STATE FORCING: TELIC INSTRUCTION FOLLOWING FOR STREAMING VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in autoregressive video generation have enabled efficient streaming beyond the native clip lengths of pretrained models. Yet visually coherent rollouts can still exhibit progress regression and re-execution of completed actions. We study *telic instruction following*, which requires completing a requested state transition without regression and preserving its consequence. Our empirical study shows that verified terminal states can guide progress toward completion, and support consequence retention in a frozen generator. Building on this finding, we introduce **State Forcing**, a state-privileged on-policy self-distillation (OPSD) framework. Through complementary ADVANCE and HOLD conditions, a frozen teacher uses offline terminal states to supervise progress and consequence retention along student-generated trajectories. The student retains the original instruction and streaming pipeline, without additional inputs or computation at inference. We also introduce TELICBENCH to jointly evaluate completion and temporal validity over full rollouts. State Forcing outperforms the evaluated baselines in completion and strict success, achieving 37.33% strict success compared with 24.93% for the strongest baseline. These gains are achieved with overall video quality comparable to the distilled streaming baselines.

1 INTRODUCTION

Recent advances in autoregressive video generation (Huang et al., 2026; Yang et al., 2025a) enable efficient streaming beyond pretrained models' native clip lengths. Yet a visually coherent video may still fail to follow its instruction. We focus on *telic* instructions, which describe events with an identifiable culmination and consequent state (Vendler, 1957; Comrie, 1976). For example, an instruction to open a closed door fully and keep it open requires the door to visibly reach and maintain a fully open state. We call this outcome the *terminal state*. Following a telic instruction requires *non-regressing progress* toward that state and *consequence retention* afterward, while allowing unrelated aspects of the scene to evolve.

Autoregressive streaming methods (Huang et al., 2026; Yang et al., 2025a; Liu et al., 2026b) primarily target long-horizon video quality without explicitly supervising telic execution. LongLive and Rolling Forcing (Yang et al., 2025a; Liu et al., 2026b) use attention sinks to preserve initial-frame context, supporting visual consistency but potentially retaining cues from a state the event has already left behind. OPSD-V (Liu et al., 2026a) supplies cleaner history to reduce generation drift, but does not explicitly identify the required terminal state or indicate that it has been reached. These approaches improve temporal coherence, yet a coherent rollout can still regress or repeat a completed action (Figure 1). TempAct (Wang et al., 2026) targets temporal instruction following with fixed-span stage prompts, which may switch before transition completion or persist afterward. Switching based on observed progress would require online state assessment, adding computation and potential decision errors. In our evaluation, TempAct improves completion but still struggles to sustain progress and retain consequences (Appendix D.2.3). These limitations leave open how to teach both progression and retention within a streaming generator without prescribing event timing at inference.

To this end, we investigate whether a *verified terminal chunk* selected from an offline rollout that visibly reaches the requested terminal state can provide the missing supervision. We extract this chunk from a terminal-state interval, retaining cases with compatible execution and without severe subject or scene drift. With model parameters and sampling fixed, replacing the initial attention sink

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.

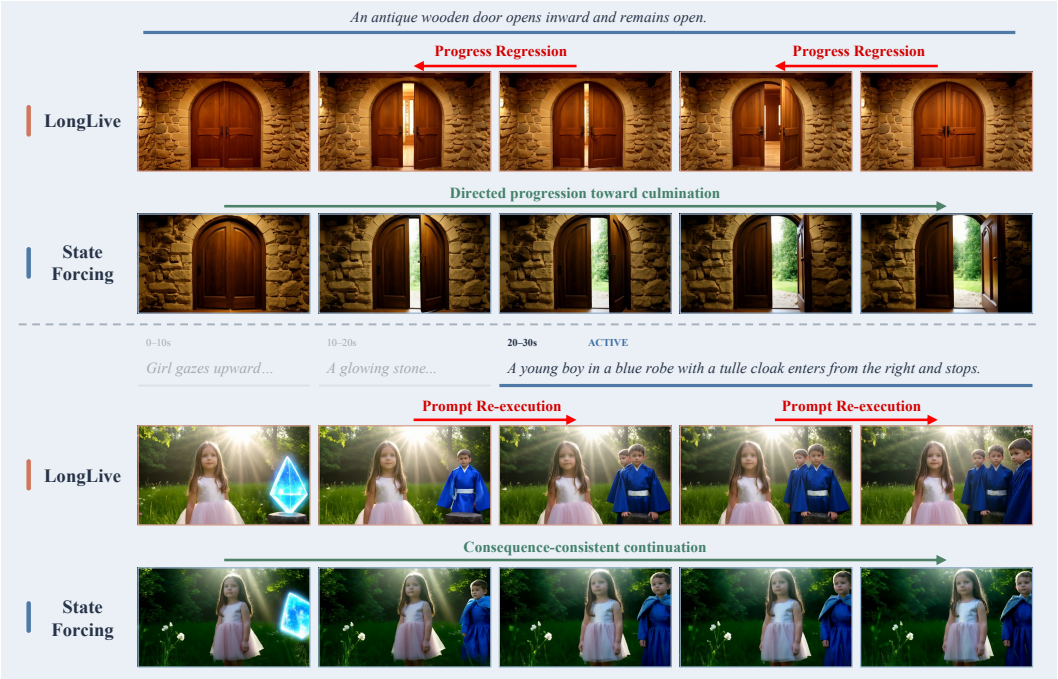


Figure 1: **Telic instruction following: progress and consequence retention.** LongLive regresses during door opening (top) and re-executes character entry instruction after completion (bottom). State Forcing sustains progress toward completion and preserves the achieved result. The bottom example further illustrates that State Forcing can be extended to multi-prompt interactive generation.

with the terminal chunk's encoding encourages progress from an unfinished prefix. Starting from the terminal chunk alone under a *consequent-state description* instead supports retention after completion (Figure 2). These observations suggest using terminal evidence in two ways, according to whether the event is still advancing or has reached its outcome.

We introduce **State Forcing**, which learns telic progression and consequence retention from terminal-state evidence available only during training. Our key idea is to use a verified terminal chunk during training to guide unfinished events toward completion and teach completed events to preserve their consequences. To obtain this evidence, we generate source rollouts offline and use a VLM (OpenAI, 2026) to verify execution and locate terminal intervals. A frozen teacher then provides complementary conditions. In ADVANCE, a verified terminal chunk replaces its initial sink while the unfinished history and original instruction remain; in HOLD, the teacher starts from that chunk under a consequent-state description. The student retains the original instruction and standard streaming context, matching teacher predictions at noisy states visited by its own sampler. Both models update their histories with student-generated chunks, keeping supervision on the student's trajectory. Together, the two conditions teach a single student to progress toward an outcome and preserve it after completion. At inference, this student needs no terminal-state input, phase detector, or additional computation.

Evaluating telic instruction following requires measuring both completion and temporal validity. A completed action may later regress or repeat, while a low violation rate may simply reflect inaction. We introduce TELICBENCH, comprising 150 telic instructions with prompt-specific rubrics for the requested transition and valid continuation. We define three complementary metrics: Completion Rate (CR) for visible completion, Telic Violation Rate (TVR) for the extent of temporal violations, and Telic Success Rate (TSR) for completion without violations.

Experimental results show that State Forcing achieves the highest Completion Rate (51.33%) and Telic Success Rate (37.33%) among the evaluated methods, while reducing Telic Violation Rate to 7.60%. Its Telic Success Rate exceeds the strongest baseline by 12.40 percentage points. Notably, VBench (Huang et al., 2024) scores on MovieGenBench (Polyak et al., 2024) prompts remain comparable to distilled streaming baselines, showing that these gains preserve general video quality.

Our contributions are:

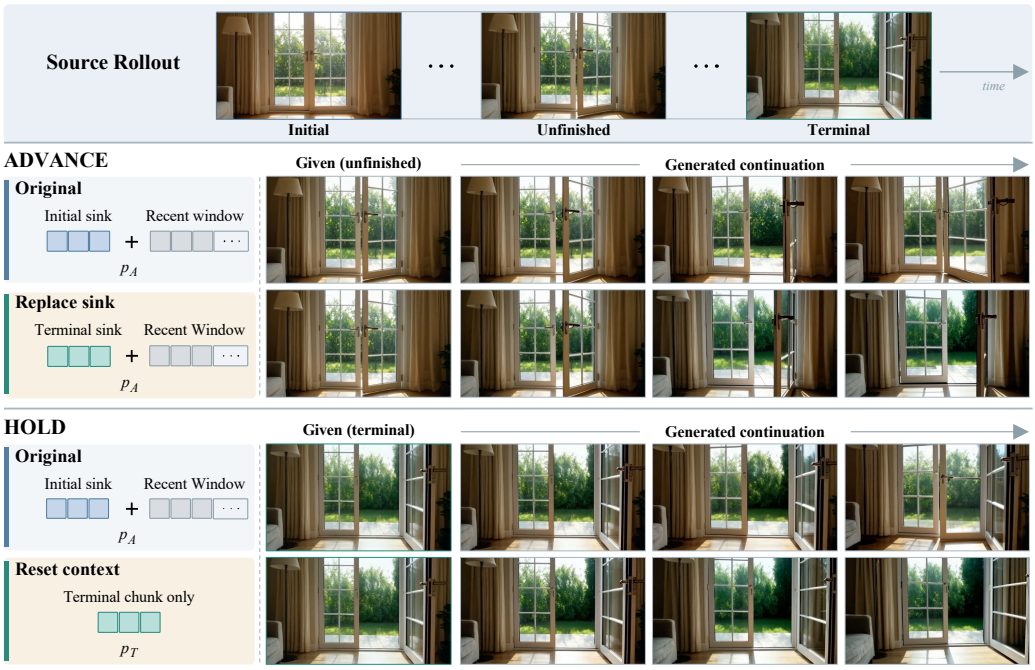


Figure 2: **Empirical motivation for state-privileged supervision.** With model and sampling fixed, a verified terminal chunk from the source rollout supports progress toward completion and consequence retention. p_A denotes a telic instruction and p_T denotes a consequent-state description.

- We study telic instruction following in streaming generation, requiring completion, non-regressing progress, and consequence retention. Our empirical study shows that verified terminal chunks can guide progress and support consequence retention in a frozen generator.
- We introduce STATE FORCING, an OPSD framework that uses offline terminal chunks to construct ADVANCE and HOLD teacher supervision on student-generated trajectories. The student retains standard causal inputs, with no additional inputs or computation at inference.
- We introduce TELICBENCH, with prompt-specific rubrics assessing completion, temporal violation duration, and strict success over full rollouts.
- Extensive experiments demonstrate improved telic execution with comparable video quality.

2 RELATED WORK

Streaming video generation. Building on video diffusion models (Ho et al., 2022; Yang et al., 2025b; Wan et al., 2025), streaming methods (Chen et al., 2024; Kodaira et al., 2026; Kim et al., 2024) seek efficient, coherent generation beyond native clip lengths. Few-step distillation accelerates autoregressive sampling (Yin et al., 2025; Zhu et al., 2026; Gu et al., 2026), while Self Forcing (Huang et al., 2026) mitigates exposure bias through self-generated training histories. Long-horizon methods extend rollout supervision (Yang et al., 2025a; Cui et al., 2026a; Liu et al., 2026b), align teacher–student contexts (Chen et al., 2026), or retain informative history (Ji et al., 2025; Hu et al., 2026; Xu et al., 2026; Zhao et al., 2026b; Henschel et al., 2025; Zhang et al., 2026). Training-free cache and positional interventions extend rollouts (Yi et al., 2025; Li et al., 2026), with LoL (Cui et al., 2026b) targeting sink collapse. Complementing these advances, STATE FORCING learns telic-instruction progression and consequence retention without additional inference cost.

Privileged-context self-distillation. Privileged-context self-distillation strengthens supervision through enriched teacher inputs. Across language, image, and video generation, OPSD (Zhao et al., 2026a), SDFT (Shenfeld et al., 2026), D-OPSD (Jiang et al., 2026), and OPSD-V (Liu et al., 2026a) supervise student-generated trajectories using privileged context, including demonstrations,

targeted images, and real-video history. STATE FORCING instead uses verified terminal states from self-generated rollouts for complementary supervision of progress and consequence retention.

Temporal control and video evaluation. Prior work studies temporal validity and video consistency. Lo et al. (2025) introduce frame-wise denoising guidance for attribute transitions and CAT-Bench for evaluation, while TempAct (Wang et al., 2026) learns action sequencing through planner–executor RL. TC-Bench (Feng et al., 2024), T2V-CompBench (Sun et al., 2025), and L3DE (Chang et al., 2025) evaluate scene transitions, compositionality, and temporal coherence. STATE FORCING learns non-regressing progress and consequence retention under a fixed instruction, without inference-time planning or prompt scheduling. TELICBENCH jointly assesses completion, non-regressing progress, and post-completion consequence retention throughout streaming rollouts.

3 METHOD

STATE FORCING (StF) learns non-regressing progress and consequence retention through state-privileged on-policy self-distillation (Figure 3). We initialize student parameters θ and teacher parameters ϕ from the same checkpoint and keep ϕ frozen. Verified terminal chunks provide the teacher with context for two training branches: ADVANCE for incomplete events and HOLD for completed events. The student retains the original instruction and standard streaming context. Both branches use the same objective to match teacher predictions along student-generated trajectories.

3.1 STATE-PRIVILEGED TEACHER CONTEXTS

We construct teacher contexts from offline source rollouts generated by the initial checkpoint. Each rollout is paired with a telic instruction p_A and a consequent-state description p_T. A VLM verifies the initial state and execution, and localizes a terminal-state interval. We retain compatible execution prefixes and terminal chunks from the same rollout, excluding severe subject or scene drift.

Let $X = (x_0, \ldots, x_{T-1})$ denote a source latent sequence, with chunk length m and recent-history length h, both measured in latent time steps. Choose a chunk boundary P such that $[P - m, P)$ lies within a verified terminal-state interval. Using half-open slices, define

$$G = X_{P-m:P}, \qquad R = X_{P-h:P}, \qquad m \le h \le P \le T. \tag{1}$$

Here G is the terminal chunk and R is recent history ending in G. A training record $d = (X, p_A, p_T, P, b) \in \mathcal{D}_{\mathrm{StF}}$ specifies one admissible branch $b \in \{\text{ADVANCE}, \text{HOLD}\}$. The record determines initialization, not a target continuation.

ADVANCE: progress toward completion. Both models first encode $X_{0:L}$, where $L \le L_{\mathrm{nat}}$ is the conditioning-prefix length and L_{nat} is the pretrained model's native clip length in latent time steps. This prefix must show correct but incomplete execution compatible with the later terminal chunk G. The student keeps its standard cache, consisting of the initial sink and recent history. The teacher replaces only the sink entries with an encoding of G at the sink's temporal positions. Its recent history, temporal cursor, and instruction p_A remain unchanged. The teacher thus observes both the ongoing execution and its terminal state.

HOLD: continuation after completion. The student starts at P with the original sink $X_{0:m}$, recent history R, and instruction p_A. The teacher starts from G under p_T at the same cursor P, using G as its persistent sink and sole initial history. Both models observe the same latest terminal chunk; the teacher's privilege is to isolate that state and describe it as already established. The selected branch stays fixed for the rollout, and context interventions are applied only at initialization. Appendix A.4 specifies exact cache positions and encoding operations.

3.2 ON-POLICY SELF-DISTILLATION

After initialization, the student generates J new chunks under p_A using the base model's few-step sampler. The prefix is excluded from these J chunks. Before chunk j, let C_j^S and C_j^T be the student and teacher caches. Let $p_b = p_A$ for ADVANCE and $p_b = p_T$ for HOLD. At a supervised denoising timestep $\tau \in \mathcal{U}_j$, both models predict denoising velocities at the same student noisy state $z_{j,\tau}^S$:

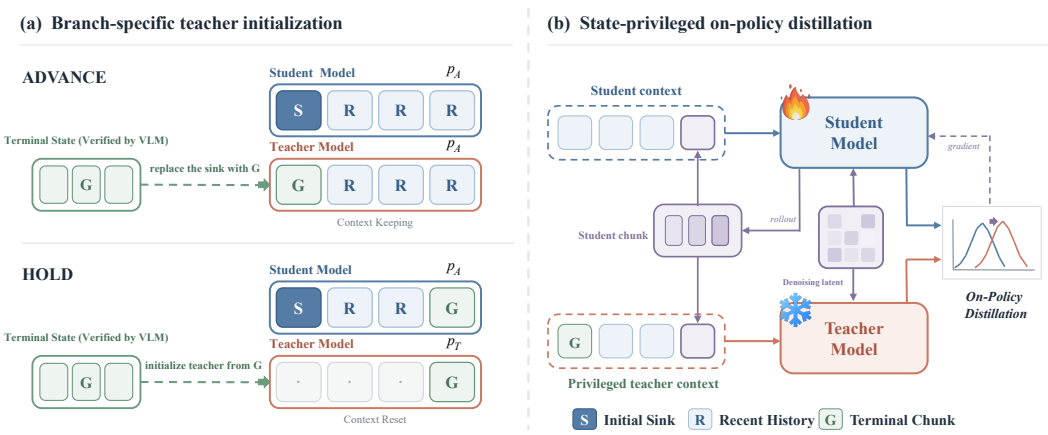


Figure 3: **Overview of State Forcing.** ADVANCE replaces the teacher's initial sink with a verified terminal chunk while retaining incomplete history and the original instruction. HOLD initializes the teacher from the terminal chunk alone under a consequent-state description. The student uses standard streaming inputs. Both predict on student noisy states and cache student-generated chunks.

$$v^S_{j,\tau} = v_\theta(z^S_{j,\tau}, \tau \mid C^S_j, p_A), \qquad v^T_{j,\tau} = v_\phi(z^S_{j,\tau}, \tau \mid C^T_j, p_b). \tag{2}$$

Thus, G enters the teacher's context rather than serving as a reconstruction target.

To keep subsequent supervision on the student's trajectory, we update both histories with the clean student chunk $\widehat{X}^S_j$. Let $S_j = \mathrm{sg}(\widehat{X}^S_j)$, where sg denotes stop-gradient. The cache updates are

$$C^S_{j+1} = \mathcal{T}_\theta(C^S_j, S_j; p_A), \qquad C^T_{j+1} = \mathcal{T}_\phi(C^T_j, S_j; p_b), \tag{3}$$

where $\mathcal{T}_\psi$ is the base cache update, including encoding with model ψ and its standard eviction or recaching policy. The teacher therefore uses student-generated history, not a separate teacher rollout. Denoising transitions and history-cache construction are stop-gradient.

Following on-policy denoising prediction matching (Liu et al., 2026a), we minimize

$$\mathcal{L}_{\mathrm{StF}}(\theta) = \mathbb{E}_{d\sim\mathcal{D}_{\mathrm{StF}},\,\xi}\left[\frac{1}{Z}\sum_{j=0}^{J-1}\sum_{\tau\in\mathcal{U}_j} w_{j,\tau}\left\|v^S_{j,\tau} - \mathrm{sg}(v^T_{j,\tau})\right\|_2^2\right], \tag{4}$$

where ξ denotes sampling randomness and $w_{j,\tau} \geq 0$ weights the supervised states, with $Z = \sum_{j=0}^{J-1}\sum_{\tau\in\mathcal{U}_j} w_{j,\tau} > 0$. We uniformly sample one denoising step per rollout and supervise that step across all J generated chunks, using unit weights ($w_{j,\tau} = 1$, $Z = J$). Only θ is updated; ϕ remains frozen. At the start of training, prediction differences arise from the distinct context despite identical model parameters. Algorithm 1 details the procedure; Section 5.1 gives training settings.

Inference. Only the trained student is used, with the original instruction, architecture, sampler, and cache policy. No terminal-state inputs, prompt rewriting, or additional computation are required.

4 TELICBENCH: TELIC INSTRUCTION EVALUATION

TELICBENCH evaluates whether streaming generators complete a requested state transition without regressing, and preserve its consequence through the remainder of the rollout.

4.1 BENCHMARK AND EVALUATION PROTOCOL

Following the evaluation setup of TC-Bench-T2V (Feng et al., 2024), we construct TELICBENCH with 150 telic instructions. All are held out from our training stages, with no training-prompt

Table 1: **Telic instruction following.** Evaluation uses 150 instructions with five seeds each over 30-second rollouts. Baselines use their documented configurations. CR, TSR, and TVR are percentages; higher CR/TSR and lower TVR are better.

Method	#Params	Resolution	Throughput (FPS) ↑	Core Metrics		
				CR ↑	TVR ↓	TSR ↑
Diffusion Forcing Causal						
SkyReels-V2 (Chen et al., 2025)	1.3B	960 × 540	0.49	28.67	30.46	6.67
MAGI-1 (Teng et al., 2025)	4.5B	832 × 480	0.19	12.93	9.12	9.47
Distilled Causal						
Self Forcing, chunk-wise (Huang et al., 2026)	1.3B	832 × 480	17.0	37.07	24.39	13.33
Rolling Forcing (Liu et al., 2026b)	1.3B	832 × 480	15.8	40.13	19.86	15.47
Context Forcing (Chen et al., 2026)	1.3B	832 × 480	17.0	41.47	15.58	24.93
LongLive (Yang et al., 2025a)	1.3B	832 × 480	20.7	39.33	21.75	16.67
STATE FORCING	1.3B	832 × 480	20.7	**51.33**	**7.60**	**37.33**

Table 2: **General video generation evaluation.** Each method generates five videos per prompt for 100 shared prompts at each duration. The 30- and 60-second videos are generated separately. Bold and underlined values indicate the best and second-best scores within each duration and metric.

Method	Subject Consistency ↑	Background Consistency ↑	Motion Smoothness ↑	Dynamic Degree ↑	Aesthetic Quality ↑	Imaging Quality ↑
30-second generation						
Self Forcing, chunk-wise	0.9694	0.9574	0.9841	**0.6107**	0.5794	67.81
Rolling Forcing	<u>0.9766</u>	0.9617	0.9863	0.4593	0.5863	**72.16**
Context Forcing	0.9729	<u>0.9632</u>	0.9845	<u>0.5800</u>	**0.6053**	69.45
LongLive	0.9751	0.9586	<u>0.9871</u>	0.4447	<u>0.5977</u>	70.10
STATE FORCING	**0.9860**	**0.9674**	**0.9916**	0.4587	0.5947	<u>70.19</u>
60-second generation						
Self Forcing, chunk-wise	0.9678	0.9540	0.9832	**0.5770**	0.5651	66.16
Rolling Forcing	0.9753	0.9600	0.9860	0.4100	0.5854	**71.42**
Context Forcing	<u>0.9758</u>	**0.9660**	0.9855	<u>0.5603</u>	**0.6030**	69.35
LongLive	0.9734	0.9597	<u>0.9866</u>	0.4643	<u>0.5969</u>	<u>69.86</u>
STATE FORCING	**0.9840**	<u>0.9640</u>	**0.9906**	0.4450	0.5798	68.74

duplicates found during manual review. Each instruction specifies an observable initial condition and an identifiable culmination, with a rubric defining the requested transition, terminal-state conditions, and consequence retention. Generators receive the instruction, while the evaluator receives the rubric. Each rollout follows one fixed instruction, with scoring continuing through the end.

For each instruction, we generate five videos per method using the same five random seeds. A frozen VLM (OpenAI, 2026) reviews the instruction, rubric, and chronological frames sampled at 2 fps, including the final available frame. It returns a completion verdict and labeled temporal intervals. A deterministic validator checks frame references, full temporal coverage without overlaps, and consistency among output fields. All methods use the same evaluation protocol, with method identity and training-time terminal inputs withheld. Appendix B details benchmark construction and scoring; Section 5.3 reports human evaluation experiments.

4.2 EVALUATION METRICS: STRICT SUCCESS AND DIAGNOSTICS

Telic Success Rate (TSR) measures overall execution success: completion with no scored violations over the observed rollout. We report two complementary diagnostics. *Completion Rate (CR)* measures the fraction of videos that complete the transition, and *Telic Violation Rate (TVR)* quantifies the fraction of video duration scored as violations. A brief regression and persistent cycling both fail strict success, but TVR distinguishes how much of the rollout is affected.

Completion Rate (CR). Completion requires the same primary subject to visibly transition from the specified initial condition to a state satisfying all terminal-state conditions simultaneously. A terminal

state present from the start does not count, whereas later regression does not revoke completion. Let $\bar{c}_i = 1$ for verified completion and 0 otherwise. For N evaluated videos,

$$\mathrm{CR} = \frac{1}{N}\sum_{i=1}^{N} \bar{c}_i. \tag{5}$$

Telic Violation Rate (TVR). We annotate three non-exclusive failure modes: *progress regression* reverses achieved progress, including loss of an established consequence; *reset* abruptly returns execution to an earlier state; and *re-execution* inappropriately restarts or repeats a one-off transition, including through a duplicate instance. For resets and regressions, violation intervals extend until the lost task-relevant progress is visibly restored. For re-execution, we count the ongoing execution, without penalizing subsequent stable frames solely because re-execution occurred. Pauses, slow progress, and non-execution alone are not violations.

Video i is partitioned into one-second bins $k \in \mathcal{B}_i$, with durations Δ_{ik} and total duration $T_i = \sum_{k\in\mathcal{B}_i} \Delta_{ik}$. Let $\bar{v}_{ik} = 0$ for bins verified to satisfy the rubric's progression and retention requirements, and $\bar{v}_{ik} = 1$ otherwise. The full label mapping is specified in Appendix B. TVR aggregates these interval judgments by duration:

$$\mathrm{TVR}_i = \frac{1}{T_i}\sum_{k\in\mathcal{B}_i} \Delta_{ik}\bar{v}_{ik}, \qquad \mathrm{TVR} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{TVR}_i. \tag{6}$$

Telic Success Rate (TSR). Strict success requires verified completion and every bin scored clean:

$$s_i = \bar{c}_i\, \mathbf{1}[\forall k \in \mathcal{B}_i,\ \bar{v}_{ik} = 0], \qquad \mathrm{TSR} = \frac{1}{N}\sum_{i=1}^{N} s_i. \tag{7}$$

5 Experiments

5.1 Experimental Setup

Model and training. We implement State Forcing with a causal generator based on Wan2.1-T2V-1.3B (Wan et al., 2025) at 832×480 resolution. We construct a pool of 23,330 telic instructions: 20,330 selected from VidProM (Wang & Yang, 2024) and 3,000 generated by GPT-5.6, which also produces a consequent-state description for each instruction. We first combine these instruction–consequence pairs with atelic prompts for LongLive's long-video tuning (Yang et al., 2025a), yielding the *Warm-up Only* checkpoint evaluated in Table 4. We then use this checkpoint to generate source rollouts from the same prompt pool, collecting 3,000 training records verified by GPT-5.6. State Forcing trains from this checkpoint for 200 iterations with a batch size of 24 and a learning rate of 10^{-5}, sampling Advance and Hold with equal probability. Each training rollout generates a 30-second continuation beyond its conditioning prefix. Appendix A provides more details.

Evaluation settings. We evaluate the six baselines in Table 1 using official implementations and the 30-second TelicBench protocol in Section 4. Following Context Forcing, we randomly sample 100 MovieGenBench prompts for general video quality evaluation. Each method generates five videos per prompt at 30 and 60 seconds, separately. We compare with the distilled causal baselines on VBench scores (Huang et al., 2024), using shared prompts, seeds, and output specifications.

5.2 Main Results

Telic instruction following. We first assess whether State Forcing improves telic instruction execution. It achieves the highest TSR (37.33%) in Table 1, exceeding Context Forcing (24.93%) by 12.40 percentage points. It also achieves the highest CR (51.33%) and lowest TVR (7.60%), compared with Context Forcing's 41.47% and 15.58%, respectively. This joint improvement matters because low violation occupancy alone can coexist with limited execution, as illustrated by MAGI-1's 9.12% TVR and 12.93% CR. State Forcing achieves higher completion rates with lower violation occupancy, demonstrating its ability to more reliably complete requested state transitions and preserve their consequences during streaming generation.

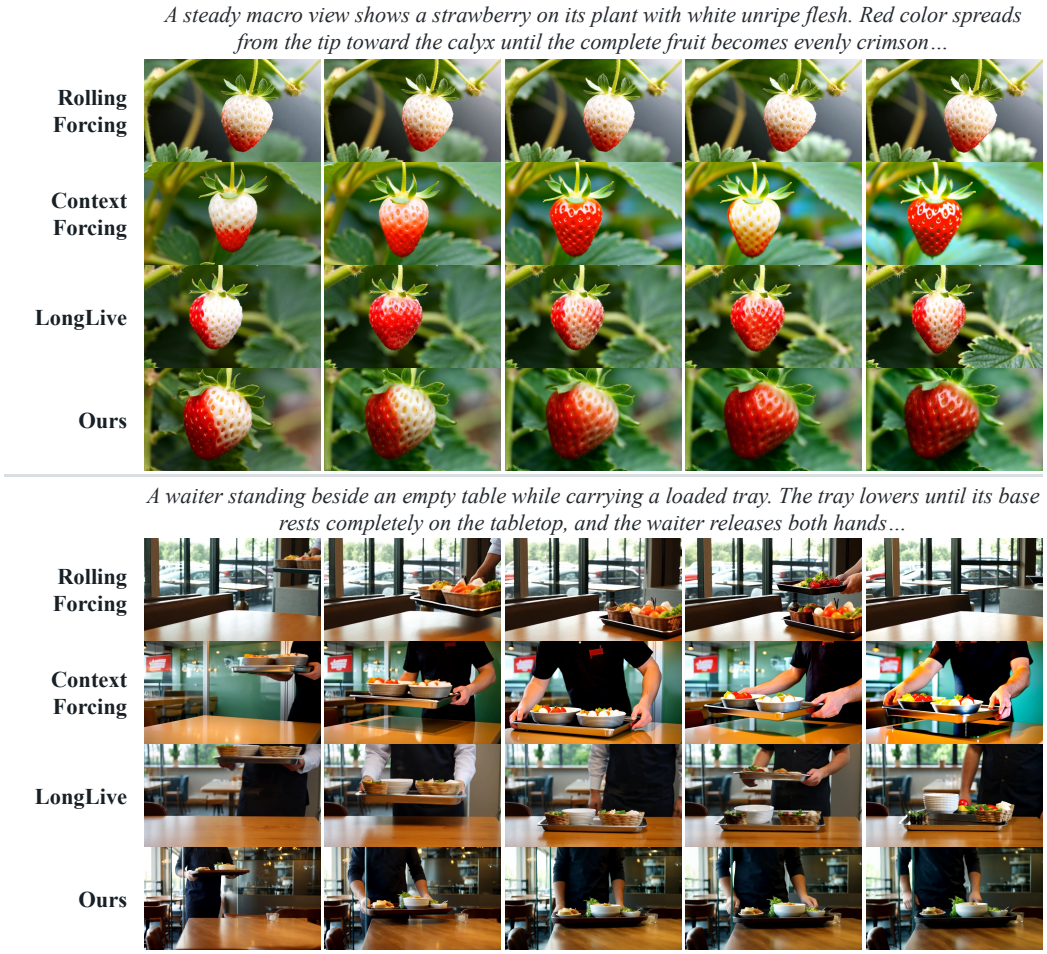


Figure 4: **Qualitative comparisons.** Chronological frames show strawberry ripening (top) and tray placement (bottom). Baselines exhibit incomplete or reversed ripening, and regression or re-execution during tray placement. STATE FORCING completes both transitions and preserves their consequences.

Qualitative comparisons. Figure 4 compares how different methods progress toward completion and retain its consequences. For strawberry ripening, Rolling Forcing leaves the fruit partially unripe, while Context Forcing and LongLive show a return from redder to paler states. With STATE FORCING, the fruit ripens and remains red. For tray placement, Context Forcing returns to holding the tray, LongLive begins another tray-placement sequence with a second tray, and Rolling Forcing reverses the placement by taking the tray away from the table. In contrast, STATE FORCING places the tray on the table, where it remains after the waiter releases it. These examples highlight STATE FORCING's ability to sustain progress toward completion and preserve the consequences.

General video quality. On MovieGenBench prompts (Table 2), STATE FORCING leads in subject consistency and motion smoothness at both settings, and ranks first and second in background consistency at 30 and 60 seconds. These consistency gains coexist with dynamic-degree scores of 0.4587 and 0.4450, close to LongLive's 0.4447 and 0.4643, respectively. These results indicate improved consistency and retained scene dynamics with comparable overall video quality.

5.3 HUMAN EVALUATION

We conduct a blinded human study to assess instruction execution and its automatic evaluation. We randomly sample 50 TELICBENCH prompts and select one random seed per prompt, using the same prompt–seed pairs across methods to obtain 350 30-second videos. We recruit 21 participants, with

each video independently evaluated by three participants. Each assigned rater provides a binary completion judgment and rates violation extent on a five-point ordinal scale (1: none; 5: dominant). Method identities and automatic scores are hidden from all raters. Human completion is determined by majority vote, and violation extent is the median rating. Strict success requires at least two raters to report both completion and a violation rating of 1.

In the human-only comparison, STATE FORCING achieves 34% strict success, compared with 20% for Context Forcing, together with the highest completion rate and lowest mean violation extent on this subset. These human judgments corroborate improvements in both completion and violation-free execution independently of the automatic evaluator. More details are provided in Appendix C.

Against human consensus, automatic judgments achieve macro-F1 of 0.7926 for completion and 0.6470 for strict success (Table 3). Automatic TVR correlates with median human violation-extent ratings at Spearman's $\rho = 0.5664$, indicating that videos with higher TVR tend to receive higher human violation ratings. Human–human statistics provide a reference for inter-rater agreement.

Table 3: **Automatic and human judgments.** F1 is macro-F1; ρ is Spearman correlation. Auto–consensus uses aggregated human judgments; Human–human pools within-video rater pairs.

Metric	Statistic	Auto–consensus	Human–human
CR	F1 ↑	0.7926	0.8808
TSR	F1 ↑	0.6470	0.8926
Violation Extent	ρ ↑	0.5664	0.7422

Table 4: **Component ablations on TELICBENCH.** Distillation variants share the initialization and training budget. Evaluation follows Table 1; scores are percentages.

Variant	CR ↑	TVR ↓	TSR ↑
Warm-up Only	39.87	20.02	19.20
w/o HOLD	47.73	12.66	21.47
w/o ADVANCE	34.67	8.24	25.33
Separate Teacher Rollout	25.33	10.29	18.27
Full STATE FORCING	51.33	7.60	37.33

5.4 COMPONENT ABLATIONS

Table 4 summarizes our designed component ablations. Telic warm-up modestly improves CR and TSR over the original LongLive; STATE FORCING further increases TSR from 19.20% to 37.33%. All distillation variants share the warm-up initialization and training budget.

Complementary supervision. We ablate the ADVANCE and HOLD branches separately. Without HOLD, TSR drops to 21.47% and TVR rises to 12.66%, despite a CR of 47.73%. This indicates that ADVANCE alone can promote completion but is insufficient to maintain violation-free execution. Without ADVANCE, TVR remains low at 8.24%, but CR drops to 34.67%. The low TVR comes with reduced completion, highlighting the role of ADVANCE supervision in promoting completion. Joint training performs best across all three metrics, supporting the branches' complementary roles.

Student-generated history. In *Separate Teacher Rollout*, the teacher retains privileged context but updates its cache with its own generated chunks. This reduces TSR from 37.33% to 18.27% and CR from 51.33% to 25.33%. As the rollout grows longer, the teacher and student histories diverge substantially, making teacher predictions increasingly misaligned with the student's current state. These results support the importance of aligning teacher supervision with student-generated history.

6 CONCLUSIONS

We have presented STATE FORCING, a state-privileged on-policy self-distillation framework for telic instruction following in streaming video generation. Our empirical study shows that verified terminal states can guide both non-regressing progress and consequence retention in a frozen generator. ADVANCE and HOLD turn this finding into complementary teacher supervision along student-generated trajectories, without additional inference inputs or computation. We also introduced TELICBENCH to assess strict success, with completion and violation duration as complementary diagnostics. Automatic and human evaluations show improved completion and strict success, alongside general video quality comparable to distilled streaming baselines. These findings highlight verified outcomes as supervisory context for event progression and consequence retention.

AI USE STATEMENT

We used GPT-5.6 to generate telic prompts and consequent-state descriptions, verify training videos, and perform automatic evaluation. Claude Opus 4.8 assisted in reviewing candidate prompts for unambiguous telic semantics. We also used GPT-6 to assist with language polishing. The authors reviewed the AI-assisted content and take responsibility for the final manuscript, claims, and reported results.

ETHICS STATEMENT

STATE FORCING improves instruction following in streaming video generation, with potential benefits for creative and educational applications. As with other video generators, its outputs may be misused to create deceptive or harmful content and may reflect biases inherited from pretrained models. We encourage content safeguards and clear labeling of generated media in downstream applications. Our experiments use existing prompt resources, language-model-generated instructions, and model-generated videos. Our blinded human study involved 21 participants assessing instruction execution in generated videos (Appendix C); this evaluation does not establish content safety. All authors have reviewed the ICLR 2027 Code of Ethics and conducted this research in accordance with its principles.

REPRODUCIBILITY STATEMENT

Section 3 describes State Forcing, and Section 5.1 reports the experimental setup. Appendix A details training-data construction, cache initialization, and the distillation procedure. The benchmark and scoring protocol are described in Section 4 and Appendix B. Appendix C describes human annotation, aggregation, and statistical analysis. We will release our code to facilitate reproducibility and support future research on telic instruction following.

REFERENCES

Anthropic. Claude Opus 4.8 system card, May 2026. URL `https://anthropic.com/claude-opus-4-8-system-card`.

Chirui Chang, Jiahui Liu, Zhengzhe Liu, Xiaoyang Lyu, Yi-Hua Huang, Xin Tao, Pengfei Wan, Di Zhang, and Xiaojuan Qi. How far are ai-generated videos from simulating the 3d visual world: A learned 3d evaluation approach. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10307–10317. IEEE, 2025.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

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.

Shuo Chen, Cong Wei, Sun Sun, Ping Nie, Kai Zhou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. *arXiv preprint arXiv:2602.06028*, 2026.

Bernard Comrie. *Aspect: An introduction to the study of verbal aspect and related problems*, volume 2. Cambridge university press, 1976.

Jiaxing 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 *International Conference on Learning Representations*, volume 2026, pp. 85802–85822, 2026a.

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. *arXiv preprint arXiv:2601.16914*, 2026b.

Weixi Feng, Jiachen Li, Michael Saxon, Tsu-jui Fu, Wenhu Chen, and William Yang Wang. Tc-bench: Benchmarking temporal compositionality in text-to-video and image-to-video generation. *arXiv preprint arXiv:2406.08656*, 2024.

Yuchao Gu, Guian Fang, Yuxin Jiang, Weijia Mao, Song Han, Han Cai, and Mike Zheng Shou. Anyflow: Any-step video diffusion model with on-policy flow map distillation. In *European Conference on Computer Vision*, pp. 163–181. Springer, 2026.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2568–2577. IEEE, 2025.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Qixin Hu, Shuai Yang, Wei Huang, Song Han, and Yukang Chen. Longlive-rag: A general retrieval-augmented framework for long video generation. *arXiv preprint arXiv:2606.02553*, 2026.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Sihui Ji, Xi Chen, Shuai Yang, Xin Tao, Pengfei Wan, and Hengshuang Zhao. Memflow: Flowing adaptive memory for consistent and efficient long video narratives. *arXiv preprint arXiv:2512.14699*, 2025.

Dengyang Jiang, Xin Jin, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Ruoyi Du, Xiangpeng Yang, Qilong Wu, Zhen Li, Peng Gao, et al. D-opsd: On-policy self-distillation for continuously tuning step-distilled diffusion models. *arXiv preprint arXiv:2605.05204*, 2026.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. *Advances in Neural Information Processing Systems*, 37: 89834–89868, 2024.

Akio Kodaira, Tingbo Hou, Ji Hou, Markos Georgopoulos, Felix Juefei-Xu, Masayoshi Tomizuka, and Yue Zhao. Streamdit: Real-time streaming text-to-video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29200–29210, 2026.

Jia Li, Xiaomeng Fu, Xurui Peng, Weifeng Chen, Youwei Zheng, Tianyu Zhao, Jiexi Wang, Fangmin Chen, Xing Wang, and Hayden Kwok-Hay So. Train short, inference long: Training-free horizon extension for autoregressive video generation. *arXiv preprint arXiv:2602.14027*, 2026.

Hongyu Liu, Chun Wang, Feng Gao, Xuanhua He, Yue Ma, Ziyu Wan, Yong Zhang, Xiaoming Wei, and Qifeng Chen. Opsd-v: On-policy self-distillation for post-training few-step autoregressive video generators. *arXiv preprint arXiv:2607.08766*, 2026a.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *International Conference on Learning Representations*, volume 2026, pp. 91177–91196, 2026b.

Ling Lo, Kelvin CK Chan, Wen-Huang Cheng, and Ming-Hsuan Yang. From prompt to progression: Taming video diffusion models for seamless attribute transition. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025.

OpenAI. GPT-5.6 system card. `https://deploymentsafety.openai.com/gpt-5-6`, July 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.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8406–8416. IEEE, 2025.

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

Zeno Vendler. Verbs and times. *The philosophical review*, 66(2):143–160, 1957.

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.

Jing Wang, Xiangxin Zhou, Jiajun Liang, Kaiqi Liu, Wanyuan Pang, Zhenyu Xie, Tianyu Pang, and Xiaodan Liang. Tempact: Advancing temporal plausibility in autoregressive video generation via planner-executor rl. *arXiv preprint arXiv:2606.28016*, 2026.

Wenhao Wang and Yi Yang. Vidprom: A million-scale real prompt-gallery dataset for text-to-video diffusion models. *Advances in Neural Information Processing Systems*, 37:65618–65642, 2024.

Boxun Xu, Yuming Du, Zichang Liu, Siyu Yang, Ziyang Jiang, Siqi Yan, Rajasi Saha, Albert Pumarola, Wenchen Wang, and Peng Li. Sparse forcing: Native trainable sparse attention for real-time autoregressive diffusion video generation. *arXiv preprint arXiv:2604.21221*, 2026.

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*, 2025a.

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 *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025b.

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. *arXiv preprint arXiv:2512.05081*, 2025.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974. IEEE, 2025.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. *Advances in Neural Information Processing Systems*, 38:30546–30566, 2026.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026a.

Zengqun Zhao, Yanzuo Lu, Ziquan Liu, Jifei Song, Jiankang Deng, and Ioannis Patras. Relax forcing: Relaxed kv-memory for consistent long video generation. *arXiv preprint arXiv:2603.21366*, 2026b.

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. *arXiv preprint arXiv:2602.02214*, 2026.

A Additional Method and Implementation Details

A.1 Training Data and State Verification

The prompt pool described in Section 5.1 is used for both warm-up training and offline record construction. GPT-5.6 Sol (OpenAI, 2026) generates a consequent-state description for each telic instruction. Before source-video generation, Claude Opus 4.8 (Anthropic, 2026) reviews candidate prompts for well-defined, unambiguous telic semantics.

The warm-up checkpoint generates source rollouts under the original telic instructions. GPT-5.6 verifies the specified initial condition and visible execution, then identifies an interval in which the requested consequence holds. Acceptance requires an observed transition from the initial state; videos starting at the endpoint are rejected. Prefixes and terminal chunks must preserve subject and scene compatibility, excluding severe drift. Generation and verification terminate upon collecting 3,000 accepted records, before exhausting the 23,330-prompt pool. These records initialize on-policy training rollouts, whose continuation chunks are generated by the current student.

A.2 Terminal-Chunk Selection and Branch Admissibility

Using the record and half-open slice notation in Section 3.1, let $\mathcal{I}_{\mathrm{term}}$ denote the verified terminal interval in latent time indices. An admissible boundary satisfies

$$P \in m\mathbb{N}, \qquad h \le P \le T, \qquad [P-m, P) \subseteq \mathcal{I}_{\mathrm{term}}. \tag{8}$$

The terminal chunk G must lie entirely within the verified interval and forms the final chunk of the recent history R.

For ADVANCE, the prefix must show correct, incomplete execution compatible with the subsequent terminal chunk G. Prefixes exhibiting static non-execution, progress regression, cycling, or shortcut transitions are excluded. For HOLD, the prefix ends at P, and the original sink and R must preserve subject identity and scene continuity. The student retains G as the final complete chunk of R. The branch stays fixed throughout the rollout, and context interventions are applied only at initialization.

A.3 Training Configuration

Both networks are initialized from the *Warm-up Only* checkpoint. Warm-up follows LongLive's long-video tuning procedure, mixing instruction–consequence pairs with atelic prompts in prompt-switch training over 30-second rollouts. For ADVANCE, we use the native-length prefix as the default and randomly truncate its end during training for augmentation. Table 5 summarizes the subsequent State Forcing configuration. Each record conditions a continuation of 40 new chunks, independent of the source boundary used for initialization; this budget excludes the conditioning prefix.

Table 5: State Forcing training configuration. Lengths are measured in latent time steps unless otherwise specified.

Setting	Value
Backbone / output resolution	Wan2.1-T2V-1.3B / 832×480
Student / teacher initialization	Warm-up Only / frozen copy
Accepted training records	3,000
Iterations / batch size / learning rate	200 / 24 / 10^{-5}
Branch sampling	$\Pr(\textsc{Advance}) = \Pr(\textsc{Hold})$
Chunk length / persistent sink length	$m = 3$ / 3
Recent-history length	$h = 9$
Native prefix length	$L_{\mathrm{nat}} = 21$
Newly generated chunks	$J = 40$
New continuation length	$Jm = 120$ latents

A.4 Cache Initialization Details

We represent a positioned history as $\mathsf{H} = [(Y_r, \pi_r)]_{r=1}^{M}$, where Y_r is a latent subsequence and π_r specifies its temporal positions. The operator $\mathcal{E}_\psi(\mathsf{H}; p)$ encodes this history with model ψ, text

condition p, and the base cache policy. The overwrite operator $\mathcal{O}_{\mathcal{I}}(C, E)$ replaces only the entries indexed by $\mathcal{I}$ in C with aligned entries from E. The cache transition $\mathcal{T}_\psi$ follows Section 3.2.

ADVANCE. Both models prefill the selected prefix $X_{0:L}$ at its original positions. Define

$$\mathsf{H}_{\mathrm{A}} = \big[(X_{0:L}, [0, L))\big], \qquad \mathcal{I}_{\mathrm{sink}} = [0, m). \tag{9}$$

Writing $\bar{C}_0^T = \mathcal{E}_\phi(\mathsf{H}_{\mathrm{A}}; p_A)$, we initialize

$$\begin{aligned} C_0^S &= \mathcal{E}_\theta(\mathsf{H}_{\mathrm{A}}; p_A), \\ C_0^T &= \mathcal{O}_{\mathcal{I}_{\mathrm{sink}}}\big(\bar{C}_0^T, \mathcal{E}_\phi\big([(G, \mathcal{I}_{\mathrm{sink}})]; p_A\big)\big). \end{aligned} \tag{10}$$

The temporal positions of G are reassigned to the sink interval $[0, m)$ for encoding. The replacement preserves the teacher's existing recent-history entries. Both models begin generation at cursor L under p_A.

HOLD. The positioned histories are

$$\begin{aligned} \mathsf{H}_{\mathrm{H}}^S &= \big[(X_{0:m}, [0, m)),\ (R, [P - h, P))\big], \\ \mathsf{H}_{\mathrm{H}}^T &= \big[(G, [P - m, P))\big]. \end{aligned} \tag{11}$$

The initial caches are

$$C_0^S = \mathcal{E}_\theta(\mathsf{H}_{\mathrm{H}}^S; p_A), \qquad C_0^T = \mathcal{E}_\phi(\mathsf{H}_{\mathrm{H}}^T; p_T). \tag{12}$$

Both models begin generation at cursor P. The teacher is initialized solely from G at $[P - m, P)$, whose entries form its persistent sink, while preserving the temporal cursor P. The student observes the same terminal chunk through R and retains its original sink and instruction. Following either branch's initialization, both histories accumulate student-generated chunks under the standard eviction or recaching policy, without further context resets.

A.5 SUPERVISED TIMESTEPS AND GRADIENT FLOW

Let $\mathcal{S}$ be the denoising-step set of the inherited sampler. A timestep $\tau_\star$ is sampled uniformly once per rollout and reused for all newly generated chunks:

$$\tau_\star \sim \mathrm{Uniform}(\mathcal{S}), \qquad \mathcal{U}_j = \{\tau_\star\}, \qquad w_{j,\tau_\star} = 1, \qquad Z = J. \tag{13}$$

Gradients are computed only for the student's denoising prediction at each selected noisy state. Sampler transitions, teacher predictions, and history construction are detached, excluding gradient propagation through earlier sampling steps and cached chunks. Both histories incorporate the same detached student-generated *latent chunk*. Each model constructs its own KV representations using its parameters and branch-specific text condition, as in Eq. 3.

A.6 DISTILLATION ALGORITHM

Algorithm 1 computes the distillation loss for one on-policy rollout. Rollout losses are averaged over the training batch to update θ, while ϕ remains frozen. Gradients are restricted to supervised student predictions; sampling and cache operations are performed without gradient tracking. The terminal chunk conditions the teacher, and supervision matches its denoising predictions on student-generated trajectories.

Training and inference. ADVANCE and HOLD provide complementary supervision for progression and consequence retention. ADVANCE uses terminal evidence to guide continuation from an incomplete transition. In HOLD, the terminal chunk is already present in the student's recent history, and supervision promotes consequence retention under the original instruction. Inference uses only the original instruction and standard streaming history, without privileged terminal evidence, consequent-state descriptions, or online completion detection.

Algorithm 1 State Forcing: loss for one on-policy rollout

Require: Admissible record $d = (X, p_A, p_T, P, b)$; student θ; frozen teacher ϕ
Require: Sampler denoising steps $\mathcal{S}$; continuation length $J = 40$
$G \leftarrow X_{P-m:P}$; $R \leftarrow X_{P-h:P}$
if $b =$ ADVANCE **then**
    Initialize C_0^S, C_0^T by Eq. 10; $p_b \leftarrow p_A$
else
    Initialize C_0^S, C_0^T by Eq. 12; $p_b \leftarrow p_T$
Sample $\tau_\star \sim \mathrm{Uniform}(\mathcal{S})$ once; $\ell \leftarrow 0$
for $j = 0, \ldots, J-1$ **do**
    Run the student sampler from C_j^S, p_A with detached transitions
    Retain the clean output $\widehat{X}_j^S$ and visited state $z_{j,\tau_\star}^S$
    $z \leftarrow \mathrm{sg}(z_{j,\tau_\star}^S)$
    $v^S \leftarrow v_\theta(z, \tau_\star \mid C_j^S, p_A)$
    $v^T \leftarrow \mathrm{sg}\big(v_\phi(z, \tau_\star \mid C_j^T, p_b)\big)$
    $\ell \leftarrow \ell + \|v^S - v^T\|_2^2 / J$
    $S_j \leftarrow \mathrm{sg}(\widehat{X}_j^S)$
    $C_{j+1}^S \leftarrow \mathcal{T}_\theta(C_j^S, S_j; p_A)$
    $C_{j+1}^T \leftarrow \mathcal{T}_\phi(C_j^T, S_j; p_b)$
return ℓ

B TELICBENCH EVALUATION DETAILS

B.1 BENCHMARK CONSTRUCTION AND RUBRICS

GPT-5.6 Sol (OpenAI, 2026) generates candidate instructions, which are reviewed by Claude Opus 4.8 (Anthropic, 2026) and manually checked by the authors for clear telic semantics. Each instruction specifies an observable transition from an initial condition to a terminal state that can persist while unrelated aspects of the scene continue to evolve. We use model-assisted and manual review to exclude candidate instructions that overlap with the training prompt pool. The resulting TELICBENCH contains 150 instructions held out from both warm-up training and record construction. Five shared generation seeds yield 750 videos per method.

Rubrics are constructed from the instructions using four fields. The `initial` field records the initial condition. The `completion` field preserves the instruction's transition and endpoint requirements, prefixed by a shared rule requiring the same primary subject to visibly execute the transition and reach all stated endpoint conditions. The `retention` field specifies the consequence that should persist after completion. The `violations` field applies shared rules for progress regression, reset, inappropriate re-execution, and invalidation of the achieved endpoint. Normal pauses and required multi-step actions are not violations. Generators receive only the instruction, while the evaluator additionally receives its rubric.

B.2 VISUAL EVIDENCE AND STRUCTURED JUDGMENTS

We use the frozen `gpt-5.6-sol` evaluator. It receives the instruction, rubric, and chronological RGB frames sampled at 2 fps over a shared 30-second horizon, including the first and final available frames. Frames retain their timestamps, are resized to 360 pixels wide with aspect ratio preserved, and are arranged in contact sheets of at most 16 frames each.

The evaluator reviews all sheets to track task progress, subject identity, and instance counts. It returns a completion judgment c_i, the first observed completion timestamp, and non-overlapping segments covering all one-second bins. Each segment assigns a common label ℓ_{ik} to its bins and cites supporting timestamps and evidence, with failure types for violations.

All methods share the evaluator configuration, prompt template, response schema, and frame-sampling procedure. Video cases are presented in randomized order; method identity and training-time terminal evidence are withheld. A deterministic validator checks frame references, complete temporal coverage, and cross-field consistency.

Example instruction and rubric: putting on sunglasses

Instruction. A fixed daylight close-up presents a woman holding sunglasses below her face. She raises the frames, slides both temples over her ears, and releases them once the lenses sit squarely before her eyes. Reflections move across the lenses while the worn position remains secure.

Rubric field	Content
`initial`	A fixed daylight close-up presents a woman holding sunglasses below her face.
`completion`	The same primary subject must visibly undergo the requested transition and reach every endpoint condition stated here: She raises the frames, slides both temples over her ears, and releases them once the lenses sit squarely before her eyes.
`retention`	the worn position remains secure.
`violations`	Any clearly achieved task progress that is undone; return to an earlier task state; inappropriate restart/repetition of the intended one-off transition; or visible invalidation of the achieved endpoint. Normal pauses and required multi-step action are not violations.

Moving reflections are compatible with retention of the worn state. The rubric imposes no prescribed completion time.

B.3 Interval Attribution and Score Mapping

The three failure modes follow Section 4.2. Regression and reset intervals persist until lost progress or consequences are visibly restored. Re-execution, including through an additional object, is counted while the inappropriate repetition continues; subsequent frames are judged by their current state. Concurrent regressions or resets retain their own intervals, and overlapping failures are counted only once per bin.

Completion requires the same subject to visibly transition from the initial condition to a state satisfying all endpoint requirements. Subsequent violations do not revoke completion credit. All metrics are computed over the full 30-second rollout. Pauses, slow progress, non-execution, and required sub-actions alone incur no violation penalty.

Let $c_i = 1$ denote verified completion and $\ell_{ik} = \text{clean}$ a bin verified to satisfy the rubric's progression and retention requirements. The inputs to Eqs. 5–7 are

$$\bar{c}_i = \mathbf{1}[c_i = 1], \qquad \bar{v}_{ik} = \mathbf{1}[\ell_{ik} \neq \text{clean}]. \tag{14}$$

CR credits completion, TVR measures the fraction of time scored non-clean, and TSR requires completion with every bin clean. All metrics include incomplete rollouts; non-execution can yield zero TVR without completion.

C Human Evaluation Protocol and Results

C.1 Study Design and Annotation

The paired subset in Section 5.3 is fixed before annotation: 50 randomly selected prompts and one random seed per prompt, shared across seven methods. The 350 unedited 30-second videos each receive three independent ratings from a pool of 21 participants, totaling 1,050 annotations. Videos are anonymized and presented in random order, with method identities, automatic scores, training conditions, and expected rankings withheld.

Each trial presents the instruction and full video (Figure 5). Raters judge completion as binary and violation extent on a five-point scale: 1 (none), 2 (isolated), 3 (limited), 4 (substantial), and 5 (dominant). Annotators receive shared guidelines specifying the completion and violation criteria in Appendix B.3 and the five rating levels. Extent is assessed holistically over the video, without frame-level annotation or duration estimation.

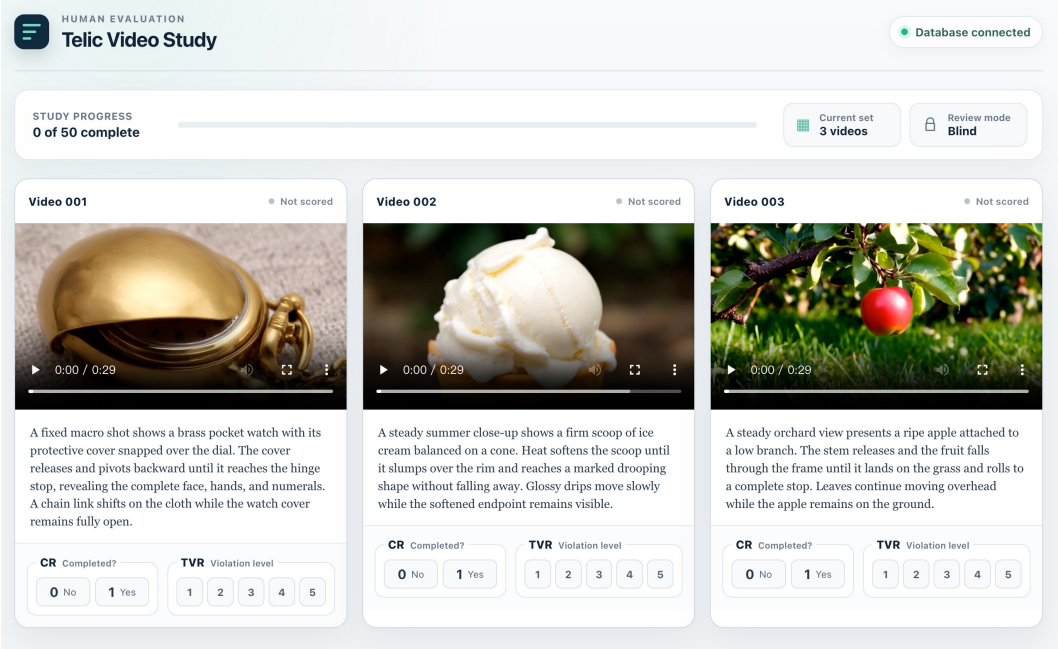


Figure 5: **Human annotation interface.** The interface labels the five-point ordinal violation-extent rating as TVR; this is distinct from automatic duration-based TVR.

C.2 Consensus and Human Metrics

For video i, annotator r provides completion $c_{ir} \in \{0, 1\}$ and extent $e_{ir} \in \{1, \ldots, 5\}$, giving strict success $a_{ir} = c_{ir}\mathbf{1}[e_{ir} = 1]$. Consensus is

$$c_i^{\mathrm{H}} = \mathbf{1}\left[\sum_{r=1}^{3} c_{ir} \geq 2\right], \quad s_i^{\mathrm{H}} = \mathbf{1}\left[\sum_{r=1}^{3} a_{ir} \geq 2\right], \quad e_i^{\mathrm{H}} = \mathrm{median}_{r=1,2,3}\, e_{ir}. \tag{15}$$

Strict success requires at least two raters to each report both completion and no violation. All videos, including 2:1 splits, are retained. For each method, H-CR and H-TSR average c_i^{H} and s_i^{H} over its 50 videos as percentages; H-Extent averages e_i^{H} on the ordinal 1–5 scale.

C.3 Direct Human Comparison

In the human-only comparison (Table 6), State Forcing has the highest H-CR and H-TSR and the lowest H-Extent.

Table 6: Direct human evaluation on the paired 50-prompt subset. H-CR and H-TSR are percentages; H-Extent is the mean per-video median rating on the 1–5 scale.

Method	H-CR ↑	H-Extent ↓	H-TSR ↑
SkyReels-V2	14.00	2.74	2.00
MAGI-1	8.00	1.66	6.00
Self Forcing, chunk-wise	52.00	2.68	4.00
Rolling Forcing	46.00	1.98	14.00
Context Forcing	52.00	1.74	20.00
LongLive	44.00	2.26	4.00
State Forcing	**60.00**	**1.56**	**34.00**

C.4 Automatic–Human and Human–Human Agreement

Table 3 reports agreement pooled across methods. Auto–consensus compares automatic and consensus labels for all 350 videos, relating automatic TVR to e_i^{H} for violation extent. The evaluator is fixed independently of human annotations. Human–human pools all three within-video rater pairs in both orientations, giving $350 \times 6 = 2{,}100$ ordered pairs, using c_{ir}, a_{ir}, and e_{ir} for completion, strict success, and extent, respectively. Each video contributes equally; pairs from the same video are dependent.

For both comparisons, completion and strict success use macro-F1, equally averaging positive- and negative-class F1 with zero-denominator terms set to zero. Extent uses Spearman correlation with average ranks for ties. Statistics are computed once over pooled video judgments or rater pairs, rather than averaged across methods. Human–human agreement is a reference, not an upper bound on agreement with consensus. TVR–H-Extent correlation assesses ordinal correspondence, not duration calibration or temporal localization.

D Additional Experimental Details

D.1 Additional Qualitative Comparisons

Figures 6 and 7 compare State Forcing with four distilled causal baselines across different telic instructions. The examples reveal failures both before and after completion: baselines may repeatedly attempt a transition without finishing it, or reach the endpoint and then reverse or repeat the event. Hat duplication further shows that an apparent removal gesture need not produce the requested state change.

In these examples, State Forcing completes all terminal requirements and preserves the resulting states, including releasing the sunglasses after putting them on and maintaining the fully cooled appearance of the iron bar. Across both human actions and physical processes, these results illustrate sustained progress toward completion followed by consequence retention under an unchanged instruction.

D.2 Additional Analysis of Telic Execution

We compare the telic gains from sink-collapse mitigation, existing video OPSD, and explicit prompt planning with those from State Forcing. To align the starting backbone, we use LongLive-based versions of these methods, matching the base generator from which our training begins. Each method retains its own training procedure; State Forcing comprises telic data construction, warm-up, and state-privileged distillation as an integrated approach. All methods use the same test instructions, seeds, output specifications, and evaluation protocol (Section 4), with videos scored against the original instruction. This provides a common backbone and testing conditions for comparing the complete methods.

D.2.1 Sink-Collapse Mitigation and Telic Execution

We examine whether mitigating sink collapse is sufficient to address telic execution failures. Sink collapse involves abrupt recurrence of sink-frame content, while telic execution requires completing the requested transition and preserving its consequence. We apply the training-free LoL method (Cui et al., 2026b) to LongLive with its checkpoint, sampler, and cache policy unchanged.

LoL reduces TVR from 21.75% to 17.94% and raises TSR from 16.67% to 19.87%, while CR remains nearly unchanged (Table 7). These gains indicate partial mitigation, but substantial telic failures remain. State Forcing reaches 51.33% CR and 37.33% TSR while lowering TVR to 7.60%, improving both completion and strict success beyond sink-collapse mitigation alone.

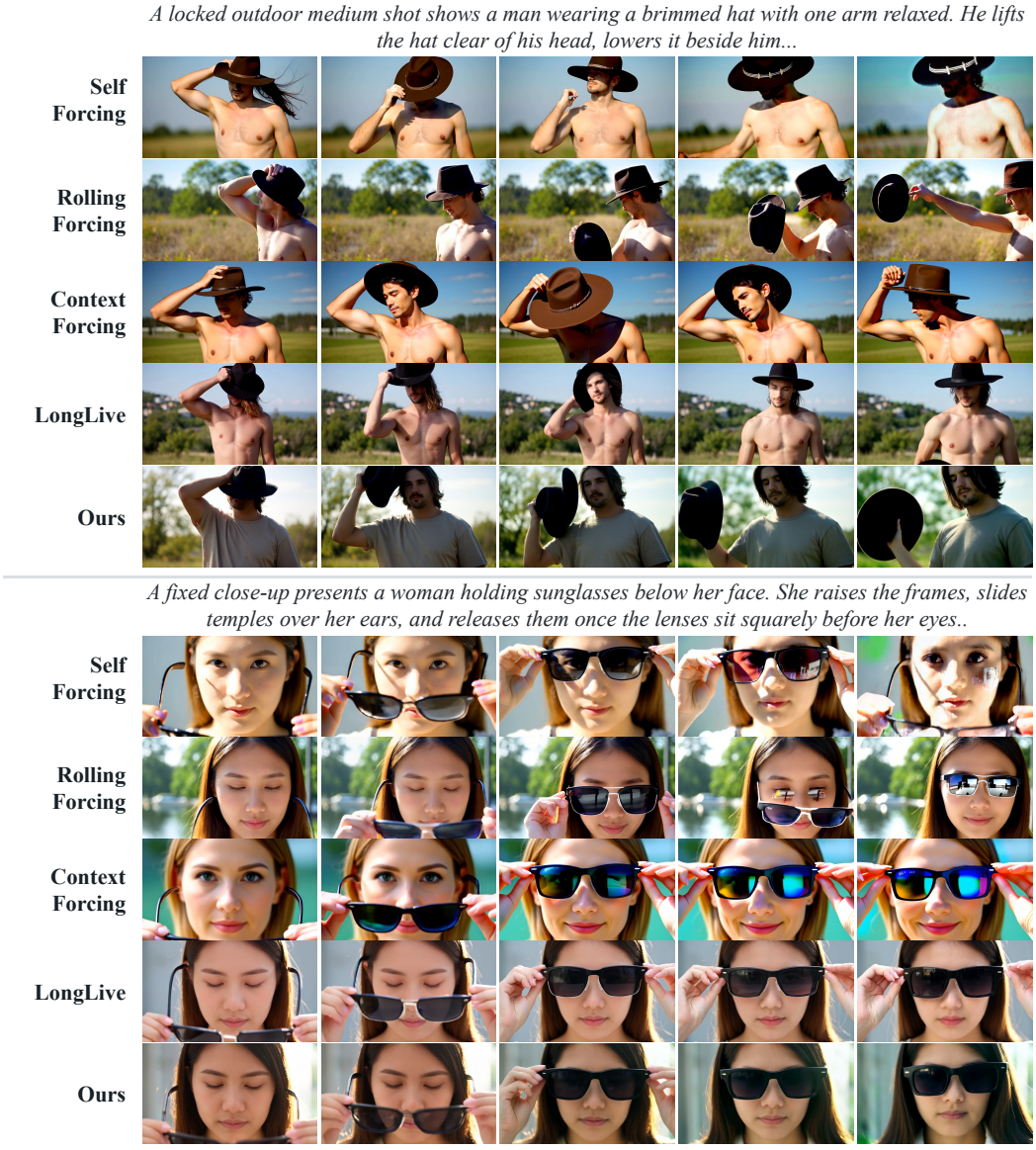


Figure 6: **Object manipulation under telic instructions.** Time progresses from left to right. **Hat removal (top):** Self Forcing and Context Forcing manipulate the hat but never remove it. Rolling Forcing and LongLive produce a second hat in hand while one remains on the head. STATE FORCING removes the hat and keeps it off. **Putting on sunglasses (bottom):** Self Forcing and Rolling Forcing repeatedly put on and remove the glasses. Context Forcing and LongLive keep attempting the transition without completing placement and releasing the glasses. STATE FORCING puts the glasses on, lowers both hands, and maintains the worn position. Best viewed in the supplementary videos.

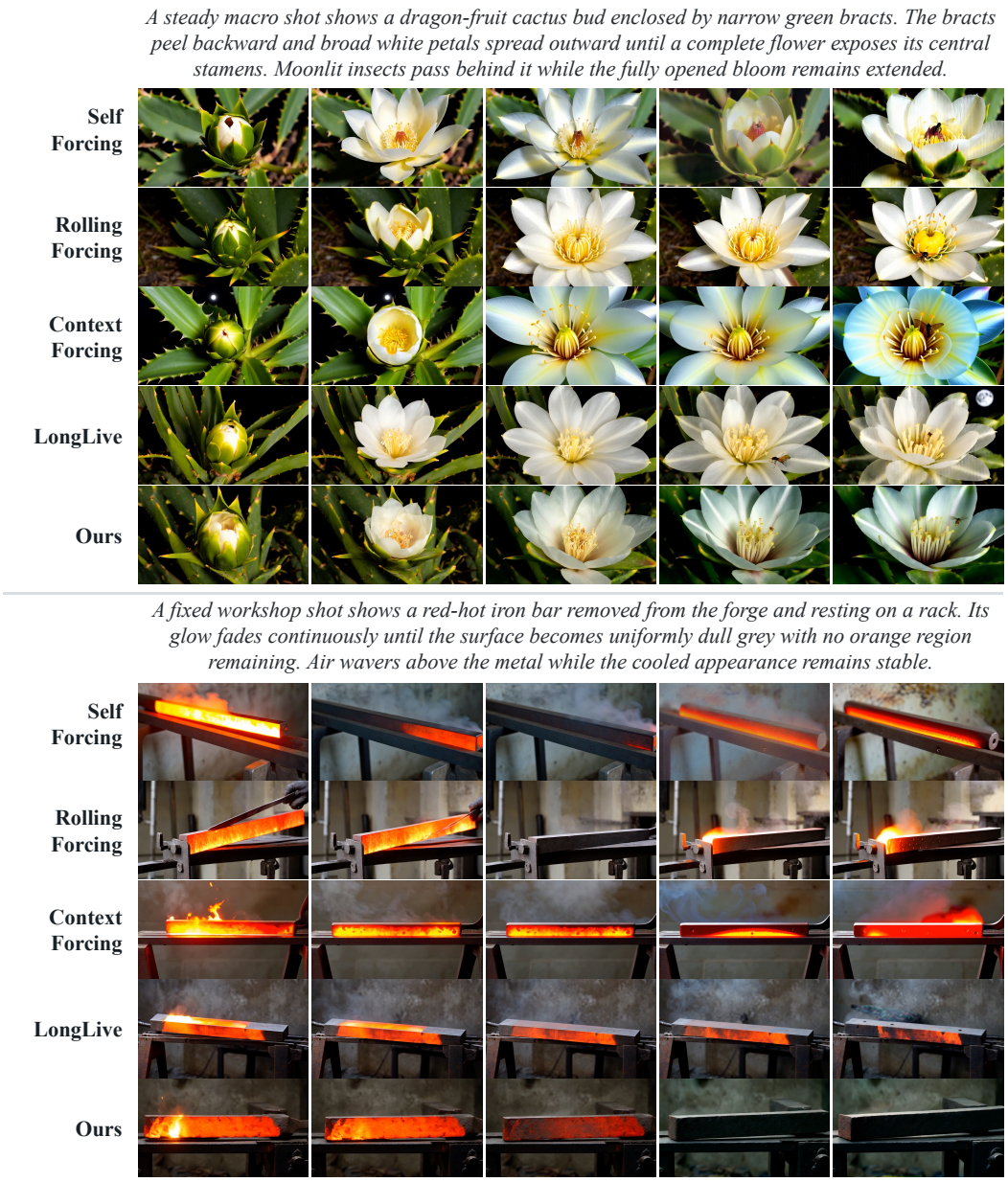


Figure 7: **Appearance changes under telic instructions.** Time progresses from left to right. **Cactus blooming (top):** All four baselines exhibit re-execution of the blooming transition, whereas STATE FORCING opens the flower and preserves the fully opened bloom. **Iron-bar cooling (bottom):** Self Forcing and Rolling Forcing complete cooling but subsequently restore the glow and restart the cooling process. Context Forcing and LongLive under-execute the transition, leaving orange regions instead of reaching a uniformly cooled state. STATE FORCING completes cooling and maintains the resulting appearance without renewed glow. Best viewed in the supplementary videos.

Table 7: **Sink-collapse mitigation and state-privileged training.** LongLive, LongLive with LoL, and STATE FORCING are evaluated under the shared TELICBENCH protocol. Scores are percentages.

Method	CR ↑	TVR ↓	TSR ↑
LongLive	39.33	21.75	16.67
LongLive + LoL	39.47	17.94	19.87
STATE FORCING	51.33	7.60	37.33

D.2.2 VIDEO OPSD AND STATE-PRIVILEGED SUPERVISION

We examine whether adopting existing video OPSD suffices to address telic execution. OPSD-V (Liu et al., 2026a) uses real-video history to provide a cleaner teacher context for long-horizon generation. STATE FORCING explicitly targets progress and consequence retention through telic data construction and state-privileged supervision. Since the original OPSD-V training data are not publicly available, we evaluate the authors' released LongLive-based checkpoint under its standard inference procedure, retaining a fixed instruction throughout generation.

OPSD-V lowers TVR from 21.75% to 17.10%, but yields no gain in CR or TSR over LongLive (Table 8). Thus, the evaluated video OPSD approach reduces violations but remains insufficient to resolve telic execution. STATE FORCING improves all three metrics, raising CR by 13.20 percentage points and TSR by 20.80 points over OPSD-V, while reducing TVR to 7.60%. These results demonstrate the advantage of our training approach tailored to completion and consequence retention over the existing video OPSD method.

Table 8: **Existing video OPSD and state-privileged training.** The official LongLive-based OPSD-V and full STATE FORCING are evaluated under the shared TELICBENCH protocol. Scores are percentages.

Method	CR ↑	TVR ↓	TSR ↑
LongLive	39.33	21.75	16.67
LongLive + OPSD-V	38.13	17.10	16.53
STATE FORCING	51.33	7.60	37.33

D.2.3 PROMPT PLANNING AND STATE-DEPENDENT EXECUTION

We examine whether learned prompt planning and execution resolve the telic failures targeted by STATE FORCING. We evaluate the complete LongLive-based TempAct system, including its learned planner and executor (Wang et al., 2026). TempAct uses stage-wise prompts, while STATE FORCING retains the original instruction. Neither system receives evaluation rubrics or oracle state information.

TempAct assigns prompts to fixed temporal spans, but completion time depends on the sampled execution. Planned switches may precede completion or leave transition prompts active afterward, and executing a new prompt still depends on the generated visual history. These requirements motivate learning progress and consequence retention within the generator.

TempAct improves CR and TSR over LongLive by 7.20 and 7.73 percentage points, respectively, showing gains from its planning and execution framework (Table 9). STATE FORCING further improves CR by 4.80 points and TSR by 12.93 points, while reducing TVR from 19.87% to 7.60%. Thus, explicit planning helps but does not fully resolve telic execution; STATE FORCING achieves stronger completion and fewer temporal violations while following an unchanged instruction.

Table 9: **Explicit prompt planning and state-privileged training.** TempAct uses its learned planner and executor with stage-wise prompts; STATE FORCING follows the original fixed instruction. All videos are assessed against that instruction under the shared TELICBENCH protocol. Scores are percentages.

Method	Inference conditioning	CR ↑	TVR ↓	TSR ↑
LongLive	Fixed instruction	39.33	21.75	16.67
LongLive + TempAct	Stage-wise prompts	46.53	19.87	24.40
STATE FORCING	Fixed instruction	51.33	7.60	37.33

E LIMITATIONS

Our current approach is built upon long-video tuning of streaming generators with attention sinks. Its self-distillation supervision depends on the underlying generator's capabilities, so telic instructions requiring complex physical transitions may remain challenging. Exploring stronger base generators and adapting STATE FORCING to other streaming frameworks are promising directions for future work. For evaluation, TELICBENCH relies on VLM judgments of sampled visual evidence: temporal sampling may miss brief violations, and VLM judgments may misinterpret state changes. Our agreement analysis with human annotations supports the reliability of the automatic assessments, while direct human evaluation corroborates the improvements in completion and strict success (Section 5.3).