ON TIME: LEARNING TO FOLLOW EVENT TIMELINES IN LONG VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-time long video generation requires actions to follow prescribed start and end times, preserve intended overlaps, and remain associated with the correct participants throughout an extended rollout. We introduce *OnTime*, a framework for this task that combines structured event timelines with a few-step autoregressive generator trained on short sequences. Its core mechanism, *Rolling Timeline Conditioning* (RTC), preserves global action intervals while assigning persistent scene and identity descriptions bounded local support, so their encoded durations do not grow with the full rollout. Event cross-attention retains the global timeline while cached visual history uses window-relative positions. We train the causal generator through teacher-guided initialization and Self-Forcing. In 60-second rollouts from 10-second training sequences, our model leads the compared systems in event-level visual quality and InternVideo2 alignment, while LongLive leads full-video Temporal-Following Score. Our code will be released.

1 INTRODUCTION

Long video generation requires a model to maintain scene continuity while following an event timeline. For example, a minute-long video may require one person to water a plant from 10 to 20 seconds while another walks past from 14 to 18 seconds. Satisfying this instruction requires both actions to occur within their prescribed intervals, preserve their overlap, and remain associated with the correct participants. We refer to this objective as *on-time long video generation*. Recent autoregressive models provide a foundation for long video generation: Self-Forcing (Huang et al., 2025) addresses exposure bias by training on self-generated histories, while LongLive (Yang et al., 2025) combines long-video tuning with visual memory and prompt transitions. Our focus is how to supply these generators with event conditions that retain their timing and relationships throughout a long rollout.

Temporal conditioning makes these relationships explicit. MinT (Wu et al., 2025) associates event captions with designated time spans, and TIE (Shu et al., 2026) encodes event intervals directly in text–video cross-attention. Transferring interval conditioning from short training clips to long rollouts, however, introduces a mismatch in the temporal support of persistent descriptions. With the adopted duration-aware encoding, assigning scene and identity descriptions to the full rollout increases their encoded duration even when their content is unchanged, potentially extending it beyond the training range. This observation motivates giving persistent information local temporal support while preserving the original action intervals on the global timeline.

We introduce *OnTime*, a framework for on-time long video generation from explicit event timelines (Figure 1). Its core conditioning mechanism, *Rolling Timeline Conditioning* (RTC), connects short-sequence training to long-rollout inference. Our method separates narrative prompts into persistent scene and identity descriptions and subject-specific timed actions. We retain the pretrained interval representation in a few-step causal student trained on short sequences using teacher-guided initialization and Self-Forcing. At inference, each generation block receives a rolling view of the global timeline: persistent descriptions are assigned local support, while actions intersecting the window retain their original start and end times, including when they extend beyond the window or cross block boundaries. Concurrent actions can therefore condition the same block without changing their prescribed intervals. Event conditioning remains on a continuous global timeline, independently of

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.

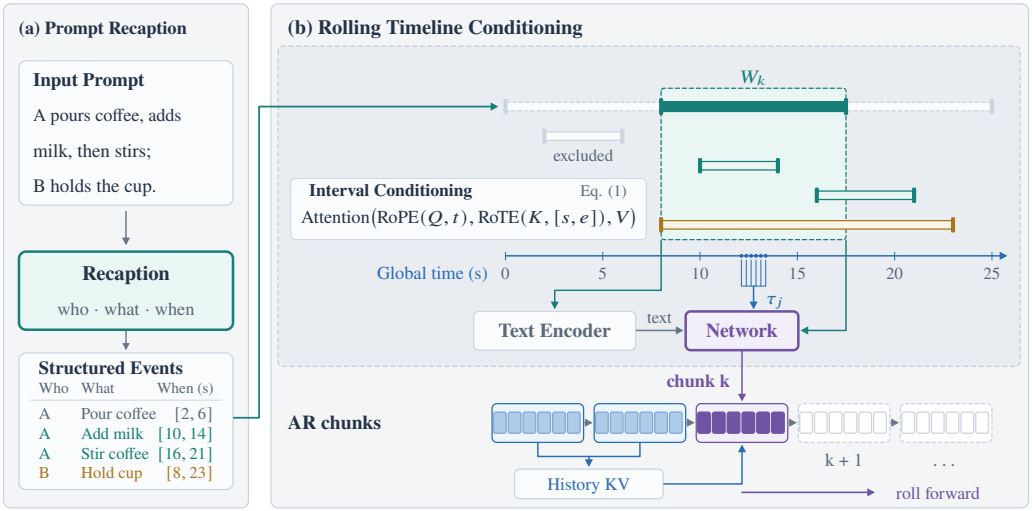


Figure 1: **OnTime combines event-timeline construction and rolling conditioning for long video generation.** (a) Recaptioning structures subject-specific actions with explicit intervals. (b) Rolling Timeline Conditioning (RTC) selects actions intersecting the current block's conditioning window W_k without clipping their intervals; the top bar assigns scene and identity descriptions local support within W_k. In the interval-conditioning inset, RoPE encodes each video query Q at its global timeline position $t = \tau_j$, while RoTE encodes the center and duration of each text interval $[s, e]$ in its key K. Cross-attention combines semantic content with the query's position relative to each interval and its duration, providing soft temporal guidance for overlapping actions. Text values V carry content without this temporal transform. Cached visual history supplies causal context while event conditioning remains on the global timeline.

the positions used to manage cached visual history. This conditioning construction adds no trainable parameters.

Our main contribution is the rolling construction of conditions that connects short-sequence training to long video inference. We also make explicit a corollary of the adopted rotary interval encoder: for fixed content features, shifting a video query and an event interval by the same amount preserves their attention logit. Its scope is the encoder's positional behavior; timing accuracy in generated videos remains an empirical question. We evaluate our method on 60-second timelines using a generator trained on 10-second sequences and existing evaluation metrics. Our model achieves the highest event-level visual quality, motion quality, and InternVideo2 alignment among the compared systems, while LongLive leads full-video temporal following and LongLive Interactive leads text alignment.

2 METHOD

We present *OnTime*, a framework for following explicit event timelines in long video generation (Figure 1). Its core mechanism, *Rolling Timeline Conditioning* (RTC), adapts event conditioning to rolling generation beyond the training horizon. We first construct event timelines from narrative prompts, separating persistent scene and identity descriptions from actions with prescribed intervals (Section 2.1). We then use pretrained interval conditioning (Shu et al., 2026) to associate video queries and event text on a shared global timeline (Section 2.2). Finally, we train a few-step causal student from a bidirectional teacher using ODE initialization (Yin et al., 2025) and Self-Forcing (Huang et al., 2025) on short sequences. We extend this generator to long rollouts by selecting nearby events and localizing persistent context while preserving global action intervals (Section 2.3).

2.1 PROMPT DECOMPOSITION AND EVENT TIMELINE CONSTRUCTION

A video of duration T is specified by a conditioning plan $\mathcal{C}$ containing a scene description g, an appearance registry $\mathcal{A}$, and an event set $\mathcal{E} = \{(u_i, d_i, s_i, e_i)\}_{i=1}^{N}$. Here u_i identifies a participant or condition track, d_i describes a visible event, and $[s_i, e_i] \subseteq [0, T]$ specifies its interval in seconds. Events may be sequential, concurrent, or staggered. Scene and appearance entries condition generation but are excluded from event-level evaluation. Physical video time is distinct from the diffusion noise level throughout.

From narrative prompts to event timelines. We transform each original narrative prompt into a structured event timeline. First, we separate global scene context and persistent participant attributes from dynamic actions. We then decompose the action narrative into fine-grained, observable events and recaption each event with an explicit subject and a self-contained action description. Finally, we place each event on the timeline with a prescribed interval, preserving causal dependencies and intended concurrency. The resulting event descriptions specify what should happen, who performs it, and when it should occur, providing the interval-conditioned generator with more explicit guidance than the original narrative alone. We use a language-model agent for this recaptioning procedure; Appendix A describes its instructions, timeline construction, and validation protocol. Alternative schedules preserve the aligned event graph's participant identities, causal order, and intended concurrency. Each token in the serialized scene, appearance, and event descriptions inherits its description's interval for conditioning.

2.2 INTERVAL CONDITIONING ON A GLOBAL VIDEO TIMELINE

Long video conditioning must distinguish actions with prescribed start and end times from scene and identity information that persists throughout the video. Untimed narrative prompts leave action boundaries implicit, while sequential prompt updates (Yang et al., 2025) alone do not specify independent intervals for concurrent actions. We therefore use explicit interval conditioning (Wu et al., 2025; Shu et al., 2026) to represent each action's temporal extent. Extending this representation to long rollouts introduces a separate issue for persistent descriptions. In the adopted encoder, interval duration modulates each text key. Full-video support makes this duration grow with the rollout, potentially beyond the short-sequence training range, even when the persistent text is unchanged. We instead keep persistent information available by supplying the same descriptions in every block with bounded local support. Actions retain their prescribed intervals on the global timeline (Section 2.3).

We adopt TIE's interval-conditioned cross-attention (Shu et al., 2026):

$$\boxed{\mathrm{Attention}(\mathrm{RoPE}(Q, t),\ \mathrm{RoTE}(K, [s, e]),\ V)} \tag{1}$$

Here Q denotes video queries, while K and V are text keys and values. Rotary position embeddings (RoPE) (Su et al., 2024) encode each query's position t on the video timeline, and RoTE encodes each text key's event interval $[s, e]$; the text values V carry content without this temporal transform.

We retain this parameter-free encoding in both the bidirectional teacher and the causal student. During autoregressive generation, video queries and event intervals share a consistent global timeline, independently of rolling-cache storage positions, so an event keeps the same temporal meaning across generation blocks. The encoding supplies a soft temporal conditioning mechanism, while Self-Forcing adapts the causal student through teacher-guided distribution matching on generated histories.

The interval encoder maps a query to $R(t)\mathbf{q}$ and an event key to $R(c)D(r)\mathbf{k}$, where $c = (s + e)/2$, $r = (e - s)/2$, R is a rotary matrix, and $D(r)$ is normalized sinc duration modulation, assumed well-defined with a nonzero normalization factor C_r. The full definition of $D(r)$ is given in Appendix B.2.

2.3 ROLLING TIMELINE CONDITIONING BEYOND THE TRAINING HORIZON

We first train a few-step causal generator on short sequences following the Self-Forcing pipeline (Huang et al., 2025). ODE initialization starts the student from the frozen bidirectional teacher's weights, retaining interval conditioning while restricting video attention to the current and

preceding blocks. With the full short-sequence event timeline available, the student learns to predict teacher trajectory endpoints from independently sampled block noise levels. We then train it on its own autoregressive rollouts using KV-cached generated history. Distribution matching uses the frozen teacher and a trainable auxiliary score model.

At inference, our method extends this generator beyond its training horizon. Long rollouts increase the temporal support of persistent descriptions beyond the short-clip training range. Extending a scene description from a short clip to the full rollout changes its duration modulation $D(r)$ even when its text is unchanged. We address this change with a rolling view of the event timeline. At each autoregressive block, we select temporally nearby actions and give persistent descriptions local support, keeping the original action intervals and rotary frequency scale. The procedure operates at inference and adds no trainable parameters.

Rolling view of the event timeline. Let $B_k = [a_k, b_k]$ be the interval covered by the video latents in block k on the global timeline. A nonnegative margin ρ controls how much context is included before and after the block:

$$\begin{aligned} W_k &= [\ell_k, h_k], \\ \ell_k &= \max(0, a_k - \rho), \\ h_k &= \min(T, b_k + \rho). \end{aligned} \tag{2}$$

We select actions whose intervals intersect this window:

$$\mathcal{I}_k = \{i \in \{1, \dots, N\} : e_i \geq \ell_k,\ s_i \leq h_k\}. \tag{3}$$

Selection uses the whole event interval, so an ongoing action remains eligible even if it started before the window. Multiple participants and overlapping actions can be selected together. The margin also includes nearby past and future actions as context; per-latent query times provide finer temporal conditioning within the block.

Let $\mathcal{P}$ index the persistent scene and appearance descriptions, separately from action indices. These descriptions are serialized with full-video support $[0, T]$. We retain them in every block together with the actions in $\mathcal{I}_k$.

Local support for persistent conditions. For each selected description j, the conditioning interval is

$$I_j^{(k)} = \begin{cases} [\ell_k, h_k], & j \in \mathcal{P}, \\ [s_j, e_j], & j \in \mathcal{I}_k. \end{cases} \tag{4}$$

Persistent text remains unchanged across blocks; only its per-token interval metadata is updated after text encoding. Its support has length at most $(b_k - a_k) + 2\rho$, independent of the total rollout duration T. This bounds the temporal support assigned to a description that supplies the same scene or identity information throughout the video. For actions, we preserve both original endpoints and the full duration, including when an action extends beyond W_k or crosses a generation-block boundary. Window edges therefore do not become artificial action boundaries.

Global timeline and cached visual history. Each video query's position on the global timeline is derived from its global latent index. Event intervals are defined on the same timeline, with its origin fixed at zero throughout the rollout. Separately, the visual KV cache stores *unrotated keys and values*. During rolling generation, we apply window-relative RoPE on the fly to the current queries and retrieved keys, including those of the attention sink; values remain unrotated. This lets retained keys receive new positional rotations as the cache window advances. Text–video cross-attention continues to use positions and intervals on the global timeline.

The selected text and interval metadata define a block condition $\mathcal{C}_k$, which remains fixed through that block's few-step denoising. The denoiser combines $\mathcal{C}_k$ with cached generated history; the text–video cross-attention path recomputes text keys and values from the active condition. After generating the block, a context forward pass updates the visual cache, and the next block receives its own $\mathcal{C}_{k+1}$. The teacher and auxiliary score model are used only during training.

Positional basis for long rollouts. The rolling procedure above preserves global action intervals while extending generation beyond the training horizon. To clarify the role of absolute position, we

consider moving a fixed-duration event and its video query later on the timeline by the same amount. RoPE's relative-position identity yields the following corollary for the adopted interval encoder (Shu et al., 2026, Section 3.2 and Appendix A.1).

Corollary 1 (Common temporal translation) *For an attention head of even width d, with fixed content features $\mathbf{q}, \mathbf{k} \in \mathbb{R}^d$ and a fixed frequency scale, define the valid-event attention logit $S(\mathbf{q}, \mathbf{k}; t, [s, e]) = d^{-1/2}(R(t)\mathbf{q})^\top R(c)D(r)\mathbf{k}$. For any common shift Δ of the frame position and event interval along the timeline,*

$$S(\mathbf{q}, \mathbf{k}; t + \Delta, [s + \Delta, e + \Delta]) = S(\mathbf{q}, \mathbf{k}; t, [s, e]). \tag{5}$$

Corollary 1 provides a positional basis for placing fixed-duration events later in a long video: for fixed content features, the logit depends on the relative offset $c - t$ and duration $2r$. The global timeline preserves the intended frame–event offsets as the visual cache rolls. In contrast, increasing an interval's duration can change $D(r)$, which is why we bound the support of persistent descriptions while retaining each action's prescribed duration. Together with rolling event selection, these choices extend conditioning along the timeline without rescaling time or changing the encoder. The proof is given in Appendix B.2.

The corollary concerns individual valid-event logits with fixed content features. Full cross-attention invariance additionally requires consistent handling of all attended keys, values, and masks, including padding. Changing text subsets and generated history can change the content features, so temporal accuracy and generation quality later in a rollout still require empirical evaluation. Generalization to longer individual actions remains a separate question.

3 Experiments

We evaluate OnTime on 60-second event timelines using a generator trained on 10-second sequences. The reported evaluation considers adherence to the full timeline and visual quality and semantic alignment within requested event intervals.

3.1 Experimental Details

Training dataset. We use a prompt-only corpus of 4,000 source scenes, combining constructed scenes with rewritten PexelsEvents (Shu et al., 2026) descriptions. Five temporal variants per scene yield 20,000 10-second schedules for Self-Forcing; these are generation conditions, not paired real videos. ODE initialization uses teacher-generated trajectories from 2,000 scenes. Appendix C.1 details corpus composition, schedule construction, text encoding, and validation splits.

Evaluation dataset. The 60-second benchmark contains 32 source scenes with four authored timing-and-detail variants each, giving 128 conditions (Table 3 in Appendix D). Scene references are sampled from the project's training-prompt pool, stratified by participant count, and adapted into 60-second event timelines. These are controlled in-domain evaluations, not tests of generalization to independent unseen scenes. The new timelines are authored conditions rather than annotations verified against source videos; no paired 60-second ground-truth video is assumed. Event records, scoring eligibility, and conditioning-input budgets are specified in Appendix D.1.

Baseline selection. We select autoregressive baselines that support 60-second generation and cover both long-rollout coherence and prompt-driven control. Self-Forcing (Huang et al., 2025) shares our student's training principle of conditioning on self-generated histories and provides a reference for few-step autoregressive generation. LongLive (Yang et al., 2025) targets coherence over extended rollouts through streaming long-video tuning and persistent visual context. We evaluate both standard LongLive and LongLive Interactive; the latter uses sequential prompt updates with KV recaching, providing a reference for temporal control through prompt transitions. These are comparisons between complete systems with different checkpoints and conditioning interfaces.

Backbone and temporal representation. OnTime uses a **5B-parameter Wan2.2-TI2V backbone** (Wan-AI, 2025) for all results in Table 1. We initialize the causal student from the released

Table 1: Long video generation on 60-second event timelines using the 128-condition benchmark. OnTime uses a **5B-parameter Wan2.2-TI2V backbone**. VideoReward reports visual quality (VQ), motion quality (MQ), and text alignment (TA), with $\mathrm{Overall} = \mathrm{VQ} + \mathrm{MQ} + \mathrm{TA}$. InternVideo2 measures video–text alignment, and TFS measures full-video temporal following. VideoReward and InternVideo2 are evaluated per event, using clips within each requested interval and the corresponding event description. The best and second-best scores in each column are bold and underlined, respectively.

	VideoReward					
Method	**VQ ↑**	**MQ ↑**	**TA ↑**	**Overall ↑**	**InternVideo2 ↑**	**TFS ↑**
Self-Forcing (Huang et al., 2025)	<u>-0.1898</u>	0.0835	-0.5646	-0.6710	0.0076	0.3585
LongLive (Yang et al., 2025)	-0.2274	0.1297	<u>-0.5317</u>	-0.6293	0.0096	**0.4165**
LongLive Interactive (Yang et al., 2025)	-0.2551	<u>0.1741</u>	**-0.4797**	<u>-0.5608</u>	<u>0.0102</u>	0.3310
OnTime (Ours)	**0.0879**	**0.5376**	-0.5971	**0.0284**	**0.0194**	<u>0.3714</u>

bidirectional TIE checkpoint, retaining its capacity and pretrained event conditioning. Self-Forcing optimizes LoRA adapters (Hu et al., 2022) in the student and auxiliary score model. Appendix C details temporal indexing, training, text encoding, and long-rollout inference. We will release our code to support reproducibility.

3.2 Evaluation metrics

For the main comparison, we omit Fréchet Video Distance (FVD) (Unterthiner et al., 2018) because our authored 60-second timelines lack a matched real-video reference collection, and VBench (Huang et al., 2024) because its standard scores do not directly assess requested event intervals. Our event scores do not penalize out-of-interval activity and cannot replace boundary-error measurements. Appendix D details sampling and aggregation.

We assess visual and motion quality, event-level semantic alignment, and full-timeline order and consistency. **Temporal-Following Score (TFS)** follows TempAct's rubric and weights (Wang et al., 2026), using Qwen3-VL-8B-Instruct (Qwen Team, 2025) to judge 40 sampled frames against the complete event timeline. It measures global temporal following, not precise event-boundary accuracy. Appendix D.2 specifies our sampling and frame-to-second mapping adaptations; Appendix Table 4 reports the six component scores.

InternVideo2-Event adds a representation-based semantic check alongside preference-based quality scores. It measures alignment between each event description with referenced actor identities and video clips sampled strictly within its requested interval, using the InternVideo2 Stage2 encoder (Wang et al., 2024). We take a duration-weighted mean of raw cosine similarities over windows within each event, then average equally across events; no softmax or interval-exterior penalty is applied.

VideoReward-Event uses VideoReward (Liu et al., 2025) to separate perceptual quality from semantic alignment, making their tradeoffs visible. It evaluates each event interval against its event description and referenced actor identities. We report its three outputs: visual quality (**VQ**), motion quality (**MQ**), and text alignment (**TA**), using the released model's score normalization and averaging equally across events. **Overall** is their sum. Higher values are better for all reported metrics; the VideoReward scores can be negative and are not percentages.

3.3 Long video evaluation

Comparison on 60-second event timelines. OnTime's clearest advantage is visual and motion quality within requested event intervals. We compare it with Self-Forcing, LongLive, and LongLive Interactive on the 128 conditions in Appendix D, using common event references and matched generation seeds. OnTime leads VQ, MQ, and Overall in Table 1, with an Overall score of 0.0284 versus -0.5608 for the strongest baseline, LongLive Interactive. Because Overall sums VQ, MQ, and TA, this advantage reflects higher visual and motion quality outweighing weaker text alignment: OnTime has the lowest TA. Its highest InternVideo2-Event score therefore does not indicate a uniform semantic improvement across evaluators. The disagreement between the two semantic scores makes

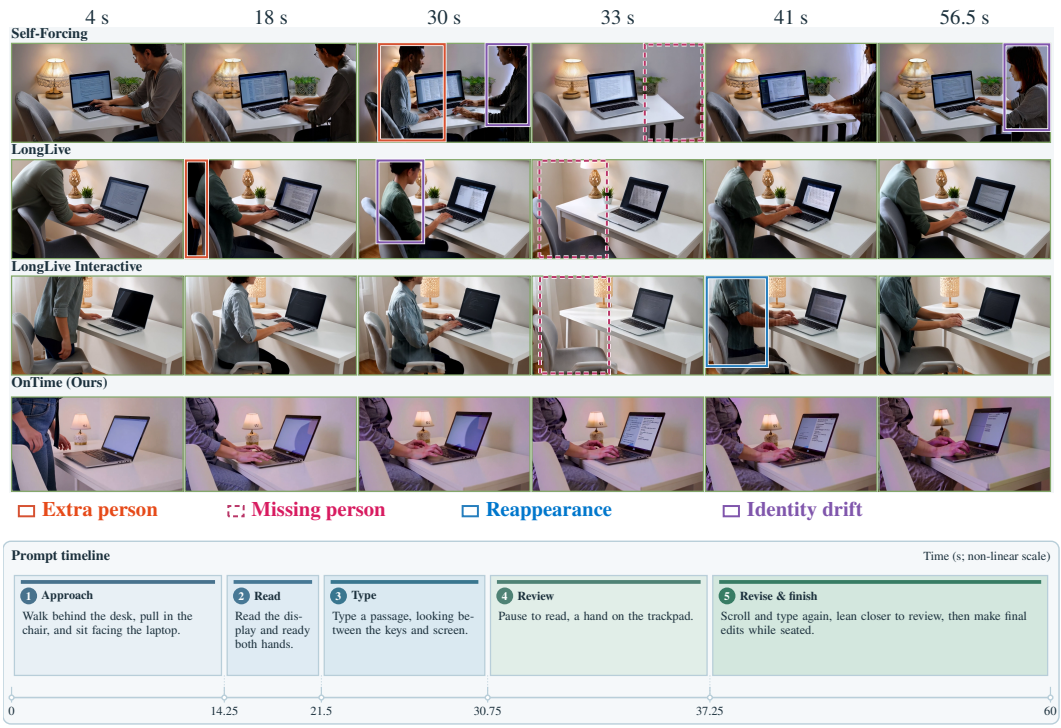


Figure 2: **OnTime maintains participant continuity in this long video example.** Self-Forcing, LongLive, LongLive Interactive, and OnTime are shown at the same positions on the 60-second timeline in uniformly spaced columns. Self-Forcing and LongLive introduce extra people and visible changes in participant appearance, while all three baselines lose the participant in the 33 s frame. The blue box highlights a later reappearance in LongLive Interactive.

it important to read quality and alignment separately rather than infer event-following accuracy from the aggregate score alone.

Full-video timeline following remains a distinct challenge. OnTime's TFS of 0.3714 exceeds Self-Forcing and LongLive Interactive but trails LongLive's 0.4165. LongLive also leads all six TFS components (Appendix Table 4), so the gap is not confined to one rubric dimension. Better quality within event intervals thus does not establish stronger adherence to the complete timeline. In particular, event-cropped scores do not penalize activity outside the requested interval, and TFS does not directly measure action onset or offset errors. The current results leave precise on-time execution unresolved. These are complete-system comparisons with different checkpoints and conditioning inputs; they do not isolate RTC's contribution. Confidence intervals are not reported.

Human evaluation. We compare OnTime, Self-Forcing, and LongLive on 40-second prefixes from ten selected scenes, ranking event fulfillment, timing, continuity, visual quality, and overall preference. OnTime leads all five mean ranking scores in Table 2, including timing (93.48) and overall preference (85.19). Appendix E details the study and scoring protocol. These descriptive results from a small, selected sample measure perceived performance; precise event-boundary accuracy requires direct start/end annotations.

Qualitative comparison of long video rollouts. Participant continuity exposes a failure mode that scene appearance alone can conceal. In Figure 2, the desk and laptop remain visible even when the intended participant disappears from all three baselines at 33 s. Extra people and changes in participant appearance introduce further ambiguity about who performs the requested actions. LongLive Interactive shows a later reappearance at 41 s, but this does not undo the missing participant in the earlier sample. OnTime retains one participant with consistent clothing and desk layout across all six sampled positions. This example illustrates why long video generation must preserve participant context alongside action conditions: later actions depend on retaining their intended subject

Table 2: Human evaluation using mean ranking scores (0–100). First, second, and third place receive 100, 50, and 0 points; ties receive the average points of the occupied ranks. Each valid scene–session group has equal weight, with 24, 23, 27, 27, and 27 groups in column order (Appendix E). Higher is better; the best and second-best scores in each column are bold and underlined, respectively.

Method	Event	Timing	Continuity	Quality	Overall
Self-Forcing	2.08	2.17	1.85	1.85	0.00
LongLive	<u>52.08</u>	<u>54.35</u>	<u>61.11</u>	<u>70.37</u>	<u>64.81</u>
OnTime (Ours)	**95.83**	**93.48**	**87.04**	**77.78**	**85.19**



Figure 3: **Timeline and attention over all descriptions.** (a) Requested timeline; dotted lines mark sampled RGB times. (b) Self-Forcing, non-interactive LongLive 1.0, and OnTime at layer 15 (zero-based), fourth denoising forward: attention summed over description tokens and averaged over all heads and full-frame queries. Colors use a shared log scale with all-key normalization; gray denotes descriptions absent from the input context. Figure 5 in the appendix includes the corresponding principal-person crops.

throughout the rollout. The displayed timeline specifies requested actions; the sparse frames do not establish their precise start and end times.

Timeline and attention diagnostic. Figure 3 aligns the full requested timeline with attention over every description at three shared timestamps. Appendix F gives the complete visualization, including the principal person's visible state at each time. This auxiliary 30-second example uses a step-600 checkpoint trained on a 20,000-prompt subset, separately from the 60-second benchmark. Scene and identity descriptions are shown separately from all action events in the attention matrices. Gray cells identify descriptions excluded by the rolling context, rather than low attention to an available condition. G0 denotes the scene, G1–G3 person identities, and E4–E26 actions; Text collects residual timestamps, separators, and instruction text. The baseline compact prompts, checkpoints, and sampling schedules differ, so attention magnitudes are descriptive and do not isolate RTC's effect or provide a cross-model quality score.

4 Related work

Autoregressive Long Video Generation. Autoregressive video diffusion provides a way to extend video duration by combining sequential generation with continuous-valued denoising. Diffusion Forcing (Chen et al., 2024) trains causal sequence models with independently sampled token noise levels. Rolling Diffusion (Ruhe et al., 2024) assigns higher noise levels to later frames and generates sequences through sliding-window denoising. In discrete token space, MaskFlow (Fuest et al., 2025) uses frame-level masking and conditioning on previously generated frames to extend generation beyond the training horizon. CausVid (Yin et al., 2025) combines teacher-trajectory initialization with asymmetric distribution matching to distill a bidirectional video model into a few-step causal generator. Self-Forcing (Huang et al., 2025) addresses exposure bias by training on self-generated histories with KV caching and video-level distribution matching. LongLive (Yang et al., 2025) extends streaming generation through long-video tuning, short-window attention with frame sinks, and KV recaching for prompt transitions. More recently, LongLive-2.0 (Chen et al., 2026) develops sequence-parallel training and low-precision execution to improve the efficiency of long-video generation. FramePack (Zhang et al., 2025) compresses visual history according to frame importance within a fixed context budget and studies sampling strategies that reduce drift. Our method uses a few-step causal generator trained with Self-Forcing. Its focus is the conditioning supplied to that generator throughout a long rollout: preserving event intervals and their relationships on a global timeline as generation proceeds.

Temporally Controllable Video Generation. Temporal control requires associating event descriptions with their intended positions and durations in a video. FreeNoise (Qiu et al., 2024) extends pretrained video diffusion through noise rescheduling and window-based attention fusion, with motion injection for multiple text prompts. At the planning level, VideoDirectorGPT (Lin et al., 2024) constructs scene descriptions, entity layouts, and cross-scene consistency groups for grounded video generation. MinT (Wu et al., 2025) introduces temporal cross-attention with rescaled rotary position embeddings to bind event captions to specified time spans. TIE (Shu et al., 2026) represents events as intervals within text–video cross-attention, integrating rotary positional evidence over each interval with duration normalization. This formulation supports conditioning on concurrent and overlapping events (Shu et al., 2026). We retain this representation in both the bidirectional teacher and the causal student (Section 2.2). The rolling construction in Section 2.3 gives persistent scene and identity descriptions local support while keeping action intervals intact on the global timeline. This connects interval conditioning learned on short sequences to long video generation, the setting addressed by the evaluation protocol above.

RoPE-based Length Extension. Adapting rotary positions or frequencies can extend a pretrained model's usable horizon. In language models, YaRN (Peng et al., 2024) combines frequency-dependent interpolation with attention rescaling, while LongRoPE (Ding et al., 2024) searches nonuniform positional interpolation factors. For video diffusion, RIFLEx (Zhao et al., 2025) reduces an intrinsic temporal RoPE frequency to suppress repetition during length extrapolation. For streaming generation, StreamingLLM (Xiao et al., 2024) assigns positions within a bounded cache of attention sinks and recent tokens, and Infinity-RoPE (Yesiltepe et al., 2026) reanchors visual-token positions within a moving local reference frame. OnTime uses related window-relative positions for cached visual history. RTC operates on text–video conditioning: it selects nearby events and bounds persistent descriptions' temporal support while preserving global action intervals and the pretrained interval encoder's frequency scale (Section 2.3).

5 Conclusion

We presented OnTime, a framework for following event timelines in long autoregressive video generation, with Rolling Timeline Conditioning (RTC) as its core conditioning mechanism. Our method gives persistent scene and identity descriptions local temporal support while preserving action intervals on a global timeline. This construction connects interval conditioning and causal generation learned on short sequences to long video inference. The common-time-shift analysis establishes a positional property for fixed content features. In 60-second rollouts, our model obtains the strongest event-level visual quality and InternVideo2 alignment among the compared systems, but does not lead full-video temporal following or text alignment.

AI USE STATEMENT

We used language-model assistants for manuscript drafting and revision, literature discovery, source-code inspection, feedback on evaluation design, interpretation of experimental results, and revision of mathematical exposition and proofs. The revision workflow included primary-source citation checks, LaTeX compilation, and rendered-PDF inspection. Language-model agents also supported recaptioning, construction of structured event timelines, and semantic review of prompt conditions, as described in Appendix A. Structural validation checks the exported conditions, while model-assisted semantic review remains distinct from independent human or source-video verification. Our automatic Temporal-Following Score uses Qwen3-VL-8B-Instruct to judge generated videos; its inputs, sampling procedure, and limitations are specified in Appendix D. These scores are model judgments. The authors retain responsibility for the manuscript, research claims, and artifacts produced with AI assistance.

ETHICS STATEMENT

Improved control over long video generation can support creative work, but can also facilitate misleading narratives, impersonation, and non-consensual depictions. Generated videos should be identified as synthetic, and use of source materials and pretrained models should respect privacy, consent, and applicable licenses. Our generator, agent-authored prompts, and learned evaluators may inherit biases from their underlying models and data; the reported quality and timeline scores do not establish fairness or safety. The prompt corpus combines constructed scenes with rewritten PexelsEvents descriptions, as detailed in Section 3.1.

REPRODUCIBILITY STATEMENT

Section 2 describes the conditioning mechanism and training pipeline, and Section 3.1 specifies the 5B backbone, prompt corpus, and data splits. Appendix A documents agent-based prompt construction and validation; Appendix C gives training and inference settings; and Appendix D specifies benchmark construction, evaluator settings, sampling, and score aggregation. Appendix E documents the human-study stimuli, anonymous presentation, ranking questions, and response aggregation. The implementation records random seeds, corpus checksums, model revisions, and source and checkpoint provenance to identify the configurations used. Reproduction also requires the corresponding model assets and prompt files: the training source snapshots do not bundle pretrained weights, the full training corpus, or training checkpoints.

REFERENCES

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *arXiv preprint arXiv:2407.01392*, 2024. URL https://arxiv.org/abs/2407.01392.

Yukang Chen, Luozhou Wang, Wei Huang, Shuai Yang, Bohan Zhang, Yicheng Xiao, Ruihang Chu, Weian Mao, Qixin Hu, Shaoteng Liu, Yuyang Zhao, Huizi Mao, Ying-Cong Chen, Enze Xie, Xiaojuan Qi, and Song Han. LongLive-2.0: An NVFP4 parallel infrastructure for long video generation. *arXiv preprint arXiv:2605.18739*, 2026. URL https://arxiv.org/abs/2605.18739.

Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. UniMax: Fairer and more effective language sampling for large-scale multilingual pretraining. In *ICLR*, 2023. URL https://openreview.net/forum?id=kXwdL1cWOAi.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. LongRoPE: Extending LLM context window beyond 2 million tokens. In *ICML*, 2024. URL https://proceedings.mlr.press/v235/ding24i.html.

Michael Fuest, Vincent Tao Hu, and Björn Ommer. MaskFlow: Discrete flows for flexible and efficient long video generation. *arXiv preprint arXiv:2502.11234*, 2025. URL https://arxiv.org/abs/2502.11234.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025. URL https://arxiv.org/abs/2506.08009.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_VBench_Comprehensive_Benchmark_Suite_for_Video_Generative_Models_CVPR_2024_paper.html.

Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. VideoDirectorGPT: Consistent multi-scene video generation via LLM-guided planning. In *COLM*, 2024. URL https://openreview.net/forum?id=sKNIjS2brr.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di Zhang, Kun Gai, Yujiu Yang, and Wanli Ouyang. Improving video generation with human feedback. In *Advances in Neural Information Processing Systems*, 2025. URL https://papers.neurips.cc/paper_files/paper/2025/hash/76227feb18ea0ee40bd15cf02c33e18e-Abstract-Conference.html.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *ICLR*, 2024. URL https://openreview.net/forum?id=wHBfxhZu1u.

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. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/15ce8e7afe5ee95bad56e3b9be28d3d1-Abstract-Conference.html.

Qwen Team. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42818–42835. PMLR, 2024. URL https://proceedings.mlr.press/v235/ruhe24a.html.

Zhilei Shu, Shangwen Zhu, Zihang Liang, Xiaofan Li, Qianyu Peng, Xinyu Cui, Bo Ye, Yiming Li, Fan Cheng, Jian Zhao, Yang Cao, Zheng-Jun Zha, and Ruili Feng. TIE: Time interval encoding for video generation over events. *arXiv preprint arXiv:2605.10543*, 2026. URL https://arxiv.org/abs/2605.10543.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. URL https://arxiv.org/abs/1812.01717.

Wan-AI. Wan2.2-TI2V-5B. Official model card, 2025. URL https://huggingface.co/Wan-AI/Wan2.2-TI2V-5B.

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. URL https://arxiv.org/abs/2606.28016.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. InternVideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/11491_ECCV_2024_paper.php.

Ziyi Wu, Aliaksandr Siarohin, Willi Menapace, Ivan Skorokhodov, Yuwei Fang, Varnith Chordia, Igor Gilitschenski, and Sergey Tulyakov. Mind the time: Temporally-controlled multi-event video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Wu_Mind_the_Time_Temporally-Controlled_Multi-Event_Video_Generation_CVPR_2025_paper.pdf.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5e5fd18f863cbe6d8ae392a93fd271c9-Abstract-Conference.html.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. LongLive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025. URL https://arxiv.org/abs/2509.22622.

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. URL https://arxiv.org/abs/2511.20649.

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*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yin_One-step_Diffusion_with_Distribution_Matching_Distillation_CVPR_2024_paper.html.

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. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yin_From_Slow_Bidirectional_to_Fast_Autoregressive_Video_Diffusion_Models_CVPR_2025_paper.html.

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. doi: 10.52202/085713-1024. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/2bde8fef08f7ebe42b584266cbcfc909-Abstract-Conference.html.

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. URL https://proceedings.mlr.press/v267/zhao25m.html.

APPENDIX CONTENTS

A Agent-based recaptioning and timeline construction **13**

A.1 Narrative decomposition and visual detail 13

A.2 State consistency and timeline variants 14

A.3 Export, validation, and review 14

B Interval-conditioned backbone and temporal translation **14**

B.1 Network architecture 14

B.2 Interval encoding and common temporal translation 15

C Training and long-rollout inference **16**

C.1 Training corpus and splits 16

C.2 ODE initialization 16

C.3 Self-Forcing 17

C.4 Long-rollout inference and text encoding 17

D Datasets and evaluation details **17**

D.1 Corpus inventory and evaluation inputs 17

D.2 Temporal-Following Score protocol 18

D.3 Score aggregation 18

D.4 Temporal-Following Score breakdown 19

E Human evaluation protocol **19**

E.1 Stimuli and scene selection 19

E.2 Anonymous presentation and viewing procedure 19

E.3 Evaluation criteria and response options 20

E.4 Scoring and the reported sample 20

F Complete timeline and attention diagnostic **21**

Appendices A–C describe timeline construction, the backbone, and training and inference. Appendix D gives evaluation details and component results for the main comparison. Appendix E describes the human-study stimuli, interface, ranking criteria, and analysis.

A AGENT-BASED RECAPTIONING AND TIMELINE CONSTRUCTION

Before video generation, a language-model agent constructs the conditions in Section 2.1 using the `temporal-action-decomposition` skill. Given a source narrative, target duration, and number of variants, it produces persistent scene context, participant identities, and subject-specific event timelines.

A.1 NARRATIVE DECOMPOSITION AND VISUAL DETAIL

The agent identifies core actions, participants, required outcomes, causal order, and intended concurrency. Stable participant identifiers connect identity descriptions to dynamic events, while scene

information remains separate. Core events must be traceable to the narrative's roles and actions, but may be rephrased. Richer source annotations provide provenance, not permission to add events or participant storylines absent from the narrative. Group membership, reflections, and camera motion require explicit interpretation: annotation tracks do not automatically represent distinct people or independent physical motion.

Compatible appearance, spatial relations, posture, object contact, preparation, and recovery may clarify how an action unfolds, provided they preserve the core action, its order, and outcome. Descriptions must match the viewpoint and visible body parts. Scene and identity entries contain persistent information; initial poses and object locations are marked as initial conditions, with subsequent changes assigned to events. A full-video condition must not assert a state that a later event changes.

A.2 State consistency and timeline variants

Before assigning intervals, the agent checks object ownership, hand availability, contact, and bodily support across events. Handovers require compatible giving and receiving actions; sustained contact constrains relative motion. A participant's concurrent actions must use compatible body states without competing for the same resource. Intervals reflect these dependencies and plausible durations for preparation, execution, and recovery. Events need not have equal lengths, and sustained actions need not be split without a meaningful change. Participants share one timeline with $0 \leq s_i < e_i \leq T$; interacting tracks must agree on contact and transfer windows.

Variants retain core events, order, outcomes, and explicit identity constraints while varying unspecified pacing, visual details, and compatible intermediate steps. Modest timing changes may use a shared, order-preserving time map across tracks to preserve boundary relationships. Large duration changes require reassessing whether actions and prerequisites fit; uniform compression does not establish plausible motion. Every variant, including the least detailed, must remain internally consistent. Different identifiers or timing perturbations alone do not establish meaningful semantic variation.

A.3 Export, validation, and review

The agent authors and revises semantic content; scripts handle temporal transformations, serialization, and structural checks. The representation stores scene context and participant records with identities, appearance intervals, and event timelines. The TIE interface uses aligned start, end, and text arrays: array position associates fields, not chronological order. Export preserves authored descriptions under the consumer's interval and text-capacity conventions. Text encoding settings appear in Appendix C.4.

The supplied `check_tie.py` checks nonempty captions, finite valid intervals, matching array lengths, and unique video identifiers when present. For hierarchical records, events must lie within their participants' appearance intervals; optional checks enforce the requested duration and record count. These format checks are separate from the agent's semantic review of coverage, identity and object continuity, causal order, compatible overlap, and agreement between text and intervals. Review reads back the final export and records the versions and coverage inspected. Model-assisted review and structural validation are recorded separately from independent human or source-video verification; neither alone establishes adherence in generated videos.

B Interval-conditioned backbone and temporal translation

This section expands the interval representation in Section 2.2: the network architecture identifies where temporal conditioning enters, and the derivation establishes the fixed-feature translation property used in the main paper.

B.1 Network architecture

Figure 4 expands the interval-conditioned cross-attention in Equation (1). The diagram shows the interval-conditioned backbone (Shu et al., 2026) used by our bidirectional teacher. The autoregres-

sive student retains these cross-attention paths while using block-causal video self-attention and cached visual history, as described in Section 2.

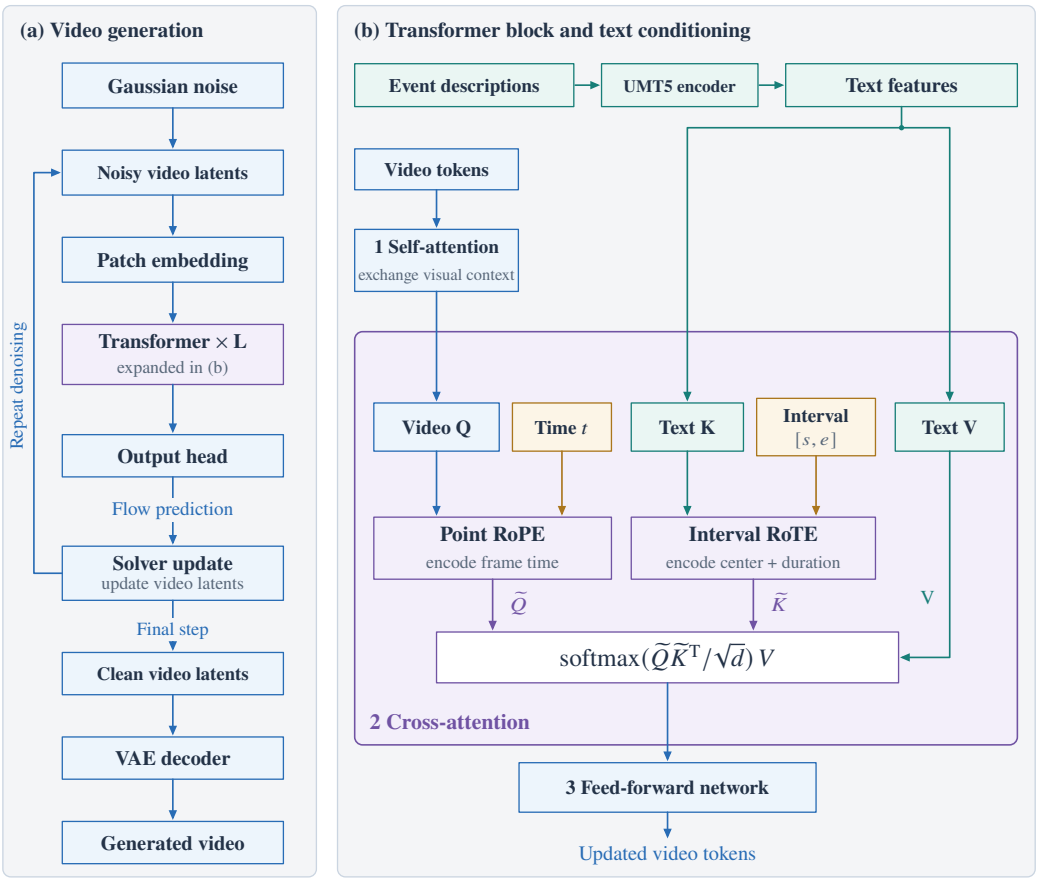


Figure 4: **Interval-conditioned network architecture.** (a) A stack of L transformer blocks predicts the flow used to update noisy video latents. The denoising loop repeatedly applies this network; the VAE decoder converts the final clean latents into a video. (b) Each block applies video self-attention, interval-conditioned cross-attention, and a feed-forward network. UMT5 encodes event descriptions into text features. The video query projection Q receives point-time RoPE, and the text key projection K receives interval RoTE, which encodes the interval center and duration. The text values V do not receive this temporal transform. Here $\widetilde{Q}$ and $\widetilde{K}$ denote the encoded queries and keys, and d is the attention head dimension. Gold boxes denote physical-time metadata; this time axis is distinct from the diffusion noise level. Normalization, residual connections, output projections, and noise-level conditioning are omitted for clarity.

B.2 Interval encoding and common temporal translation

This section gives the definition and proof used in Corollary 1, following TIE's closed-form encoder (Shu et al., 2026, Section 3.2 and Appendix A.1). For one attention head of even width d, let $\mathbf{q}, \mathbf{k} \in \mathbb{R}^d$ be content features before temporal encoding, and let $c = (s+e)/2$ and $r = (e-s)/2$ for event interval $[s, e]$. Write $R(t)$ for the block-diagonal rotary matrix with frequency-pair angles $\omega_m t$, absorbing the fixed time scale into ω_m. The normalized duration modulation is

$$D(r) = \operatorname{diag}\left(\frac{\operatorname{sinc}(\omega_1 r)}{C_r} I_2, \ldots, \frac{\operatorname{sinc}(\omega_{d/2} r)}{C_r} I_2 \right),$$

$$C_r = \frac{2}{d} \sum_{m=1}^{d/2} \operatorname{sinc}(\omega_m r),$$

where $\mathrm{sinc}(x) = \sin(x)/x$, extended continuously at zero, and $C_r \neq 0$. TIE encodes the query and event key as $R(t)\mathbf{q}$ and $R(c)D(r)\mathbf{k}$, respectively.

Proof. Rotary matrices satisfy $R(a)^\top R(b) = R(b-a)$. Hence

$$S(\mathbf{q}, \mathbf{k}; t, [s, e]) = d^{-1/2}\mathbf{q}^\top R(c-t)D(r)\mathbf{k}.$$

A common shift changes c and t to $c + \Delta$ and $t + \Delta$, while leaving r unchanged. Both $c - t$ and $D(r)$ are therefore preserved, which proves Equation (5). ∎

C TRAINING AND LONG-ROLLOUT INFERENCE

We detail the training corpus in Section 3.1 and expand the short-sequence training pipeline and rolling inference in Section 2.3. ODE initialization transfers the bidirectional teacher to a causal student; Self-Forcing then trains on self-generated histories. The final subsection specifies how the trained generator consumes a rolling view of a longer event timeline.

C.1 TRAINING CORPUS AND SPLITS

The prompt-only training corpus contains 20,000 structured source scenes. Of these, 15,000 are composed from controlled scene, participant, object, and action banks; 5,000 retain selected PexelsEvents (Shu et al., 2026) scene facts with rewritten descriptions. Each source has stable participant identifiers and an aligned event graph. Five temporal variants use uniform, early-weighted, late-weighted, and two alternating duration allocations. Participant-specific offsets and endpoint adjustments create staggered overlaps on a quarter-second grid while preserving source ordering and marked concurrency. Self-Forcing uses all 100,000 schedules in the 10-second export. These are generation conditions, not paired real videos. The separate 5-second export is constructed independently, rather than by rescaling the 10-second timelines, and is not used in this configuration. Descriptions are concatenated before UMT5 (Chung et al., 2023) encoding, with a 512-token limit.

ODE initialization uses teacher-generated trajectories for one variant from each of 2,000 distinct source scenes; another 64 scenes form its validation subset. This split is scene-disjoint within ODE initialization, but the 64 scenes are not held out from the full Self-Forcing prompt corpus.

C.2 ODE INITIALIZATION

We use a pretrained bidirectional TIE model as a frozen teacher. Its video self-attention can access the complete latent sequence. The student starts from the same weights, retains the interval conditioning in Section 2.2, and replaces video self-attention with block-causal attention: a block accesses itself and preceding blocks, but not future video blocks. The full event timeline for the short training sequence remains available during this initialization stage.

At 16 fps, a 10-second training sequence contains 41 latent frames, partitioned into causal blocks $[1, 8, 8, 8, 8, 8]$. TIE uses time scale $\gamma = 4$ and temporal stride four. At frame rate f, latent cells are $[0, 1/f]$ for $j = 0$ and $[(4j-3)/f, (4j+1)/f]$ for $j > 0$; their centers define τ_j.

For each selected prompt and Gaussian initial latent, the teacher generates an ODE trajectory with classifier-free guidance. We retain trajectory states at the student's denoising levels, together with the clean endpoint. Write the resulting trajectory as $\{\mathbf{z}_\sigma^{\mathrm{T}}\}$ and its final clean endpoint as $\mathbf{z}_0^{\mathrm{T}}$. For each causal block b, independently sample one of these states, assemble the selected blocks into $\widetilde{\mathbf{z}}$, and predict the shared clean endpoint:

$$\mathcal{L}_{\mathrm{ODE}} = \mathbb{E}\left[\frac{\sum_b m_b \left\|\mathbf{D}_{\theta,b}(\widetilde{\mathbf{z}}, \boldsymbol{\sigma}, \mathcal{C}) - \mathbf{z}_{0,b}^{\mathrm{T}}\right\|_2^2}{\sum_b m_b n_b}\right], \qquad m_b = \mathbb{1}[\sigma_b > 0]. \tag{6}$$

Here $\boldsymbol{\sigma} = (\sigma_b)_b$ collects the sampled noise levels, $\mathbf{D}_\theta$ predicts clean latents, and n_b counts scalar entries in block b. The expectation excludes batches with no active entries. Clean blocks may supply context but contribute no regression loss. Thus, this stage learns endpoint prediction from mixed trajectory states; it does not regress the next ODE state or require clean context for every preceding block. We update all student parameters during initialization.

Teacher trajectories use a dense 50-step Euler schedule, flow shift five, and classifier-free guidance scale five. The student levels $\{1, 0.9375, 5/6, 0.625, 0\}$ are inserted into the integration grid.

C.3 Self-Forcing

Following Self-Forcing (Huang et al., 2025), we next generate causal blocks conditioned on the student's own preceding outputs, stored in a KV cache. Training selects a denoising step at which to compute the learning signal. Gradients pass through the prediction at that step; earlier denoising operations and cached historical states are detached. This exposes the student to its own generated contexts while limiting the retained computation graph. The generator and an auxiliary score model use LoRA adapters; the bidirectional real-score teacher remains frozen.

Let $\widehat{\mathbf{z}}$ denote the generated video latent and $\mathbf{z}_\sigma = (1-\sigma)\widehat{\mathbf{z}} + \sigma\boldsymbol{\epsilon}$ its noised version. The frozen teacher and trainable fake-score model predict clean latents $\mathbf{D}_{\mathrm{T}}$ and $\mathbf{D}_\phi$ from the same $(\mathbf{z}_\sigma, \sigma, \mathcal{C})$. The teacher prediction uses classifier-free guidance. Following the score-difference principle of distribution matching distillation (Yin et al., 2024), our Self-Forcing implementation uses

$$\begin{aligned} \mathbf{g}_b &= \frac{\mathbf{D}_{\phi,b} - \mathbf{D}_{\mathrm{T},b}}{s_b}, \\ \mathcal{L}_{\mathrm{DMD}} &= \frac{1}{2}\,\mathrm{MSE}(\widehat{\mathbf{z}}, \mathrm{sg}[\widehat{\mathbf{z}} - \mathbf{g}])\,. \end{aligned} \tag{7}$$

The operator sg stops gradients, and s_b is a blockwise normalization factor specified below. The fake-score model is trained separately with flow-matching target $\boldsymbol{\epsilon} - \mathrm{sg}[\widehat{\mathbf{z}}]$ on generated samples. Generator updates use deterministic blockwise replay to propagate the stopped DMD direction without retaining the entire rollout graph. This objective trains the student toward the teacher's conditional video distribution.

The full-parameter step-2000 student initializes Self-Forcing. The generator and auxiliary score model then use rank-128 LoRA adapters. Training samples an exit among four denoising positions. For Equation (7), the first latent uses $s_1 = 1$, while subsequent blocks use $s_b = \max(\mathrm{mean}\,|\widehat{\mathbf{z}}_b - \mathbf{D}_{\mathrm{T},b}|, 10^{-6})$.

C.4 Long-rollout inference and text encoding

At inference, we use a temporal margin of $\rho = 4$ seconds. A regular six-latent block at 16 fps with temporal stride four spans 1.5 seconds, giving a 9.5-second conditioning window away from the video endpoints. To accommodate UMT5's 512-token input limit, event selection precedes text encoding. We repack persistent descriptions with the selected actions in their source order, so excluded events do not enter the current text representation. Text embeddings are cached for repeated event subsets. Selected actions retain their original intervals on the global timeline.

D Datasets and evaluation details

This section documents the conditions and scoring procedure used in Section 3.1, followed by the component scores behind the reported full-video results.

D.1 Corpus inventory and evaluation inputs

Table 3 summarizes the short-sequence training corpus and long video evaluation set described in Section 3.1.

Evaluation source selection. The benchmark uses a fixed source-selection seed of 20260925. Participant counts are stratified into one, two, three, four, and at least five tracks, with quotas $(17, 10, 3, 1, 1)$ for the 32 evaluation scenes. A group or camera track counts as one structured participant, so this count need not equal the number of people. The captions and event plans are authored adaptations, with no independent video-grounded verification; participant-count stratification does not establish matched action or scene distributions.

Table 3: Dataset inventory. Counts refer to prompt conditions, not real videos or completed model evaluations. The ODE subsets are drawn from the Self-Forcing corpus. Evaluation scenes are adapted from training-prompt references.

Use	Source scenes	Conditions	Duration (s)
Self-Forcing corpus	4,000	20,000	10
ODE initialization	2,000	2,000	10
ODE validation	64	64	10
Main comparison	32	128	60

Event records and conditioning-input budgets. Each record contains scene context, participant identities, and visible events with start/end times on a 0.25-second grid. Only the marked action conditions are scored; persistent scene and appearance descriptions are context. The main set averages 10.13 action conditions per video.

Our method's conditioning inputs are checked against the 512-token limit in every rolling 10-second window, using the UMT5 tokenizer and the backbone's text cleaning. This is a local-window constraint, not a 512-token bound on the complete 60-second plan.

Native single-prompt baselines require the entire serialized timeline to fit 512 tokens. Where needed, their descriptions are compressed before encoding so that the input fits without truncation. Compression preserves event intervals but can remove appearance and action details. Thus, the inputs share the schedule but do not have an identical descriptive budget. All methods use the same evaluation references and sampling protocol.

D.2 Temporal-Following Score protocol

We use TempAct's full-video, framewise rubric and weights (Wang et al., 2026), with Qwen3-VL-8B-Instruct (Qwen Team, 2025) revision `0c351dd0` (the full revision is pinned in the evaluator). The six scores are normalized from $[0, 10]$ to $[0, 1]$ and combined as

$$\mathrm{TFS} = 0.2\overline{F} + 0.3G + 0.2O + 0.1P + 0.1C + 0.1A, \tag{8}$$

where $\overline{F}$ is the mean frame-level score, G the global assessment score, O temporal order, P physical plausibility, C visual consistency, and A prompt alignment.

For a video with n decoded frames, the 40 indices are

$$k_j = \min(j \max(\lfloor n/40 \rfloor, 1), n-1), \qquad j = 0, \ldots, 39.$$

The judge receives the complete timed reference and a numbered map from each sampled frame to its time k_j/f, where f is the decoded frame rate. For 961 frames at 16 fps, these samples cover 0–58.5 seconds at 1.5-second increments; the final frame is not sampled. Sparse observations can miss short actions, so this score cannot establish subsecond boundary accuracy. This protocol retains the original rubric and weights while changing TempAct's original 6/10-frame profiles to 40 frames and adding the explicit mapping from sampled frames to the video timeline. It does not reproduce the original benchmark sampling setting.

Frames retain their resolution and are encoded as RGB JPEGs at quality 85. The evaluator uses one vote, temperature zero, top-p one, and thinking enabled. The current deployment uses BF16/SDPA and a 16,384-token generation cap. Invalid JSON, missing scores, an incorrect number of frame scores, failed decoding, or a truncated response are evaluation errors, not zero-quality videos.

D.3 Score aggregation

For event metrics, we first average within each event and then equally over the scored events of a video. For each metric, repeated generation seeds are averaged within a prompt, and prompts are averaged equally. Equal numbers of variants also give equal source-scene weight. Missing or failed evaluations remain missing; the complete benchmark mean is not reported from a model-dependent subset.

D.4 Temporal-Following Score breakdown

Table 4 reports the six normalized components of TFS for the main comparison. The global assessment component G belongs to the TFS rubric and is distinct from the VideoReward-Event Overall score in Table 1. LongLive leads all six components; our model is second on five, while Self-Forcing is second on temporal order. This breakdown reinforces the distinction in the main comparison: stronger event-level visual quality does not imply stronger full-video timeline following.

Table 4: Component breakdown of Temporal-Following Score (TFS) on the 128-condition benchmark of 60-second event timelines. Entries are normalized component scores in $[0, 1]$ before weighting; the column headers give their weights in Equation (8). Higher is better in every column. The best and second-best scores are bold and underlined, respectively.

Method	**Frame-level mean** $\overline{F}$ (20%)	**Global assessment** G (30%)	**Temporal order** O (20%)	**Physical plausibility** P (10%)	**Visual consistency** C (10%)	**Prompt alignment** A (10%)
Self-Forcing	0.5263	0.2680	<u>0.3617</u>	0.3727	0.3664	0.2664
LongLive	**0.6236**	**0.3297**	**0.3742**	**0.4258**	**0.4297**	**0.3250**
LongLive Interactive	0.5189	0.2625	0.3016	0.3125	0.3102	0.2586
OnTime (Ours)	<u>0.5442</u>	<u>0.2852</u>	0.3547	<u>0.3867</u>	<u>0.3938</u>	<u>0.2805</u>

E Human evaluation protocol

This section describes the human evaluation underlying Table 2. Each trial presents three anonymous videos of the same scene and collects a separate ranking for each of five criteria. The study compares OnTime, Self-Forcing, and LongLive; LongLive Interactive is not included.

E.1 Stimuli and scene selection

We use ten distinct source scenes from the main benchmark, with one timing variant per scene and one existing generation per method. Each stimulus is the first 40 seconds of its 60-second generation, giving 30 clips at 16 fps. All clips retain their source resolution and aspect ratio and use the same encoding settings. The underlying systems retain their benchmark conditioning interfaces and checkpoints, as described in Appendix D; this study does not isolate the effect of RTC from other system differences.

Scene selection uses available automatic evaluations and visual inspection, with content screening for nudity. It is a purposively selected subset, not a random sample of the benchmark. Trials are organized into two fixed blocks of five scenes, with scene order shuffled independently within each block. Each scene appears once per session; there are no hidden repeated trials.

E.2 Anonymous presentation and viewing procedure

The interface labels the methods A, B, and C and withholds method names, original filenames, and other participants' answers. For each scene, a random base permutation of the methods is cyclically rotated across consecutive enrolments. Thus, each group of three consecutive enrolled sessions assigns each method to each position once for that scene. This balances assignments, but attrition can leave completed ratings unbalanced.

All three videos share synchronized playback, pause, replay, and seeking at normal speed; individual videos can be enlarged. The scene and participant descriptions accompany the videos. Original English action prompts appear as subtitles during their requested intervals, and the complete interval list for the excerpt is available. Instructions and response controls are in Chinese. Events beginning at or after 40 seconds are omitted, while intervals crossing the cutoff retain their original endpoints and are marked as continuing beyond the excerpt. Participants judge only the visible prefix and are not asked to require such actions to finish by 40 seconds.

Participants are instructed to watch all three clips completely and confirm that they have inspected them before submitting. Submission requires at least 39.5 seconds of unique, visible playback per

video; seeking and replaying previously viewed intervals do not increase coverage. This check measures playback rather than visual attention. There are no catch trials or repeat-based attention exclusions.

Participation is voluntary, and no names or contact information are requested. Participants may stop at any time; completed responses from partially finished sessions are retained.

E.3 Evaluation criteria and response options

Table 5 gives the five criteria, translating the participant-facing instructions into English. Overall is presented first, followed by Event, Timing, Continuity, and Quality. Each criterion is answered independently, so the preferred ordering may differ across criteria.

Table 5: Criteria used in the human study, in questionnaire order.

Criterion	Instruction
Overall	Rank the videos by overall preference, considering both the requested scene/actions and visual quality.
Event	Ignoring exact timing, compare whether the correct subjects perform the requested actions. Consider missing actions, incorrect actions, and actions performed by the wrong subject.
Timing	Compare approximate action onsets, offsets, durations, order, and overlap against the requested intervals. Frame-by-frame measurement is not required.
Continuity	Compare the stability of people, objects, and the scene, and the naturalness of transitions. Consider extra or missing people, sudden appearances/disappearances, identity changes, and abrupt object or background changes.
Quality	Ignoring prompt agreement, compare visual fidelity and naturalness of motion, including deformation, flicker, blur, and abnormal movement.

Participants select A/B/C from best to worst, with no preselected answer. They can revise the ordering, tie adjacent positions, or declare all three tied. A tie means that the videos can be compared but no preference is expressed; *cannot judge* is a separate option for insufficient information. Every criterion requires either a complete ranking containing each video exactly once or a cannot-judge response. Incomplete or duplicate-label rankings are rejected. Optional comments can identify the events or time ranges that influenced the judgment.

A separate technical-failure option skips the entire trial and requires a reason, such as a file failing to load or decode. Poor generated content, including blur, distortion, or unnatural motion in a playable file, remains eligible for evaluation. Technical failures are not encoded as ties or low scores.

E.4 Scoring and the reported sample

For criterion c, let $\mathcal{V}_c$ contain the submitted scene–session groups with a valid ranking. For method m in group i, let r_{imc} be its occupied rank, using the average occupied position for a tie. The normalized ranking score and its reported mean are

$$q_{imc} = 50(3 - r_{imc}), \qquad \overline{q}_{mc} = \frac{1}{|\mathcal{V}_c|} \sum_{i \in \mathcal{V}_c} q_{imc}. \tag{9}$$

Thus, first, second, and third place receive 100, 50, and 0 points. A tie for first gives 75 points to each tied method and 0 to the third; a tie for second gives 25 points to each tied method; an all-method tie gives 50 points each. Cannot-judge responses are excluded only for their criterion, and technical failures are excluded for all criteria. Valid scene–session groups receive equal weight, so sessions with more completed trials contribute more observations.

Pairwise preferences between OnTime and each baseline follow from their relative positions in the same ranking and are not separately elicited votes. Table 2 reports the three-method ranking score in Equation (9).

The snapshot underlying Table 2 contains seven enrolled anonymous sessions. Three sessions submitted 28 scene responses: 27 ranking responses covering all ten scenes and one technical failure.

Two sessions completed all ten assigned trials. There are three cannot-judge responses for Event and four for Timing, leaving 24, 23, 27, 27, and 27 valid groups for Event, Timing, Continuity, Quality, and Overall, respectively.

These are descriptive results from a small, selected sample. Anonymous browser sessions are not verified distinct people. Responses from the same session and responses to the same scene are dependent; the five criterion answers and derived pairwise outcomes are not independent raters. We report no confidence intervals or significance tests, and do not infer benchmark-wide or population-wide superiority. Timing rankings measure perceived adherence to the requested intervals; they do not replace direct annotations of action boundaries.

F Complete timeline and attention diagnostic

Figure 5 expands Figure 3 with fixed crops of the principal person at 15.00, 15.75, and 16.75 seconds. The principal person in OnTime raises his gaze toward the upper right across the sampled frames; Self-Forcing keeps the head lowered, while LongLive turns from a rightward head pose toward a lowered chin. These labels describe visible head and gaze states, not the correctness of the complete pen action or the full scene. The timeline and attention panels use exactly the same data and color scales as the main-paper figure. This is the same auxiliary 30-second diagnostic, separate from the 60-second benchmark; the cross-model attention magnitudes are not a controlled ablation.

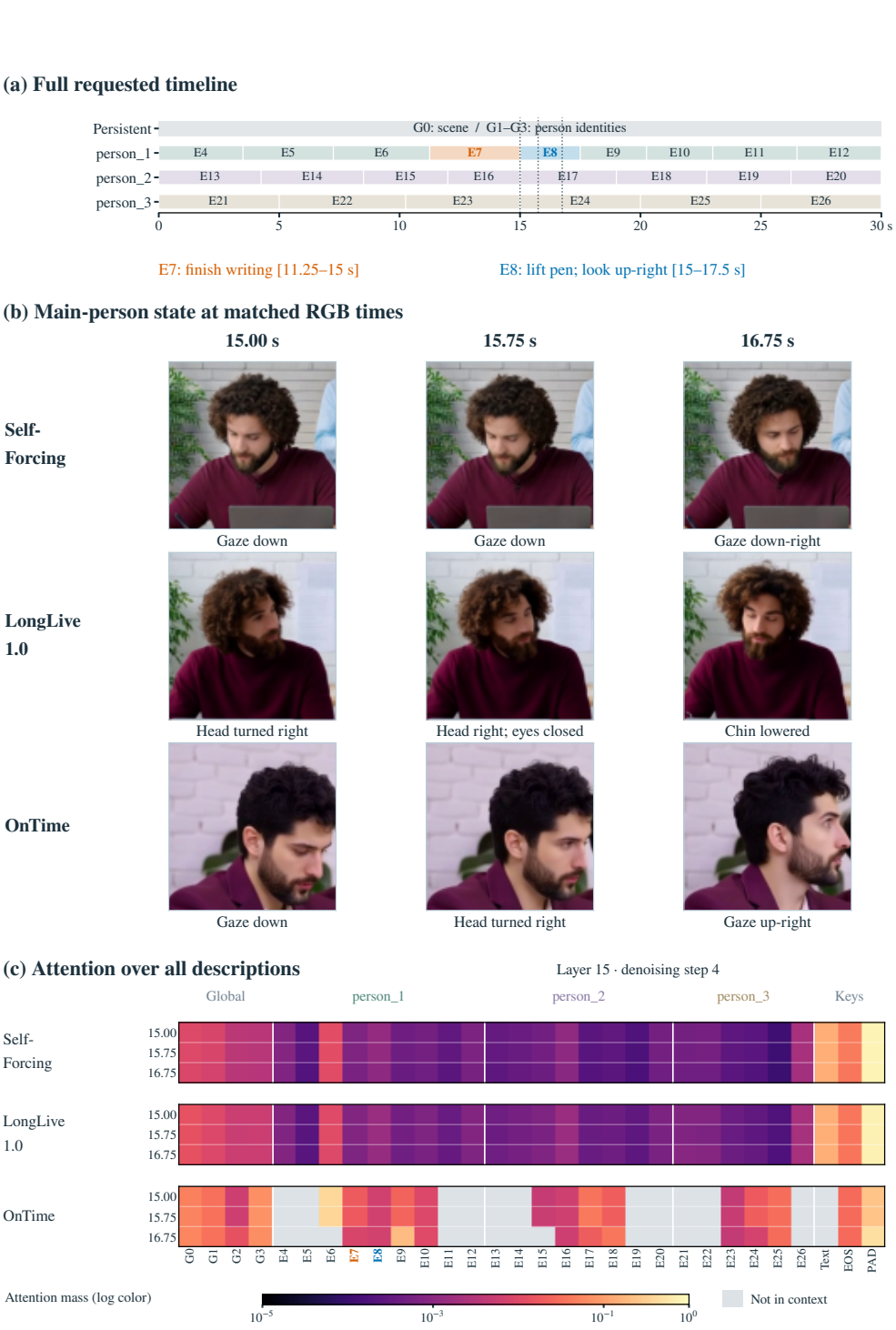


Figure 5: **Complete timeline, subject-state, and attention diagnostic.** (a) Requested timeline; dotted lines mark sampled RGB times. (b) Fixed person crops for Self-Forcing, non-interactive LongLive 1.0, and OnTime. (c) Layer 15 (zero-based), fourth denoising forward: attention summed over description tokens and averaged over all heads and full-frame queries. Colors use a shared log scale with all-key normalization; gray denotes descriptions absent from the input context.