ReGen: Agentic Video World Modeling with Synergized Reasoning and Generation

Chenguo Lin[1], **Yu Tang**[2], **Weiqiao Zheng**[2], **Enhua Jiang**[2], **Zhiguang Liu**[2], **Maomao Li**[2], **Muxi Chen**[2], **Xiaojia Chen**[2], **Shisheng Huang**[2], **Jianhuan Zhuo**[2], **Zilin Yang**[2], **Jiarong Ou**[2], **Rui Chen**[2], **Yadong Mu**[1]
[1]Peking University, [2]Tencent Hy
https://chenguolin.github.io/projects/regen

Abstract

Video world models aim to simulate environmental dynamics under interaction. However, most existing approaches entangle transition reasoning with pixel-level generation within a single model, thereby limiting narrative continuity and long-horizon coherence. We reformulate video world modeling as an **agentic process that synergizes reasoning and generation**, and present **REGEN**, which couples a streaming vision-language planner with a causal diffusion generator. At each step, the planner infers a **structured action-state specification** from history, while the generator realizes it as the next video segment. The planner further verifies the outcome and decides to continue, end, or re-generate, forming a closed-loop sense-plan-act-verify pipeline. To support this formulation, we carefully construct **a large-scale action-grounded video dataset with over 2M segments** that instantiates the planner-generator interface, and introduce **action-grounded reward alignment** that turns the action specification into an enforced reward and mitigates semantic drift across segments. Experiments show that REGEN improves long-horizon narrative coherence over prior work, supporting autonomous long-horizon rollouts without step-by-step user intervention.

1 Introduction

Video world models aim to simulate how visual environments evolve over interactions, with applications in embodied intelligence, autonomous driving, and interactive content creation [1, 2, 3, 4, 5, 6, 7, 8]. A dominant line of recent progress treats this problem as long video generation [9, 10, 11, 12, 13], where a diffusion transformer (DiT) [14] rolls out frames autoregressively from an initial observation and user interactions [15, 16, 17, 18]. However, as horizons grow, this forces a single model to solve two problems at once: **reasoning what happens next** and **generating how it appears visually**.

This monolithic paradigm has at least three inherent limitations. First, causal reasoning about state transitions is compressed into implicit pattern matching over pixels, far exceeding what a diffusion model is trained for [19, 20, 21, 22, 23]. Meanwhile, prior works generally operate under fixed initial conditioning [9, 10, 24, 25] or treat changing text prompts as interaction signals [15, 17, 18] rather than autonomous planning, which severely limits narrative dynamics and often leads to first-frame sticking and repetitive actions [26, 27, 28]. Most critically, the entire generation pipeline is open-loop: without an explicit transition plan to check against, there is no mechanism to detect drift or recover from accumulated errors, leaving the rapidly growing reasoning capabilities of large language models (LLMs) and vision-language models (VLMs) [29, 30, 31, 32, 33] untapped.

In this work, we approach video world modeling as an **agentic process** rather than a pure generation task. As shown in Figure 1, a capable world model should not merely generate pixels, but also

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

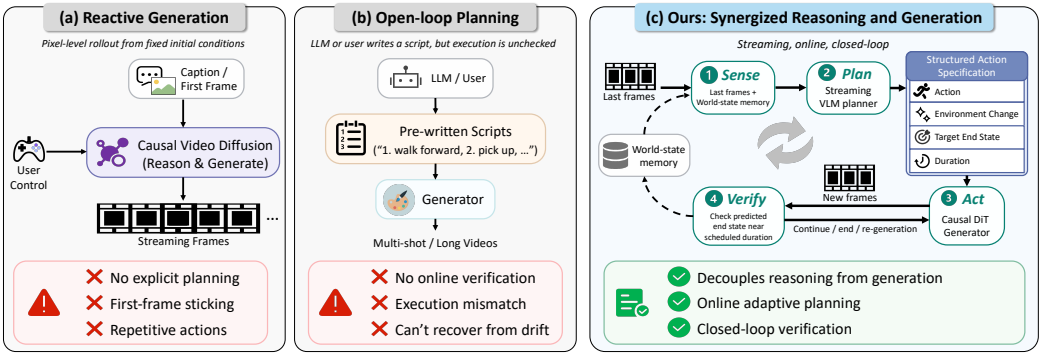

Figure 1: **Agentic World Modeling**. (a) **Reactive generation** [15, 17, 12, 25] rolls out pixels from a fixed initial condition and user interactions without reasoning. (b) **Open-loop planning** [34, 35, 36, 37] commits to a pre-written script without online verification. (c) **Ours** closes the loop, coupling a streaming VLM planner with a causal DiT via structured action specifications checked at every step.

repeatedly (1) sense the current world state, (2) plan the next transition, (3) simulate the corresponding event, and (4) verify whether the result matches the intended transition.

Based on this perspective, we present **REGEN**, **an agentic framework that synergizes reasoning and generation** for video world modeling. It consists of a streaming VLM as the underlying planner and a causal DiT as the generation engine. The VLM maintains a compact world-state memory as context, and at each step, issues a **structured action specification**, including (1) the next intended action, (2) the expected environmental change, (3) the target end state, and (4) the predicted duration. The DiT realizes this specification causally, and its output is fed back to the VLM for inline verification near the predicted duration, deciding to *continue*, *end*, or *re-generate* based on whether the simulated visual content matches the contracted end state, so that the next segment is planned from what was *actually* observed, closing the *sense-plan-act-verify* loop.

To support this formulation, we carefully curate a large-scale **action-grounded video dataset** with segment-level annotations, including subjects driving world evolution, action sequences, environmental changes, and corresponding end states, that standard video datasets [38, 39, 40, 41, 42] do not provide. These annotations instantiate the shared protocol between planner and generator: the VLM is trained to predict them and the DiT to realize them, aligning their interfaces by design. We further tighten this coupling with an **action-grounded reward alignment** stage that uses action specifications as a structured reward target, turning the instructions from a soft textual hint into an enforced reward and reducing semantic drift across segments.

Quantitative evaluations show that REGEN improves over strong long-video generation baselines on long-horizon generation fidelity and semantic narrative continuity. Each essential component of our method is verified through ablation studies.

Our contributions can be summarized as follows:

- We reformulate video world modeling as an **agentic process** and propose REGEN, a closed-loop framework coupling a streaming VLM planner with a causal DiT generator.
- We design a **structured action specification protocol** as a shared planner-generator interface, and curate an **action-grounded dataset** with 400K videos and 2M annotated segments.
- We introduce **action-grounded reward alignment**, optimizing generator outputs against planner-consistent verification signals to enforce the contract and reduce semantic drift across segments.
- Experiments show that REGEN improves long-horizon fidelity and narrative coherence, supporting coherent autonomous long-horizon rollouts without step-wise guidance.

2 Related Work

2.1 Video World Models

Video world models predict visual evolution under specified conditions through video generation. A dominant line targets long-horizon generation, where causal diffusion [43, 44, 9, 45] and follow-up

methods based on more sophisticated context modeling [10, 24, 46] and scaling up model size [11, 12] have extended coherent generation to multiple minutes, and efficient streaming architectures [47, 48, 49, 50] further enable real-time inference. Some of these methods support interactive prompt switching during generation [15, 17, 51], allowing users to steer narrative direction on the fly. However, narrative evolution still relies on user-supplied prompts or a static initial condition.

Building on these generation capabilities, interactive world models go a step further by conditioning on low-level actions to simulate environment dynamics. Early work demonstrates playable game environments at interactive rates [52, 53, 54], and subsequent efforts extend the paradigm to open-ended worlds and real-time generation [2, 55, 56, 13, 57], robotic and embodied simulation [58, 7, 6, 8], and large-scale foundation models pretrained on massive video corpora [59, 60, 25]. Despite rapid scaling in both lines of work, existing methods largely adopt a monolithic design in which a single model implicitly handles both state reasoning and pixel synthesis, without a dedicated module for causal planning or verification.

2.2 LLM/VLM-Guided Visual Generation

LLMs and VLMs have been increasingly adopted as planners for visual content creation. Early methods use an LLM to decompose a prompt into multi-scene plans, such as spatial layouts, pose sequences, or shot scripts, and execute generation in a single forward pass [61, 62, 63, 64]. These pipelines are open-loop, as the planner commits to an entire plan before any frame is produced, with no mechanism to detect or correct deviations during generation. More recently, multi-agent frameworks have introduced iterative refinement for cinematic storytelling and multi-modal production [65, 66, 67], knowledge-grounded and skill-augmented generation [68, 69], and planner-generator loops with self-improvement or verifier-triggered re-generation for video synthesis [70, 36, 37], yet in these works the plan is largely written upfront from a goal or prompt, rather than conditioned step by step on the frames the generator actually produces, and the video backbone is treated as a plug-and-play executor without joint architectural or efficiency co-design. Similarly, multi-shot video generation methods [71, 34, 35, 72] address cross-shot visual consistency through shared attention or context tuning, but focus on the generation infrastructure rather than autonomous planning and verification.

In embodied AI, the sense-plan-act loop is well established, from grounding LLM plans in physical affordances [73, 74] to closed-loop replanning from environment feedback and self-verifying agents [75, 76, 77]. REGEN brings this paradigm into video world modeling in a step-wise closed-loop form, where a VLM autonomously plans structured action specifications with predicted durations, a causal DiT executes generation, and the same VLM verifies outcomes inline before the next cycle.

3 Method

3.1 Agentic Framework Overview

Video world modeling aims to simulate the next visual observation V_t given the history $\mathcal{H}_{t-1}$ and action a_t. Most prior studies [2, 9, 15, 11, 12, 13, 25] formulate this problem as causal video generation, in which viewpoint controls or user-specified prompts are treated as "actions", and train an autoregressive DiT $p_\psi(V_t|\mathcal{H}_{t-1}, a_t)$. Rather than entangling transition reasoning with visual generation in a single model, we propose REGEN, which *externalizes high-level transition reasoning* into an explicit planner while leaving low-level dynamics to the DiT, via an agentic *sense-plan-act-verify* closed loop. As shown in Figure 2, REGEN is composed of a causal DiT generator $p_\psi(V_t|\mathcal{H}_{t-1}, s_t, a_t)$ (Section 3.2) and a streaming VLM planner $\pi_\theta(s_t, a_t|\mathcal{H}_{t-1})$ (Section 3.3):

$$p_{\theta,\psi}(V_t|\mathcal{H}_{t-1}) = \mathbb{E}_{(s_t,a_t)\sim\pi_\theta(\cdot|\mathcal{H}_{t-1})}\left[p_\psi(V_t|\mathcal{H}_{t-1}, s_t, a_t)\right], \tag{1}$$

where s_t denotes the next environment state corresponding to the action a_t. As detailed in Section 4, we introduce a structured action specification protocol, and define:

$$s_t \coloneqq (c_t^{\text{delta}},\ e_t,\ T_t), \tag{2}$$

where c_t^{delta} describes the change of the environment relative to the previous state, e_t denotes the resulting end state, and T_t specifies the action duration in physical time.

Starting from an initial condition $\mathcal{H}_0 \coloneqq (I_0, a_0)$, including the first frame I_0 and/or a global instruction a_0, the framework evolves autoregressively. At step t, the planner π_θ takes the history $\mathcal{H}_{t-1}$ and emits (s_t, a_t), while the generator p_ψ simulates the next observation V_t conditioned on them. An online verifier $\phi_\theta(\cdot|V_t, s_t, a_t)$, which shares the same weights with the planner π_θ,

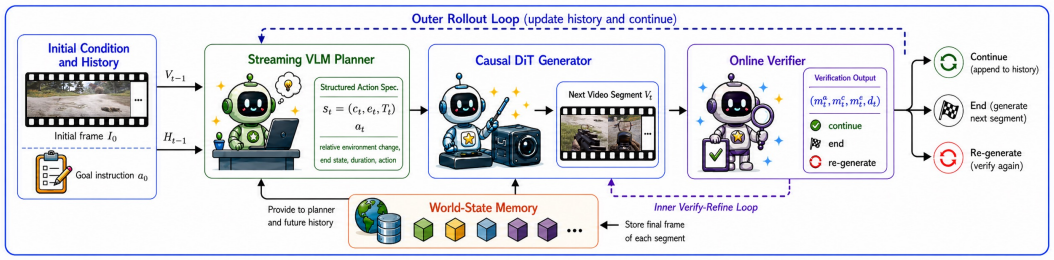

Figure 2: **Agentic Framework Overview**: a causal DiT generator (Section 3.2) couples with a streaming VLM planner (Section 3.3) in a verified loop for long-horizon video world modeling.

is additionally introduced to evaluate whether V_t satisfies (s_t, a_t), and provides a decision $d_t \in \{\text{CONTINUE}, \text{END}, \text{RE-GENERATE}\}$ around time T_t, enabling adaptive correction and preventing long-horizon drift. For re-generation, the planner first revises the failing fields of s_t via a REFINE prompt, providing the generator with a different conditioning signal rather than re-sampling under the same specification. Every retry restarts from the last accepted history, and rejected candidates are discarded without entering $\mathcal{H}_t$ or the KV cache, so failed attempts cost computation but rarely accumulate drift. To further align the generator with the planner's intent, we perform reinforcement post-training as described in Section 3.4. The complete rollout pseudocode is provided in Appendix B.1.

3.2 Causal DiT Generator

3.2.1 Action-grounded Conditioning

Segment-wise Cross Attention Unlike conventional DiT models [78, 20, 19, 23] that condition all video latents on a single global caption via cross-attention, REGEN treats each video as a sequence of action-defined segments, each associated with a structured caption. We impose a block-diagonal cross-attention mask so that latents within each segment attend only to their corresponding text embeddings, while temporal dependencies across segments are handled by self-attention. Each structured caption contains a shared global instruction and control object, together with the segment-specific environmental change c_t^{delta}, action label a_t, and resulting end state e_t.

Action Injection To further strengthen action grounding, we encode action labels via SigLIP [79], and inject them via segment-scoped FiLM modulation [80]. Specifically, for segment j, we apply

$$\mathbf{h}_j \leftarrow (1 + \boldsymbol{\gamma}_j) \odot \mathrm{LN}(\mathbf{h}_j) + \boldsymbol{\beta}_j, \quad (\boldsymbol{\gamma}_j, \boldsymbol{\beta}_j) = \mathrm{MLP}(\mathrm{SigLIP}(a_j)),$$

to the corresponding video latents after each self-attention block $\mathbf{h}_j$. These two text conditioning mechanisms serve complementary roles: templated structured captions preserve and leverage pre-trained text-conditioning capabilities, while explicit action injection provides disentangled control signals, similar to viewpoint or low-level action conditioning in previous works [11, 12, 25].

End State as Memory The generator p_ψ is a causal DiT, where previously generated clean frames are used as context for denoising the current ones. This design enables long-horizon extrapolation via sliding-window attention and attention sink [81, 15, 10], while KV caching significantly accelerates inference. However, sliding window limits accessible history and introduces forgetting. Prior work addresses this based on viewpoint coverage [82, 83, 84] or sophisticated retrieval mechanisms [18, 34]. In contrast, our action-grounded formulation provides a simpler and more effective solution by retaining the final frame of each segment as memory $\mathcal{H}_{t-1}$. Since each segment terminates at a well-defined post-action world state, these frames serve as compact and semantically aligned summaries of past transitions, offering an efficient trade-off between memory and fidelity. More details about action-grounded conditioning are provided in Appendix B.2.

3.2.2 Distilled Causal Generation

Building on the above conditioning design, we distill a multi-step diffusion model into a few-step causal generator via self-forcing [9]. Unlike prior approaches [44, 9, 56, 11], we omit the ODE-based distillation stage. Instead, after action-driven supervised fine-tuning, a bidirectional DiT is trained with teacher forcing to learn causal attention, and is then directly reused as the self-forcing student, while the bidirectional one serves as the teacher and fake score model. This simplification eliminates

the need for constructing ODE pairs while maintaining strong performance. Moreover, because segment-wise conditioning is already applied during teacher training and is designed to be a relative transition, the student can smoothly generate across prompts without KV cache refreshing [15, 17]. Training details for the causal DiT generator are provided in Appendix B.2.

3.3 Streaming VLM Planner

Plan and Verify as a Unified Token Policy While the DiT generator supports action-conditioned streaming, requiring step-wise structured prompts is impractical. We instead target autonomous rollout: given an initial condition, the model plans, generates, and verifies online without human intervention. To this end, REGEN uses a streaming VLM that unifies planning and verification under a single token policy $p_\theta(Y|\tau, X)$ with prompting type $\tau \in \{\text{PRED}, \text{JUDGE}\}$:

$$\begin{aligned} \text{PRED}: X &= (V_{t-1},\ \mathcal{H}_{t-1}),\ Y = (s_t, a_t) = (c_t^{\text{delta}},\ e_t,\ T_t,\ a_t), \\ \text{JUDGE}: X &= (V_t,\ \mathcal{H}_{t-1},\ s_t,\ a_t),\ Y = (m_t^a,\ m_t^c,\ m_t^e,\ d_t), \end{aligned} \tag{3}$$

where $m_t^c, m_t^e, m_t^a \in \{0, 1\}$ are per-field agreement flags, and d_t is the verification decision, including CONTINUE, END, and RE-GENERATE. When $d_t =$ RE-GENERATE, the per-field flags indicate which specification fields failed, and the planner revises only those fields via a zero-shot REFINE prompt before the generator re-generates with the modified condition. We also provide a compact textual summary of the history in $\mathcal{H}$ to enhance VLM memory, as detailed in Appendix B.3. This formulation casts the planner and verifier as two prompted views of a shared distribution: $\pi_\theta \equiv p_\theta(\cdot|\text{PRED}, \cdot)$ and $\phi_\theta \equiv p_\theta(\cdot|\text{JUDGE}, \cdot)$, sharing a common schema and representation space. A frozen copy $p_{\bar{\theta}}(\cdot|\text{JUDGE}, \cdot)$ can further serve as a reward model to train both π_θ and p_ψ.

Format Prior SFT and Cross-Task GRPO We first distill the structured output format from the action-grounded annotations in Section 4 via a small mixed-task supervised fine-tuning (SFT) over (PRED, JUDGE). As cross-entropy quickly saturates, this stage serves as a *format prior*. Semantic quality is then optimized with GRPO [85] under a three-term reward:

$$R(Y, X) = R_{\text{fmt}}(Y) + R_{\text{match}}(Y, Y^\star) + R_{\text{judge-as-reward}}(Y, V_t^\star;\ p_{\bar{\theta}}(\cdot|\text{JUDGE}, \cdot)), \tag{4}$$

comprising a format term (JSON schema validity) and a field-match term (BLEU-1 for text fields, MAE for duration prediction, and verdict correctness in JUDGE). For the PRED task, we add the third term $R_{\text{judge-as-reward}}$, where a frozen copy $p_{\bar{\theta}}$ operating in JUDGE mode receives the predicted specification $(c_t^{\text{delta}}, e_t, a_t)$ together with the *real* end frames $V_t^\star$ and returns an accept/reject verdict as reward. Since the verifier is jointly trained alongside the planner under the same schema, it directly serves as a grounded reward model for planning, providing visual-semantic supervision without any external reward model. Reward details and prompt templates are provided in Appendix B.3.

3.4 Action-grounded Reward Alignment

Forward-Process Policy Optimization on Segment Rollouts The distilled generator p_ψ operates in a few-step regime, making reverse-process likelihood gradients intractable. We therefore optimize the forward process following DiffusionNFT [86, 87, 88], and tailor the rollout to our action-grounded segment structure. At each iteration, we perform a self-forced prefix rollout, freeze the KV cache, and draw G candidate segments $\{V_t^{(i)}\}_{i=1}^G$ that share the prefix and differ only within one action-grounded segment. With group-centered advantage $A_i = R_i - \frac{1}{G}\sum_j R_j$ and normalized weight $\tilde{r}_i = \text{clip}(A_i/A_{\max})/2 + 0.5$, we train a LoRA [89] adapter on top of p_ψ via

$$\mathcal{L}_{\text{NFT}}(\psi) = \mathbb{E}_{t, \mathcal{H}_{t-1}, s_t, a_t}\left[\tilde{r}_i \left\|p_\psi^+ - \mathbf{u}_{\text{target}}\right\|_2^2 + (1 - \tilde{r}_i)\left\|p_\psi^- - \mathbf{u}_{\text{target}}\right\|_2^2\right], \tag{5}$$

where $p_\psi^\pm = (1 \mp \beta)\, p_{\psi_{\text{old}}} \pm \beta\, g_\psi$ are DiffusionNFT's implicit positive and negative policies [86], built from the old policy $p_{\psi_{\text{old}}}$ and the current LoRA-adapted prediction g_ψ. Rewards are computed on segment-local samples conditioned on $(s_t,\ a_t)$, which localizes credit assignment to the segment-t trajectory and mitigates cross-segment leakage.

VLM-driven Composite Reward We define a composite segment-level reward for reinforcement post-training:

$$R(V_t, (s_t, a_t)) = R_{\text{verdict}}(V_t, (s_t, a_t);\ p_{\bar{\theta}}) + R_{\text{VideoAlign}}(V_t, (s_t,\ a_t)) + R_{\text{HPS}}(V_t, (s_t,\ a_t)), \tag{6}$$

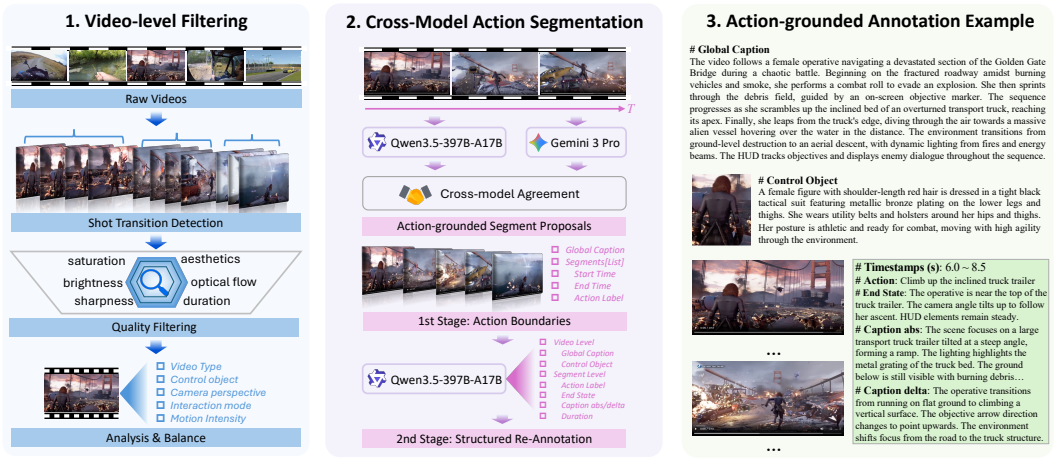

Figure 3: **Action-grounded Video Dataset Curation**. Raw videos pass through multi-stage quality and balance filtering, then a two-stage cross-model pipeline produces action-grounded segments and structured captions. The right panel shows a real annotation example.

where $R_{\text{VideoAlign}}$ [90] and R_{HPS} [91] are different reward models based on pretrained VLMs for human preference alignment. R_{verdict} provides an *agentic verdict*, in which we query the frozen verifier $p_{\bar{\theta}}(\cdot|\text{JUDGE}, \cdot)$ with $(V_t, (s_t, a_t))$, extract per-field agreement (m_t^c, m_t^e, m_t^a) as Equation 3, and set $R_{\text{verdict}} = \frac{1}{3}(m_t^c + m_t^e + m_t^a)$. Since $p_{\bar{\theta}}$ shares the same schema as the inference verifier ϕ_θ, Equation 6 directly optimizes acceptance under the contract that will later gate the rollout, turning the action specification from a soft prompt into a trainable target. To guard against reward hacking, we (i) keep $p_{\bar{\theta}}$ disjoint from the held-out evaluation judges, (ii) use $R_{\text{VideoAlign}}$ and R_{HPS} to penalize trivial spec-matching, and (iii) report held-out evaluation metrics in Section 5.

4 Action-Grounded Video Dataset

Most video generation datasets [38, 39, 40, 41, 42] focus on dense captioning and lack action-grounded control annotations, making them less suitable for world modeling. To support the agentic formulation in Section 3, we carefully construct a large-scale action-grounded video dataset with over **400K** videos and **2M** segments, and detailed segment-level control and state transition annotations.

Video-level Filtering We collect raw videos from license-screened public and internal sources, and apply cascaded filters before action-grounded captioning. We remove low-quality videos using visual quality and motion cues, discard videos with excessive overlays or shot transitions, and retain clips with a clear control object, minimal competing signals, and suitable agent-centric viewpoints. This reduces the corpus from 11.9M to 6.2M, then 3.7M, and finally to 700K videos for annotation.

Action-grounded Annotation As illustrated in Figure 3, the filtered 700K videos are then annotated by a two-stage cross-model pipeline that jointly produces action boundaries and structured descriptions. In the first stage, two strong VLMs, Qwen3.5-397B-A17B [92] and Gemini 3 [93], independently predict per-segment tuples of (`action_label`, `start_time`, `end_time`). Segments with large timestamp disagreements are dropped, segments shorter than 0.5 seconds or longer than 10 seconds are also discarded, and the surviving segments are temporally adjusted to tile each video without gaps. In the second stage, the agreed timestamps are used as priors in Qwen3.5 to re-identify actions and produce final annotations, including video-level fields (`global_caption`, `control_object`) and segment-level tuples (`action_label`, `caption_abs`, `caption_delta`, `end_state`, `start_time`, `end_time`). Here, `caption_abs` describes the segment in isolation, `caption_delta` captures visual and behavioral changes relative to the previous segment, and `end_state` records the resulting environmental state after the action completes. This process yields **400K videos** and **2M action-grounded segments**. These annotations instantiate the shared protocol consumed by the VLM planner and the causal DiT generator in Section 3, providing a common interface for reasoning and generation rather than a learned alignment. The final videos average **20 seconds** and **five segments** each, with annotation details and dataset statistics provided in Appendix A.

Table 1: **Quantitative Comparison** on test prompts under two horizons of 241 frames (**15s**) and 961 frames (**60s**). The **best**, second-best, and third-best results are highlighted for clarity.

	↑ Caption Align.		↑ Imaging Qual.		↓ Semantic Drift		↓ Visual Drift	
	15s	60s	15s	60s	15s	60s	15s	60s
Rolling Sink [51]	0.1450	0.1205	0.6720	0.5520	0.0295	0.0795	0.0428	0.0585
Infinity-RoPE [17]	0.1485	0.1235	0.6850	0.5650	0.0268	0.0728	0.0395	0.0532
Helios [50]	0.1508	0.1263	0.7048	0.6369	0.0212	0.0863	0.0331	0.0447
LongCat-Video [21]	0.1608	0.1529	**0.7604**	0.5395	**0.0196**	0.0652	0.0386	0.0831
Krea [99]	0.1667	0.1274	0.7358	0.7125	0.0302	0.0450	0.0288	0.0410
REGEN	**0.1848**	**0.1676**	0.7418	**0.7413**	0.0201	**0.0372**	**0.0161**	**0.0382**

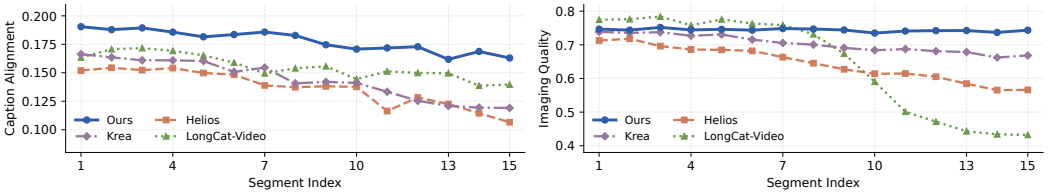

Figure 4: **Drift over Time**. Unlike Table 1, which evaluates each full rollout, caption alignment (left) and imaging quality (right) are computed per segment over 15 segments, against each segment's structured caption and its frame-level perceptual quality, respectively.

5 Experiments

5.1 Experimental Setup

Implementation Details The causal DiT generator p_ψ is initialized from Wan2.1-T2V-14B [19] and fully fine-tuned at 480P on the proposed dataset in Section 4, using two-stage teacher forcing and self forcing in Section 3.2. The VLM planner and verifier π_θ/ϕ_θ are initialized from Qwen3-VL-2B-Instruct [94] and fine-tuned with rank 16 LoRA adapters under the two-stage SFT and GRPO protocol in Section 3.3. In inference, the distilled causal DiT performs block-wise generation with 3 latents per block and 4 denoising steps per block. We set the attention sink size to 3 and the sliding window size to 9, so the KV cache stores tokens for 12 latent frames plus one end-state latent frame per completed segment. The rollout allows up to $K{=}2$ re-generation attempts before accepting the segment with a failure flag. To improve memory efficiency, we apply gradient checkpointing [95], BF16 mixed precision [96], FSDP [97], and Ulysses-style sequence parallelism (SP) [98]. See Appendix B for more implementation and training details.

Baselines and Evaluation Metrics We conduct quantitative comparisons with several recent strong long-video generation baselines, including Infinity-RoPE [17], Rolling Sink [51], Krea [99], LongCat-Video [21], and Helios [50]. All models have been distilled to support a few-step denoising. Following previous evaluation protocols [100, 50], we report four metrics, including (1) **caption alignment** via ViCLIP similarity [101] between each segment and its caption, (2) **imaging quality** via the VBench [100] predictor, (3) **semantic drift** as the change in caption alignment between the first and last 15% of the rollout, and (4) **visual drift** as the corresponding change in visual quality. All methods are evaluated on the same held-out inputs, consisting of a global caption, per-segment action labels, and durations. Since baselines do not expose a planner, we provide them with the ground-truth segment-level structured captions with $(c_t^{\text{delta}}, e_t)$ at each boundary, giving them the strongest dynamic text conditioning available under our annotation schema. In contrast, REGEN receives only the shared global caption, action labels, and segment durations for fair comparison, and its VLM planner predicts $(c_t^{\text{delta}}, e_t)$ online from rollout history, while the verifier checks each realized segment during generation.

5.2 Comparison Results

Quantitative Comparison By disentangling reasoning and generation, REGEN demonstrates stronger robustness as the rollout horizon increases. As shown in Table 1, it achieves the best results

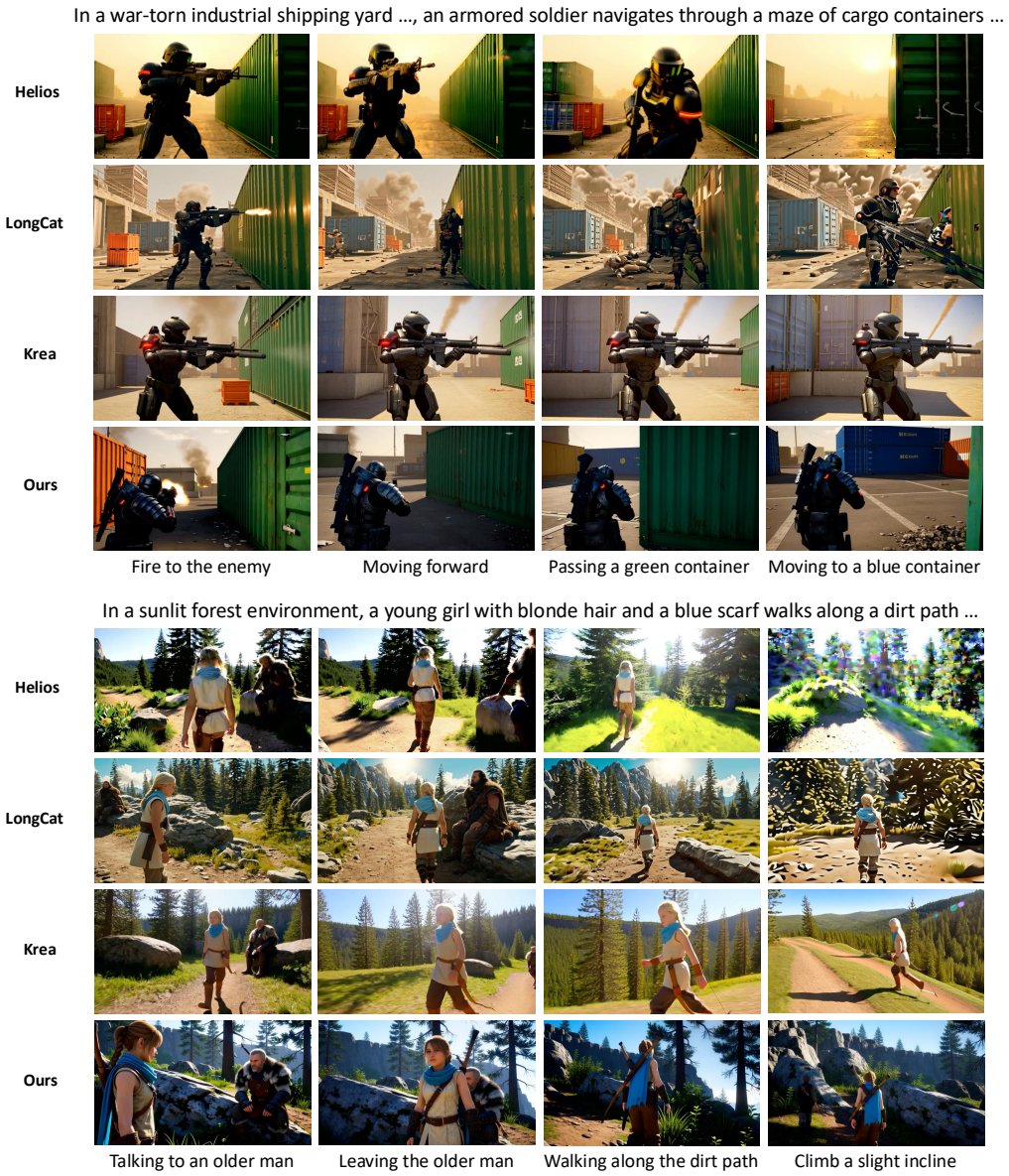

Figure 5: **Qualitative Comparison** under the same global caption (top) and per-segment action labels (bottom). More visualization results are provided in Appendix C.

on all four metrics at 961 frames (60s) and remains among the top two at 241 frames (15s). More importantly, REGEN trails the strongest baseline on imaging quality and semantic drift at 15s but leads on all four metrics at 60s, suggesting that the benefit of REGEN is not a short-term quality gain but a reduction of accumulated error. The improvement in visual drift further supports that structured action specifications and closed-loop verification help prevent perceptual collapse when generation is repeatedly extended across many segments.

Drift over Time We further analyze per-segment metrics in Figure 4, with segments aligned to the action specifications. Baselines degrade at different rates: Krea preserves early visual quality but gradually loses alignment, Helios deteriorates in the second half, and LongCat-Video suffers a sharp imaging-quality drop in later segments. LongCat-Video extends videos by bidirectional video-to-video continuation and is not trained on its own long-horizon rollouts as in self forcing, so its errors accumulate faster as the horizon grows. In contrast, REGEN degrades the least across the full trajectory. Together with the 15s-to-60s gap in Table 1, this indicates that our gains come from suppressing per-step drift, rather than from improving only the first few generated segments.

Table 2: **Ablation Study** of REGEN. Each row replaces one design choice, and (c2) and (d2) are 60s only. The bottom row is the full REGEN from Table 1. Cells are shaded by relative degradation against the full REGEN: mild (2−15%), moderate (15−50%), severe (> 50%).

	↑ Caption Align.		↑ Imaging Qual.		↓ Semantic Drift		↓ Visual Drift	
	15s	60s	15s	60s	15s	60s	15s	60s
(a) Plain text conditioning (`caption_abs` only)	0.1350	0.1080	0.6800	0.5950	0.0250	0.0950	0.0380	0.0560
(b) w/o action-grounded RL	0.1545	0.1215	0.7330	0.7265	0.0265	0.0540	0.0180	0.0420
(c1) Offline planning (no planner & verifier)	0.1700	0.1510	0.7400	0.7180	0.0207	0.0475	0.0240	0.0405
(c2) Offline planning + verifier (no planner)	–	0.1573	–	0.7291	–	0.0441	–	0.0398
(d1) Same-spec. noise resampling	0.1795	0.1618	0.7425	0.7395	0.0210	0.0388	0.0158	0.0388
(d2) No re-generation (K=0)	–	0.1602	–	0.7368	–	0.0402	–	0.0393
REGEN (full)	0.1848	0.1676	0.7418	0.7413	0.0201	0.0372	0.0161	0.0382

Qualitative Comparison Figure 5 visualizes multi-segment rollouts from the same prompt across baselines and REGEN. Krea maintains strong early visual quality but often falls into repetitive motion or inconsistent state transitions, while Helios and LongCat-Video drift as the horizon grows. REGEN remains aligned with the structured action specifications across segments and can recover from mid-rollout alignment errors. Additional visualization results are provided in Appendix C.

5.3 Ablation Study and Analysis

Action Specification Protocol Table 2(a) replaces structured action specifications with plain text conditioning, where the generator receives only the per-segment caption c_t^{abs}. Without the structured contract, all four metrics degrade at both horizons, most severely on the drift metrics, and 60s imaging quality drops far more than in any other ablation. Long-horizon visual quality thus stems mainly from the structured fields themselves: the relative change c_t^{delta} and the explicit end state e_t tell the generator what to change and where each segment should stop, instead of re-describing the full scene in every segment, so the generator only needs to render what the specification states. Structured fields are hence not auxiliary metadata for generation, but the core interface for planning visible state transitions and for providing the verifier with a checkable contract.

Reward Alignment Table 2(b) removes the action-grounded RL stage and trains the generator only with teacher forcing and self forcing. The main degradation appears in semantic alignment: caption alignment drops and semantic drift worsens at both horizons, while visual quality changes only mildly. Verifier-derived rewards thus chiefly improve specification following and suppress semantic drift, complementing sparse inference-time interventions, while long-horizon visual quality is largely carried by the specification protocol in (a).

Offline vs. Online Table 2(c1) replaces the closed sense-plan-act-verify loop with offline planning, where ground-truth action specifications are provided for all segments, and no verification or refinement happens during rollout. The open-loop variant stays close to the full REGEN on caption alignment and imaging quality, but shows moderate gaps on visual drift at 15s and on semantic drift at 60s. Table 2(c2) adds the verifier back on top of a full plan frozen before rollout, so a rejected segment can only be resampled under the same specification. The verifier alone closes only part of the gap in caption alignment and semantic drift. A frozen plan can reject segments that violate it, but only online re-planning can revise future transitions once the realized state deviates, preventing a single deviation from propagating to all later segments.

Re-generation Strategy Table 2(d1) replaces planner-side specification refinement with noise resampling under the same conditions, keeping the verifier and budget K unchanged. While resampling keeps visual quality on par by exploiting the residual stochasticity of the few-step generator, caption alignment still drops consistently. This shows that failures caused by incorrect or incomplete specifications require targeted planner edits, not repeated sampling under unchanged conditioning. Table 2(d2) disables re-generation entirely, so every segment is generated exactly once, matching the sampling budget of the baselines. This variant still outperforms every baseline in Table 1 on all four 60s metrics, indicating that the structured specification and online planning account for most of the gain, while re-generation further refines it.

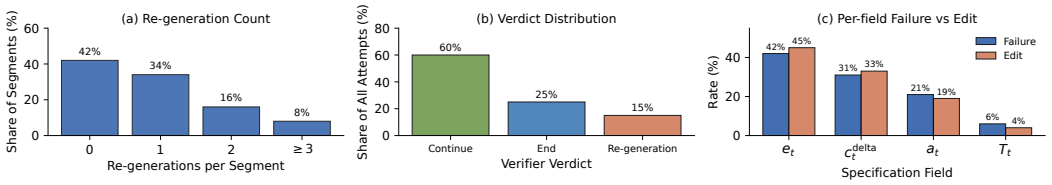

Figure 6: **Closed-Loop Diagnostics**. (a) Per-segment re-generation count under an enlarged retry budget, used to select K. (b) Verifier verdict distribution across all attempts at $K{=}2$. (c) Per-field failure rate (verifier flag) and edit rate (planner refinement) over the specification fields $\{e_t, c_t^{\text{delta}}, a_t, T_t\}$.

Closed-Loop Diagnostics Figure 6 analyzes the runtime behavior of the closed loop. Under an enlarged retry budget, most segments require zero or one re-generation and only a small fraction need more than two, so we set $K{=}2$ in all other experiments. Verdicts are dominated by CONTINUE, with fewer RE-GENERATE decisions, showing that the loop intervenes selectively. Per-field diagnostics identify end state e_t as the main failure source, followed by caption delta c_t^{delta} and action label a_t, while duration T_t rarely fails. The planner's edit rate closely tracks the failure rate, suggesting targeted field updates rather than wholesale rewrites. A module-level attribution of failures to the planner, generator, and verifier is provided in Appendix D. We also test an off-the-shelf Qwen3-VL planner-verifier without our SFT-plus-GRPO training, and it often produces schema-invalid outputs and performs worse than the offline variant in Table 2(c1).

6 Conclusion

By externalizing high-level transition reasoning from pixel generation in video world modeling, we reformulate the problem as an **agentic process**. We present **REGEN**, a closed-loop framework that couples a streaming VLM planner with a causal DiT generator through structured action specifications. Both modules are trained on a carefully curated **action-grounded video dataset** and further coupled via **action-grounded reward alignment**. Experiments demonstrate that REGEN enhances long-horizon fidelity and narrative continuity over prior methods. We hope this agentic reasoning-generation paradigm can serve as a promising starting point toward general, controllable, and verifiable video world models. Limitations and future work are discussed in Appendix D.

Acknowledgment

This work was supported by the Peking University Education Foundation (No. 050621_001).

References

[1] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[2] Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model, 2024.

[3] Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe

Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models, 2025.

[4] Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

[5] Jiannan Xiang, Yi Gu, Zihan Liu, Zeyu Feng, Qiyue Gao, Yiyan Hu, Benhao Huang, Guangyi Liu, Yichi Yang, Kun Zhou, et al. Pan: A world model for general, interactable, and long-horizon world simulation. *arXiv preprint arXiv:2511.09057*, 2025.

[6] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

[7] Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

[8] Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

[9] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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

[11] Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, et al. Relic: Interactive video world model with long-horizon memory. *arXiv preprint arXiv:2512.04040*, 2025.

[12] Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025.

[13] Zile Wang, Zexiang Liu, Jaixing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, et al. Matrix-game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026.

[14] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

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

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

[17] 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. *arXiv preprint arXiv:2511.20649*, 2025.

[18] 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.

[19] 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.

[20] Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. Hunyuanvideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025.

[21] Meituan LongCat Team, Xunliang Cai, Qilong Huang, Zhuoliang Kang, Hongyu Li, Shijun Liang, Liya Ma, Siyu Ren, Xiaoming Wei, Rixu Xie, et al. Longcat-video technical report. *arXiv preprint arXiv:2510.22200*, 2025.

[22] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[23] Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

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

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

[26] 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*, 2026.

[27] 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*, 2026.

[28] Xunzhi Xiang, Zixuan Duan, Guiyu Zhang, Haiyu Zhang, Zhe Gao, Junta Wu, Shaofeng Zhang, Tengfei Wang, Qi Fan, and Chunchao Guo. Pathwise test-time correction for autoregressive long video generation. *arXiv preprint arXiv:2602.05871*, 2026.

[29] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[30] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[31] ByteDance Seed Team. Seed1.5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.

[32] Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025.

[33] Jiaze Li, Jingyang Chen, Yuxun Qu, Shijie Xu, Zhenru Lin, Junyou Zhu, Boshen Xu, Wenhui Tan, Pei Fu, Jianzhong Ju, et al. Xiaomi mimo-vl-miloco technical report. *arXiv preprint arXiv:2512.17436*, 2025.

[34] Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyan Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, et al. Mixture of contexts for long video generation. *arXiv preprint arXiv:2508.21058*, 2025.

[35] Yawen Luo, Xiaoyu Shi, Junhao Zhuang, Yutian Chen, Quande Liu, Xintao Wang, Pengfei Wan, and Tianfan Xue. Shotstream: Streaming multi-shot video generation for interactive storytelling. *arXiv preprint arXiv:2603.25746*, 2026.

[36] Qinglin Zeng, Kaitong Cai, Ruiqi Chen, Qinhan Lv, and Keze Wang. Coagent: Collaborative planning and consistency agent for coherent video generation. *arXiv preprint arXiv:2512.22536*, 2025.

[37] Yu Yang, Yue Liao, Jianbiao Mei, Baisen Wang, Xuemeng Yang, Licheng Wen, Jiangning Zhang, Xiangtai Li, Hanlin Chen, Botian Shi, et al. Spiral: A closed-loop framework for self-improving action world models via reflective planning agents. *arXiv preprint arXiv:2603.08403*, 2026.

[38] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.

[39] Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. Miradata: A large-scale video dataset with long durations and structured captions. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[40] Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, et al. Sekai: A video dataset towards world exploration. *arXiv preprint arXiv:2506.15675*, 2025.

[41] Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. *arXiv preprint arXiv:2509.09676*, 2025.

[42] Kairun Wen, Yuzhi Huang, Runyu Chen, Hui Zheng, Yunlong Lin, Panwang Pan, Chenxin Li, Wenyan Cong, Jian Zhang, Junbin Lu, et al. Dynamicverse: A physically-aware multimodal framework for 4d world modeling. *arXiv preprint arXiv:2512.03000*, 2025.

[43] 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 (NeurIPS)*, 2024.

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

[45] 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.

[46] Yunhong Lu, Yanhong Zeng, Haobo Li, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jiapeng Zhu, Hengyuan Cao, Zhipeng Zhang, Xing Zhu, et al. Reward forcing: Efficient streaming video generation with rewarded distribution matching distillation. *arXiv preprint arXiv:2512.04678*, 2025.

[47] Ryan Po, Yotam Nitzan, Richard Zhang, Berlin Chen, Tri Dao, Eli Shechtman, Gordon Wetzstein, and Xun Huang. Long-context state-space video world models. In *International Conference on Computer Vision (ICCV)*, 2025.

[48] 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.

[49] SII-GAIR, Sand. ai, Ethan Chern, Hansi Teng, Hanwen Sun, Hao Wang, Hong Pan, Hongyu Jia, Jiadi Su, Jin Li, Junjie Yu, Lijie Liu, Lingzhi Li, Lyumanshan Ye, Min Hu, Qiangang Wang, Quanwei Qi, Steffi Chern, Tao Bu, Taoran Wang, Teren Xu, Tianning Zhang, Tiantian Mi, Weixian Xu, Wenqiang Zhang, Wentai Zhang, Xianping Yi, Xiaojie Cai, Xiaoyang Kang, Yan Ma, Yixiu Liu, Yunbo Zhang, Yunpeng Huang, Yutong Lin, Zewei Tao, Zhaoliang Liu, Zheng Zhang, Zhiyao Cen, Zhixuan Yu, Zhongshu Wang, Zhulin Hu, Zijin Zhou, Zinan Guo, Yue Cao, and Pengfei Liu. Speed by simplicity: A single-stream architecture for fast audio-video generative foundation model. *arXiv preprint arXiv:2603.21986*, 2026.

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

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

[52] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

[53] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[54] Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. Mineworld: a real-time and open-source interactive world model on minecraft. *arXiv preprint arXiv:2504.08388*, 2025.

[55] Xiaofeng Mao, Zhen Li, Chuanhao Li, Xiaojie Xu, Kaining Ying, Tong He, Jiangmiao Pang, Yu Qiao, and Kaipeng Zhang. Yume-1.5: A text-controlled interactive world generation model. *arXiv preprint arXiv:2512.22096*, 2025.

[56] Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Cyrus Wu, Wei Li, Xuchen Song, Yang Liu, Eric Li, and Yahui Zhou. Matrix-game 2.0: An open-source, real-time, and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

[57] InSpatio Team, Donghui Shen, Guofeng Zhang, Haomin Liu, Haoyu Ji, Hujun Bao, Hongjia Zhai, Jialin Liu, Jing Guo, Nan Wang, et al. Inspatio-world: A real-time 4d world simulator via spatiotemporal autoregressive modeling. *arXiv preprint arXiv:2604.07209*, 2026.

[58] Boyuan Chen, Tianyuan Zhang, Haoran Geng, Kiwhan Song, Caiyi Zhang, Peihao Li, William T Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, et al. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025.

[59] Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

[60] Team HunyuanWorld. Hy-world 1.5: A systematic framework for interactive world modeling with real-time latency and geometric consistency. *arXiv preprint*, 2025.

[61] Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning. *arXiv preprint arXiv:2309.15091*, 2023.

[62] Kunpeng Song, Tingbo Hou, Zecheng He, Haoyu Ma, Jialiang Wang, Animesh Sinha, Sam Tsai, Yaqiao Luo, Xiaoliang Dai, Li Chen, et al. Directorllm for human-centric video generation. *arXiv preprint arXiv:2412.14484*, 2024.

[63] Zun Wang, Jialu Li, Han Lin, Jaehong Yoon, and Mohit Bansal. Dreamrunner: Fine-grained compositional story-to-video generation with retrieval-augmented motion adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

[64] Shaobin Zhuang, Kunchang Li, Xinyuan Chen, Yaohui Wang, Ziwei Liu, Yu Qiao, and Yali Wang. Vlogger: Make your dream a vlog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[65] Wenzhang Sun, Zhenyu Wang, Zhangchi Hu, Chunfeng Wang, Hao Li, and Wei Chen. Muse: A multi-agent framework for unconstrained story envisioning via closed-loop cognitive orchestration. *arXiv preprint arXiv:2602.03028*, 2026.

[66] Qian Wang, Ziqi Huang, Ruoxi Jia, Paul Debevec, and Ning Yu. Mavis: A multi-agent framework for long-sequence video storytelling. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (ACL)*, 2026.

[67] Weijia Wu, Zeyu Zhu, and Mike Zheng Shou. Automated movie generation via multi-agent cot planning. *arXiv preprint arXiv:2503.07314*, 2025.

[68] Kaituo Feng, Manyuan Zhang, Shuang Chen, Yunlong Lin, Kaixuan Fan, Yilei Jiang, Hongyu Li, Dian Zheng, Chenyang Wang, and Xiangyu Yue. Gen-searcher: Reinforcing agentic search for image generation. *arXiv preprint arXiv:2603.28767*, 2026.

[69] Zefeng He, Siyuan Huang, Xiaoye Qu, Yafu Li, Tong Zhu, Yu Cheng, and Yang Yang. Gems: Agent-native multimodal generation with memory and skills. *arXiv preprint arXiv:2603.28088*, 2026.

[70] Achint Soni, Sreyas Venkataraman, Abhranil Chandra, Sebastian Fischmeister, Percy Liang, Bo Dai, and Sherry Yang. Videoagent: Self-improving video generation. *arXiv preprint arXiv:2410.10076*, 2024.

[71] Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and Lu Jiang. Long context tuning for video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[72] Zhaochong An, Menglin Jia, Haonan Qiu, Zijian Zhou, Xiaoke Huang, Zhiheng Liu, Weiming Ren, Kumara Kahatapitiya, Ding Liu, Sen He, et al. Onestory: Coherent multi-shot video generation with adaptive memory. *arXiv preprint arXiv:2512.07802*, 2025.

[73] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[74] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International conference on robotics and automation (ICRA)*, 2023.

[75] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[76] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

[77] Marta Skreta, Zihan Zhou, Jia Lin Yuan, Kourosh Darvish, Alán Aspuru-Guzik, and Animesh Garg. Replan: Robotic replanning with perception and language models. *arXiv preprint arXiv:2401.04157*, 2024.

[78] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[79] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[80] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

[81] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.

[82] Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. *arXiv preprint arXiv:2506.03141*, 2025.

[83] Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. Vmem: Consistent interactive video scene generation with surfel-indexed view memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[84] Junyi Chen, Haoyi Zhu, Xianglong He, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Zhoujie Fu, Jiangmiao Pang, et al. Deepverse: 4d autoregressive video generation as a world model. *arXiv preprint arXiv:2506.01103*, 2025.

[85] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[86] Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. *arXiv preprint arXiv:2509.16117*, 2025.

[87] Zehan Wang, Tengfei Wang, Haiyu Zhang, Xuhui Zuo, Junta Wu, Haoyuan Wang, Wenqiang Sun, Zhenwei Wang, Chenjie Cao, Hengshuang Zhao, et al. Worldcompass: Reinforcement learning for long-horizon world models. *arXiv preprint arXiv:2602.09022*, 2026.

[88] Songchun Zhang, Zeyue Xue, Siming Fu, Jie Huang, Xianghao Kong, Y Ma, Haoyang Huang, Nan Duan, and Anyi Rao. Astrolabe: Steering forward-process reinforcement learning for distilled autoregressive video models. *arXiv preprint arXiv:2603.17051*, 2026.

[89] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[90] Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, et al. Improving video generation with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[91] Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, 2025.

[92] Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026.

[93] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[94] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[95] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[96] Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, et al. A study of bfloat16 for deep learning training. *arXiv preprint arXiv:1905.12322*, 2019.

[97] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

[98] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020.

[99] Erwann Millon. Krea realtime 14b: Real-time video generation, 2025.

[100] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[101] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023.

[102] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

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

[104] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 2020.

[105] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

[106] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[107] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

A Dataset Details

A.1 Dataset Statistics

Figure 7 shows the distribution of video durations and segment counts across the 400K-video corpus. The majority of videos fall in the 10 to 25 second range, with a long tail extending beyond one minute, reflecting the diversity of activity lengths in the wild. Segment counts per video are concentrated between 3 and 6, consistent with the average clip containing about five action-grounded transitions.

Figure 8 visualizes the vocabulary of four annotation fields. The `action_label` cloud is dominated by locomotion and interaction verbs (`Walk`, `Run`, `Navigate`, `Approach`, `Fire`), indicating broad coverage of physically grounded behaviors. The `control_object` cloud centers on body-part and appearance descriptors (`hand`, `dark`, `wear`, `person`), reflecting that the dataset captures both human-centric and agent-centric perspectives. The `end_state` cloud features spatial and postural terms (`stand`, `facing`, `ready`, `room`, `path`), confirming that end states describe concrete, verifiable world configurations rather than abstract goals. The `caption_delta` cloud is dominated by change-of-state language (`shift`, `moved`, `transitions`, `change`, `camera`), showing that the delta field effectively captures inter-segment dynamics distinct from absolute descriptions.

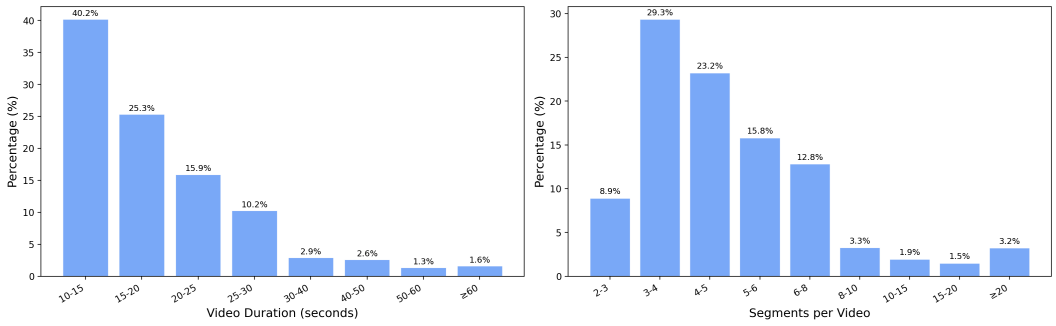

Figure 7: **Video Duration and Segment Count Distributions.** Left: most videos are 10 to 25 seconds long, with a tail of longer clips. Right: most videos contain 3 to 6 segments, consistent with the average of about five action-grounded transitions per clip.

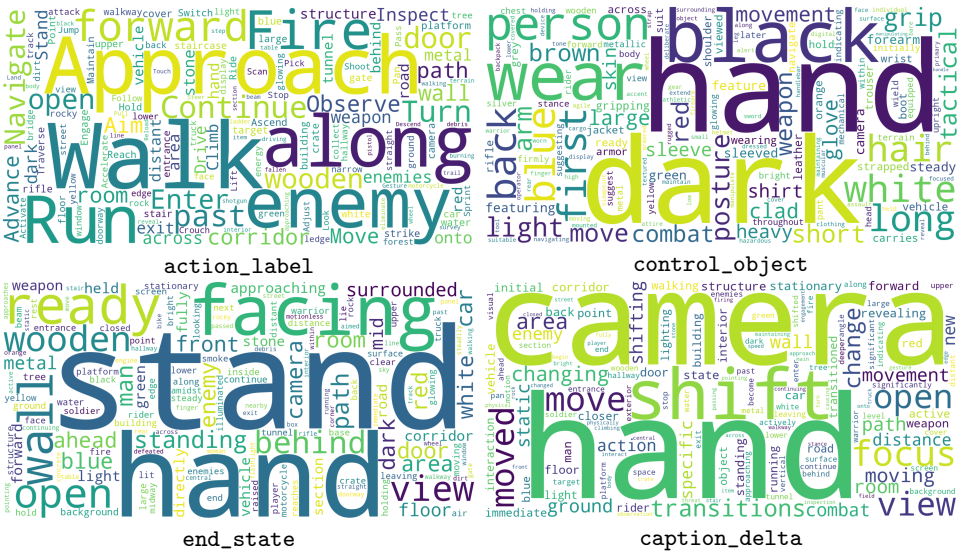

Figure 8: **Annotation Field Word Clouds.** Each cloud visualizes the vocabulary of one contract field. `action_label` spans diverse locomotion and interaction verbs. `control_object` reflects the physical appearance of objects. `end_state` describes verifiable post-action world configurations. `caption_delta` captures inter-segment transitions through change-of-state language.

A.2 Sample Examples from the Dataset

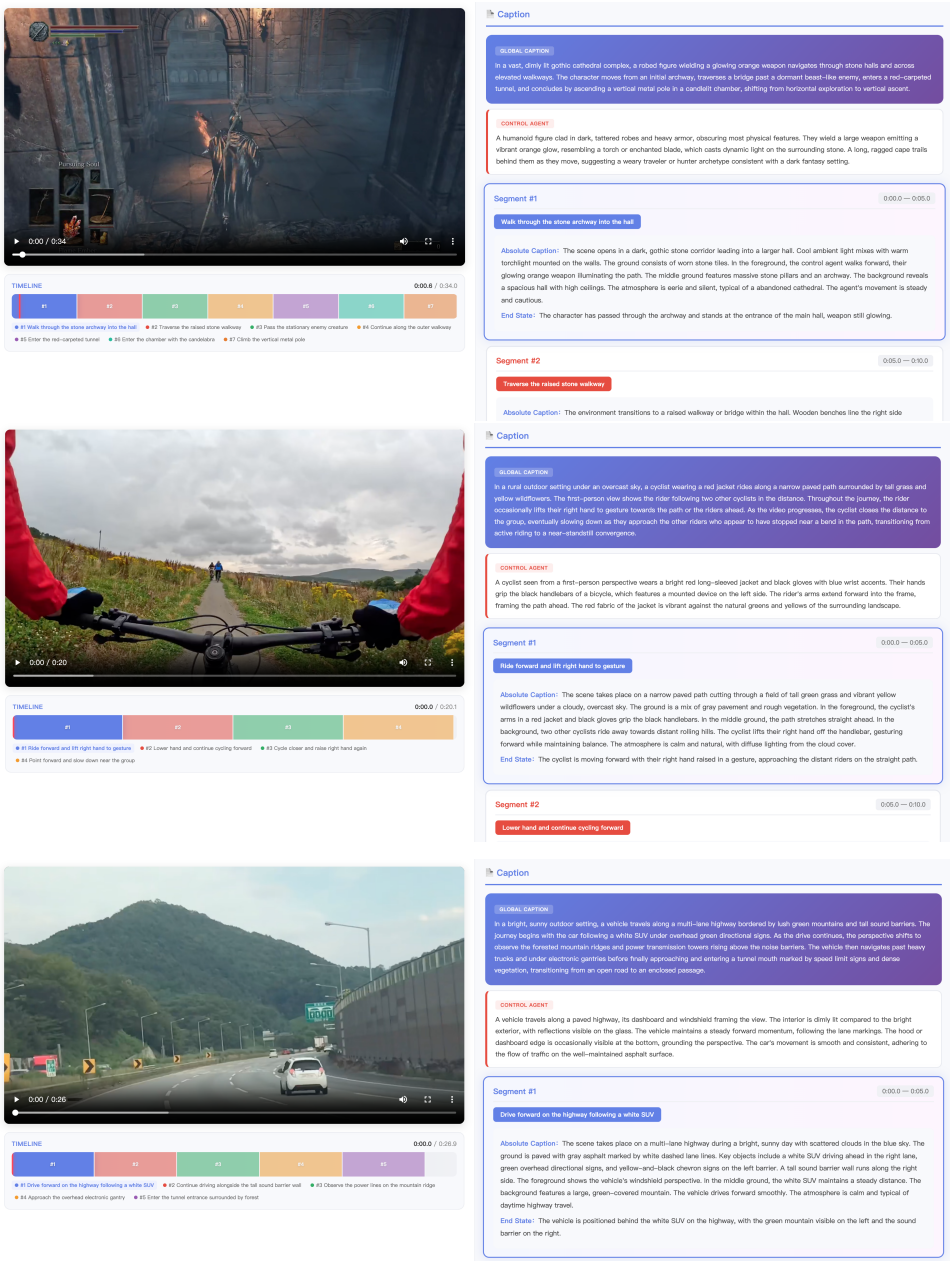

Figure 9: **Dataset Annotation Examples.** Each example shows the segmented timeline, global caption, control object description, and per-segment action label, absolute caption, and end state.

Figure 9 shows some sample annotation examples spanning different domains in the dataset. Each example displays the video timeline with color-coded action segments alongside the full annotation: a **global caption** describing the overall scenario, a **control object** description capturing the physical appearance of the primary actor, and per-segment fields including the **action label**, **absolute caption**, and **end state**. The examples illustrate the diversity of the corpus: a third-person dark-fantasy game sequence with seven semantically distinct segments, a first-person cycling clip with four transitions capturing subtle gesture and motion changes, and a dashcam driving video with five segments following a highway route into a tunnel. Across all three, the **end state** field provides a concrete, verifiable description of the world after each action completes, directly instantiating the contract interface consumed by the planner and verifier in REGEN.

B Implementation Details

B.1 Closed-Loop Rollout Algorithm

Algorithm 1 gives the complete pseudocode for the REGEN inference loop described in Section 3.1. At each segment step, the VLM planner π_θ predicts the next action specification, the causal DiT p_ψ generates the corresponding video chunks autoregressively, and the verifier ϕ_θ assesses the outcome once the predicted duration T_t is reached. A CONTINUE verdict means the action is still in progress, so the segment is extended by one more chunk and verified again. An END verdict accepts the segment, and a RE-GENERATE verdict triggers specification refinement and a retry from the last accepted history. Segments that still fail after K retries are absorbed into the history with a failure flag so that the planner can adapt future actions accordingly.

Algorithm 1 REGEN Closed-Loop Rollout

Require: Initial frame I_0, global instruction a_0, max segments $T_{\max}$, retry budget K
Ensure: Generated video V

$V \leftarrow []$; $t \leftarrow 1$
Init: $(c_0^{\text{abs}}, e_0, T_0) \leftarrow \pi_\theta(I_0, a_0)$ ▷ absolute caption for first segment
$\mathcal{H}_0 \leftarrow \text{Summary}(a_0, c_0^{\text{abs}}, e_0)$
while $t \leq T_{\max}$ **do**
  Plan: $(c_t^{\text{delta}}, e_t, T_t, a_t) \leftarrow \pi_\theta(\mathcal{H}_{t-1})$ ▷ PRED prompt
  $r \leftarrow 0$
  repeat
    $i \leftarrow 1$; $\tau \leftarrow 0$; $S_t \leftarrow []$; $d_t \leftarrow$ CONTINUE
    while $d_t =$ CONTINUE **do**
      Generate: $V_t^{(i)} \leftarrow p_\psi\big(c_t^{\text{delta}}, e_t, a_t,\ \mathcal{H}_{t-1},\ S_t\big)$
      Append $V_t^{(i)}$ to S_t; $\tau \leftarrow \tau + \Delta\tau_t^{(i)}$; $i \leftarrow i + 1$
      if $\tau \geq T_t$ **then**
        Verify: $(m^a, m^c, m^e, d_t) \leftarrow \phi_\theta(S_t, c_t^{\text{delta}}, e_t, a_t,\ \mathcal{H}_{t-1})$ ▷ JUDGE prompt
      end if
    end while
    if $d_t =$ RE-GENERATE **and** $r < K$ **then**
      Refine: $(c_t^{\text{delta}}, e_t, T_t, a_t) \leftarrow \pi_\theta^{\text{REFINE}}(c_t^{\text{delta}}, e_t, a_t,\ m^a, m^c, m^e,\ S_t)$ ▷ revise failing fields
    end if
    $r \leftarrow r + 1$
  until $d_t =$ END **or** $r > K$
  Append S_t to V
  $\mathcal{H}_t \leftarrow \text{Summary}(\mathcal{H}_{t-1},\ a_t,\ c_t^{\text{delta}},\ e_t,\ S_t[-1])$ ▷ text summary + last-frame anchor
  $t \leftarrow t + 1$
end while
return V

B.2 Details of DiT Generator

DiT Structured Caption Template (per segment)

```
{global_caption}
[Control Object]: {control_object}
[Action]: {action_label}
[Environment]: {caption_abs}                    (first segment only)
[Environment Change]: {caption_delta}           (subsequent segments)
[End State]: {end_state}
```

Figure 10: **Structured caption template for DiT cross-attention.** The template is shared across training and inference. Field values are sourced from action-grounded annotations during training and from the planner during inference. The action label is additionally fed into the action-FiLM modulation path independently of this textual caption.

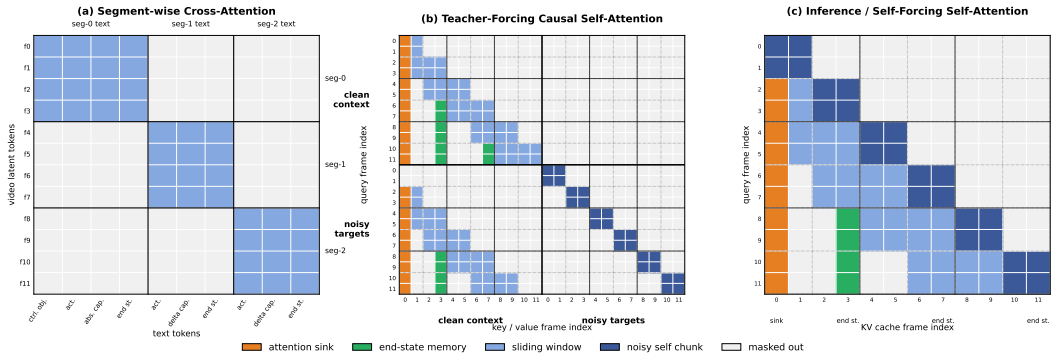

Figure 11: **Attention masks of the DiT generator across training stages.** (a) Stage I: segment-wise cross-attention, (b) Stage II: teacher-forcing causal self-attention on the concatenated [clean | noisy] sequence, and (c) Stage III: inference self-attention, where the clean half collapses into the KV cache with sink and end-state memory pinned.

B.2.1 Structured Caption Template

The per-segment caption fed to the DiT cross-attention is assembled from the action-grounded annotation fields according to the template in Figure 10. The first segment uses an absolute environment description (`caption_abs`) to anchor the scene, while subsequent segments switch to a relative environmental change (`caption_delta`) so that the generator models state transitions rather than re-describing the full scene. The same template is reused at inference for both the offline (scripted) and closed-loop (VLM-driven) variants of REGEN, with each field provided either by the dataset annotation (offline) or the planner output (closed-loop).

B.2.2 Stage I: Bidirectional SFT

Starting from Wan2.1-T2V-14B [19], we incorporate the action-grounded conditioning of Section 3.2 under the original bidirectional self-attention. With the action-FiLM projection zero-initialized, the new conditioning channel begins as identity. Given a clip with J segments, the cross-attention mask is block-diagonal:

$$\mathbf{M}^{\text{xattn}}_{p,q} = \mathbf{1}[\text{seg}(p) = \text{seg}(q)], \tag{7}$$

where p indexes a video latent token, q indexes a text token, and $\text{seg}(\cdot)$ returns the owning segment (Figure 11(a)). For video latents $\mathbf{X}_\sigma := (1-\sigma)\,\mathbf{X}_0 + \sigma\,\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and diffusion time $\sigma \in [0, 1]$, we minimize the flow-matching loss [102, 103]:

$$\mathcal{L}_{\text{SFT}}(\psi) := \omega(\sigma) \cdot \left\| v_\psi(\mathbf{X}_\sigma,\, \sigma,\, \mathbf{T},\, \mathbf{a},\, \mathbf{M}^{\text{xattn}}) - (\boldsymbol{\epsilon} - \mathbf{X}_0) \right\|_2^2, \tag{8}$$

where $\mathbf{T}$ stacks the per-segment T5 embeddings [104] of the structured captions, and $\mathbf{a}$ is the per-segment SigLIP action label embeddings [79].

B.2.3 Stage II: Teacher-Forcing Causal SFT

Stage II converts the bidirectional backbone into a chunk-wise causal generator while inheriting all Stage I conditioning interfaces. Each training clip is duplicated into a clean half $\mathbf{X}_0$ and a noisy half $\mathbf{X}_\sigma$, concatenated into a single sequence of length $2N$, and processed under the mask $\mathbf{M}^{\text{causal}}$ in Figure 11(b). This separates noise from clean context within one forward pass and matches the Stage III KV-cache layout exactly. Let B be the chunk size, R the sink size, W the sliding-window length, and $\mathcal{E} \subset [0, N)$ the latent indices of the end-state frame of every completed segment. With $\bar{x} := x \bmod N$, $c(x) := \lfloor \bar{x}/B \rfloor$, and chunk-end upper bound $u_i := (c(i) + \mathbf{1}[i < N])\, B$,

$$\mathbf{M}^{\text{causal}}_{ij} = \begin{cases} \text{Vis}(i,j), & j \in [0, N), \\ \mathbf{1}[c(i) = c(j)], & i, j \in [N, 2N), \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

$$\text{Vis}(i,j) := \mathbf{1}[\bar{j} < u_i] \cdot \big(\mathbf{1}[\bar{j} \geq u_i - W] \vee \mathbf{1}[\bar{j} < R] \vee \mathbf{1}[\bar{j} \in \mathcal{E}]\big).$$

unifying chunked causality with the union of sliding window, attention sink [81], and end-state memory $\mathcal{E}$. The end-state memory pins the last latent frame of each completed segment, yielding an action-grounded summary of the realized post-action world state without any retrieval module. The flow-matching residual is restricted to the noisy half:

$$\mathcal{L}_{\text{TF}}(\psi) \coloneqq \omega(\sigma) \cdot \left\| v_\psi^{\text{noisy}}([\mathbf{X}_0,\, \mathbf{X}_\sigma],\, \sigma,\, \mathbf{T},\, \mathbf{a},\, \mathbf{M}^{\text{causal}}) - (\boldsymbol{\epsilon} - \mathbf{X}_0) \right\|_2^2. \tag{10}$$

Rotary embeddings [105] are re-assigned dynamically within the active window so that training and inference share a fixed positional range.

B.2.4 Stage III: Self-Forcing Distillation

The few-step student p_ψ is initialized from Stage II, while the Stage I bidirectional checkpoint provides both the frozen teacher S_{real} and the initialization of the online fake-score model S_{fake} [106]. At rollout, the clean half of the Stage II sequence collapses into the KV cache and only the current denoising chunk remains as a noisy query, yielding the inference mask in Figure 11(c). Sink and end-state memory $\mathcal{E}$ are pinned across all sliding-window evictions, so each segment's post-action world state remains accessible for the rest of the rollout. At each iteration, the student draws a multi-segment trajectory $\hat{\mathbf{X}}_0^{1:S}$ in 4 denoising steps and updates via Distribution Matching Distillation (DMD) [107, 44]:

$$\nabla_\psi \mathcal{L}_{\text{DMD}} \coloneqq -\mathbb{E}_{\hat{\mathbf{X}}_0^{1:S} \sim p_\psi,\, \sigma} \left[\left(S_{\text{real}}(\hat{\mathbf{X}}_0^{1:S}, \sigma) - S_{\text{fake}}(\hat{\mathbf{X}}_0^{1:S}, \sigma) \right) \frac{\partial \hat{\mathbf{X}}_0^{1:S}}{\partial \psi} \right]. \tag{11}$$

Both scores condition on the same per-segment captions and action-injected FiLM as the student, so the teacher steers the student towards $p_\sigma^{\text{teacher}}$ conditioned on each action rather than an unconditional video prior. Together with $\mathcal{E}$, this enables prompt-to-prompt switching across segments without KV-cache refreshing or boundary blending [15, 17, 99, 50].

Partial Freezing for 14B DMD DMD at the 14B scale is very expensive, as training requires simultaneously hosting a trainable student, a trainable fake-score model, a frozen teacher score model, and a frozen EMA. Even with FSDP and SP, updating both models end-to-end exceeds the compute budget. We therefore adopt a partial-freezing strategy. For the student, we freeze the first 25% of DiT layers inherited from Stage II to preserve stable low-level spatio-temporal representations while adapting the remaining layers to the few-step regime. For the fake-score model, we freeze the first 75% of DiT layers from Stage I and update only the later layers, which are more directly responsible for estimating the online distributional gap. This reduces optimizer state and activation memory while preserving distillation quality in our internal validation.

B.3 Details of VLM Planner and Verifier

The VLM planner and verifier share a single Qwen3-VL-2B [94] backbone with LoRA fine-tuning, and operate in **four** prompting modes for different tasks: (1) The **INITIAL** mode takes the global instruction and the first frame, and outputs a full seg-0 specification (control object, action label, absolute caption, end state, duration). (2) The **PRED** mode takes the history context and predicts the next segment (action label, caption delta, end state, duration). (3) The **JUDGE** mode takes generated frames plus the intended specification, and returns per-field agreement flags and a verdict $\in \{\text{END}, \text{CONTINUE}, \text{RE-GENERATE}\}$. (4) The **REFINE** mode revises a failed specification based on the verifier feedback. The first three modes are trained via SFT followed by GRPO as described in Section 3.3, while REFINE leverages the schema knowledge already acquired after the training.

B.3.1 History Summary and Specification Refinement

History Summary To keep the VLM context bounded, similar to the memory mechanism in DiT (Section 3.2), the VLM planner organizes its history $\mathcal{H}_{t-1}$ into three zones: (1) a **sink** that always retains seg-0 in full (text + key frames) as a global anchor, (2) a **context summary** that compresses skipped intermediate segments into a deterministic text block (under 100 tokens), and (3) a **window** of the most recent $N{=}2$ segments with full text and key frames. The context summary preserves the action chain (`Story so far`), the last skipped segment's scene description (`Current scene` from `caption_delta`), and its end state (`Last end state`).

System Prompt: Initial Prediction (INITIAL)

```
You are a video understanding assistant. Given the first frame of a video or a global caption of a
video, produce a structured caption in JSON format.

You must output a JSON object with these fields:
- "control_object": A detailed physical description of the main character or agent (appearance,
clothing, distinguishing features). If no clear agent, describe the main subject.
- "action_label": A short imperative phrase (5-10 words) describing the main action in this segment.
- "caption_abs": A 2-4 sentence description of the scene, environment, and what happens in this
segment.
- "end_state": One sentence describing the visual state at the end of this segment.
- "duration": A float estimating the segment duration in seconds.

Output ONLY valid JSON. No markdown fences, no extra text.
```

System Prompt: Next-Segment Prediction (PRED)

```
You are a video prediction assistant. Given what happened in previous segments (text + key frames),
predict what will happen in the NEXT segment.

You must output a JSON object with these fields:
- "action_label": A short imperative phrase (5-10 words) predicting the main action in the next
segment.
- "caption_delta": 1-2 sentences predicting what will CHANGE compared to the previous segment (new
actions, scene changes, camera movement, etc.).
- "end_state": One sentence predicting the visual state at the end of the next segment.
- "duration": A float estimating the segment duration in seconds.

Output ONLY valid JSON. No markdown fences, no extra text.
```

System Prompt: Verification (JUDGE)

```
You are a video generation quality judge. Given frames from a video segment and the intended caption
describing what SHOULD happen, evaluate whether the segment matches the intended caption.

You must output a JSON object with exactly these fields, in this order:
- "action_match": true if the frames show the described action, false otherwise.
- "caption_match": true if the scene matches the caption description, false otherwise.
- "end_state_match": true if the intended end state has been reached, false otherwise.
- "reason": One sentence explaining your overall assessment.
- "verdict": "end" if all fields match and visual quality is acceptable, "continue" if the action is
still in progress but on track, "regenerate" if the content is wrong or quality is poor.

Output ONLY valid JSON. No markdown fences, no extra text.
```

System Prompt: Specification Refinement (REFINE)

```
You are a video generation planner revising a failed specification. You will receive the original
spec, the judge's per-field assessment, and frames from the failed generation.

Rules:
1. ONLY revise fields marked NEEDS FIX. Copy passing fields verbatim.
2. Stay close to the original intent. Make minimal, targeted changes.
3. If the action was cut short, increase the duration.
4. If the end state was not reached, make it more specific or achievable.

You must output a JSON object with these fields:
- "action_label": A short imperative phrase (5-10 words).
- "caption_delta": 1-2 sentences describing the intended scene change.
- "end_state": One sentence describing the intended end state.
- "duration": A float estimating the duration in seconds.

Output ONLY valid JSON. No markdown fences, no extra text.
```

Figure 12: **System Prompts.** The four prompting modes cover initial specification from a frame or global caption, next-segment prediction from history context, verification of generated frames against the intended specification, and refinement of failing fields.

Specification Refinement In a few-step distilled generator, noise re-sampling yields limited diversity, so naive re-generation brings only marginal gains (Table 2(d1) and (d2)). We instead achieve diversity through *conditioning change*: when the verifier returns $d_t = \text{RE-GENERATE}$ with per-field flags (m^a, m^c, m^e), the failing fields are annotated with `(NEEDS FIX)` and passed to the REFINE prompt. The planner modifies only those fields, providing the generator with a different text conditioning signal. This loop runs up to $K{=}2$ times without any additional training.

B.3.2 System and user Prompts

Figure 12 presents the complete system prompts for all four modes, and Figure 13 shows a complete predict prompt with the history structure, and a specification refinement input.

(a) Predict User Prompt (Segment t=5)

```
[System] You are a video prediction assistant.  ...
[User]
Main character:  A muscular warrior wearing iron plate armor with a fur-lined cloak, carrying a large
battleaxe.
_____________________________________________________________________________________________
Segment 0 (3.5s) [Sink]   <image> <image>
Action:  Approach the frost giant
Caption:  The warrior walks steadily across a snowy mountain pass toward a towering frost giant.
End state:  The warrior stands 10 meters from the giant, weapon raised.
_____________________________________________________________________________________________
Context summary (segments 1 to 2)
- Completed:  3 segments
- Story so far:  Approach the frost giant → Strike with axe → Dodge counterattack
- Current scene:  The warrior leaps sideways as the giant's fist slams into the snow
- Last end state:  The warrior lands in a crouch beside the giant, unharmed
_____________________________________________________________________________________________
Segment 3 (3.1s) [Window]   <image> <image>
Action:  Reposition behind rocks
Delta:  The warrior rolls behind a cluster of snow-covered boulders.
End state:  The warrior is crouching behind rocks, catching breath.
_____________________________________________________________________________________________
Segment 4 (2.8s) [Window]   <image> <image>
Action:  Charge forward from cover
Delta:  The warrior bursts out from behind the rocks and sprints toward the staggering giant.
End state:  The warrior is running at full speed with weapon raised.
_____________________________________________________________________________________________
Based on the history above, predict what happens in the NEXT segment (segment 5).  Output ONLY valid
JSON.
```

(b) Spec Refinement User Prompt

```
[System] You are a video generation planner revising a failed specification.  ...
[User]   <image> <image> (frames from the failed generation)
Original spec:  {"action_label":  "Strike the giant with a sweeping blow", "caption_delta":  "The
warrior swings the axe in a wide arc toward the giant's torso", "end_state":  "The giant staggers
backward", "duration":  3.0}
Judge assessment:
- action_match:  true
- caption_match:  false (NEEDS FIX)
- end_state_match:  false (NEEDS FIX)
- reason:  The frames show a swing motion but it misses the target entirely.
Revise the spec to address the NEEDS FIX fields.  Output ONLY valid JSON.
```

Figure 13: **Prompt Examples.** (a) The predict prompt with three-zone history: sink (seg-0), context summary (compressed), and window (two most recent segments). (b) A spec refinement input showing per-field pass/fail annotations from the verifier.

B.3.3 Reward and Training

GRPO Reward Panel The VLM is trained with GRPO under task-specific outcome reward models, which are listed in Table 3 and instantiate Equation 4.

The format reward is a binary indicator $R_{\text{fmt}}(Y) = 0.2 \cdot \mathbf{1}[\text{valid}(Y)]$. The field-match reward for INITIAL is a weighted BLEU-1:

$$R^{\text{INITIAL}}_{\text{match}} = 0.3\, B_1(a, a^\star) + 0.3\, B_1(c^{\text{abs}}, c^{\text{abs}\star}) + 0.2\, B_1(e, e^\star) + 0.2\, B_1(g, g^\star), \tag{12}$$

where a, c^{abs}, e, g denote action label, absolute caption, end state, and control object. For PRED, the primary signal is $R^{\text{PRED}}_{\text{match}} = B_1(a, a^\star)$. Duration is rewarded via bounded relative MAE: $R_{\text{dur}} = \max(0,\ 0.3(1 - |T - T^\star|/(T^\star + 0.5)))$. The judge-as-reward uses a frozen verifier $p_{\bar{\theta}}$:

$$R_{\text{judge-as-reward}}(Y, V_t^\star) = \begin{cases} +1.0 & \text{if } p_{\bar{\theta}}(\text{JUDGE} \mid Y, V_t^\star) = \text{END}, \\ -0.5 & \text{if } p_{\bar{\theta}}(\text{JUDGE} \mid Y, V_t^\star) = \text{RE-GENERATE}, \\ 0.0 & \text{otherwise.} \end{cases} \tag{13}$$

Training Configuration Table 4 summarizes both training stages. SFT serves as a format prior (loss saturates within a few hundred steps). GRPO converges within 400 to 600 steps, with early stopping to prevent verdict collapse.

Table 3: **GRPO Reward Panel.** Each ORM activates only for its task and returns 0 otherwise.

Reward	Task	Scale	Description
`format`	all	$[0, 0.2]$	JSON schema validity
`caption_match`	INITIAL	$[0, 1.0]$	Weighted BLEU-1 across 4 fields
`duration_mae`	INITIAL/PRED	$[0, 0.3]$	Duration prediction accuracy
`action_match`	PRED	$[0, 1.0]$	Action label BLEU-1
`judge_as_reward`	PRED	$[-0.5, 1.0]$	Frozen verifier verdict on predicted spec
`verdict`	JUDGE	$[0, 1.0]$	Verdict correctness
`field_match`	JUDGE	$[0, 0.3]$	Per-field bool accuracy
`regen_penalty`	JUDGE	$[-0.2, 0]$	False-positive re-generate penalty

Table 4: **VLM Training Configuration.**

	SFT (Format Prior)	**GRPO (RL Post-Training)**
Base	Qwen3-VL-2B-Instruct	SFT checkpoint
Adapter	LoRA rank 16, α=32	LoRA rank 16, α=32
Data	273K (init 11% / pred 28% / judge 61%)	60K prompts (init 20% / pred 32% / judge 48%)
Epochs / Steps	1 epoch	600 steps (early stop at 400 to 600)
LR	1e-4 (cosine)	5e-6 (constant)
Group size	–	8
KL coeff.	–	0.01

C Additional Results

We provide more visualization results for comparison and evaluation in Figure 14 and 15.

D Limitations and Future Work

REGEN decouples reasoning from generation in video world modeling through an agentic framework, enabling autonomous long-horizon simulation that open-loop systems struggle to sustain. However, several limitations remain. Firstly, planning quality is bounded by the capability of the underlying VLM: complex multi-step causal chains with subtle preconditions can produce specifications that are schema-valid but physically implausible, which exhausts the retry budget or, once accepted, propagates into long-horizon drift. Out-of-distribution action chains beyond our annotation coverage can amplify this effect. A systematic OOD evaluation across action vocabularies, viewpoints, and physical regimes, as well as physics-level evaluation in simulators with ground-truth states, is left to future work. Beyond planning fidelity, the closed-loop verification and re-generation mechanism adds inference overhead proportional to the retry budget K, limiting throughput in latency-sensitive settings. More broadly, the agentic framework introduces a structural barrier to real-time generation, breaking the continuous pipeline that streaming causal DiTs otherwise enable. Achieving real-time rollout will require non-trivial asynchronous engineering, such as overlapping VLM planning with DiT generation, speculative execution of the next segment before the current one is verified, or distilling verification into a lightweight head running alongside the generator.

Failure Attribution We attribute each failure to the earliest module at which the error becomes observable, and to the intervention that recovers it.

- **Planner failures.** The specification is schema-valid but omits a necessary precondition or conflicts with the accepted world state, so it is infeasible before generation starts. Resampling under the same specification cannot resolve such inconsistencies, while editing the action or end state can.
- **Generator failures.** The specification is feasible, but the segment does not complete the requested action, contact, or state transition. A fresh sample under the same specification can succeed, indicating stochastic execution errors rather than planning errors.
- **Verifier failures.** The verdict disagrees with the visible match between segment and specification. False acceptance is the most harmful case, since an incorrect end state enters the accepted history and propagates, whereas false rejection only costs computation.



Figure 14: **Visualization results** for baselines and REGEN under the same global caption (top) and per-segment action labels (bottom).



Figure 15: **Visualization results** for baselines and REGEN under the same global caption (top) and per-segment action labels (bottom).

Table 2(d1) separates the first two cases: resampling under a fixed specification recovers part of the generator failures, while field-level refinement further improves caption alignment and semantic drift. Figure 6(c) shows that planning failures concentrate on the end state, followed by caption delta and action label, while duration rarely fails, so the resulting world state is the hardest field to predict.

Several directions are worth pursuing. The plug-and-play schema design means a stronger VLM backbone directly improves both planning and verification quality without retraining the DiT, so future VLM advances translate naturally to better rollout. A more principled fix would be to apply GRPO jointly to the VLM planner and the DiT generator under the shared contract reward, enabling fully cooperative optimization rather than sequentially trained components. Beyond these improvements, extending the framework to embodied and robotic settings is a natural next step: in those domains, ground-truth physical state observations can replace VLM-based verification, providing stronger and more reliable supervision for both planning and generation.

E Broader Impacts

REGEN enables long-horizon, action-grounded video simulation, with direct benefits for embodied AI research, interactive world modeling, and video content creation. The primary risk is that improved generative fidelity lowers the barrier to creating deceptive synthetic videos. We mitigate this by applying safety filtering throughout dataset curation. The action-state contract structure also makes generated content more auditable than end-to-end black-box synthesis, which may assist forensic detection. The dataset is built from license-screened public and internal sources.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction claim three contributions that are all present in the paper: an agentic framework with a structured action specification protocol shared across planning, generation, and verification (Section 3), a large-scale action-grounded video dataset of 400K videos and 2M segments (Section 4), and action-grounded reward alignment (Section 3.4). Experimental claims are scoped to the metrics reported in Section 5.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix D, covering planning failures on complex multi-step causal chains, the lack of systematic OOD and physics-level evaluation, the inference overhead of the re-generation loop, and a module-level failure attribution.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper contains no theoretical results or proofs.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5.1 describes model choices, training protocol, and inference hyperparameters. The complete rollout algorithm is given in Algorithm 1 (Appendix B). Further training details are in Appendix B.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Part of the training data comes from internal and license-restricted sources.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.1 covers model architecture, LoRA rank, denoising steps, and retry budget. Full hyperparameters are in Appendix B.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported because the primary experiments involve long-video generation with large DiT and VLM models, where multiple full runs would be prohibitively expensive. This is consistent with standard practice in the video generation literature.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The VLM planner and verifier are trained on 48 H20 GPUs for about 6 hours, and the DiT generator is trained on 64 H20 GPUs for about 5 days.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics. The dataset is curated from license-screened sources and filtered for quality and safety.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Appendix E. Positive impacts include enabling richer interactive simulation for embodied AI and content creation. Potential negative impacts include the risk of misuse for generating deceptive video content (deepfakes), which we mitigate through multi-stage safety filtering in data curation.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The dataset curation pipeline (Section 4) includes multi-stage safety filtering, and misuse risks are discussed in Appendix E.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All pretrained models (Wan2.1-T2V-14B, Qwen3-VL-2B, Qwen3.5-397B-A17B, Gemini 3) and external datasets are cited.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The action-grounded video dataset is documented in Section 4 and Appendix A, covering the curation pipeline, annotation schema, and dataset statistics.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This paper does not involve crowdsourcing or research with human subjects.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

Answer: [Yes]

Justification: VLMs are a core and non-standard component of the method. Qwen3-VL-2B serves as the planner and verifier (Section 3.3), Qwen3.5-397B-A17B and Gemini 3 are used for dataset annotation (Section 4), and a frozen copy of the trained Qwen3-VL-2B verifier serves as the training reward judge (Section 3.4). All usages are described in detail in the respective sections.