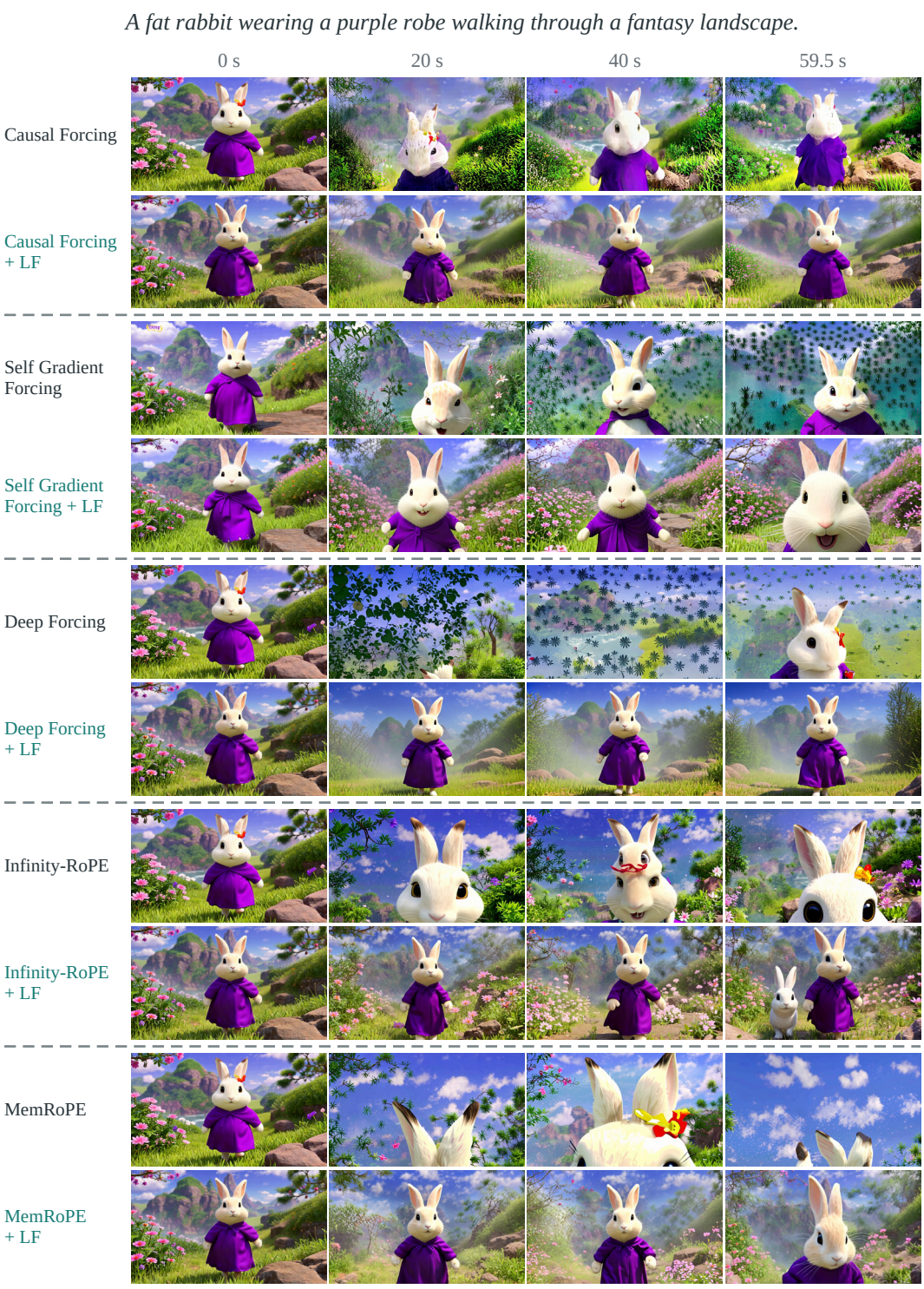
S.Figure 3: Matched-time comparison of each method with and without LF.



S.Figure 4: Matched-time comparison of each method with and without LF.



S.Figure 5: Matched-time comparison of each method with and without LF.



S.Figure 6: Matched-time comparison of each method with and without LF.