# STATEFLOW: BUILDING, EVOLVING, AND ACCESSING 3D WORLD STATES FOR PREVISUALIZATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Previsualization bridges ideas and production in film, games, architecture, and urban design by letting creators iteratively refine scenes, actions, cameras, and spatial-temporal dynamics. Existing generative methods instead jointly control these factors through one-shot image or video synthesis, providing limited support for local editing, long-horizon evolution, and scene revisitation. Revisiting and selectively editing interacting elements requires an explicit, persistent world state that separates scene structure, evolution, and observation. We present **StateFlow**, a state-centric framework built around an editable 3D world with three components. State Construction generates complementary front-view and bird's-eye references, matches their object evidence, and lifts the matched objects into a spatially grounded initial 3D state under geometric and physical priors. State Evolution converts user intent into selective state transitions while preserving unaffected content and accumulated world history. State Access closes the loop between camera proposals and rendered feedback to obtain feasible views of the evolving world. Experiments on complex dynamic scenes demonstrate spatial-temporal consistency across long sequences and multi-object interactions, together with high-quality video creation and game-like prototyping.

## 1 INTRODUCTION

Previsualization is a fundamental step in filmmaking, game development, and architectural or urban design. Before final production, creators repeatedly block scenes, stage actions, test cameras, and revise spatial-temporal dynamics to communicate intent and compare alternatives (Ardal et al., 2019; Christie et al., 2008; Halperin & Lukin, 2025; Lino et al., 2011). Its central requirement is therefore not a polished final clip, but a reusable world that remains editable throughout this process.

Recent image and video generators produce increasingly realistic visual content (Rombach et al., 2022; Brooks et al., 2024; Yang et al., 2024c; Wan et al., 2025; Bruce et al., 2024), yet expose only rendered observations. Scene structure, motion, and cameras remain entangled in pixels, so a local request may trigger global regeneration without preserving object identity, unaffected content, spatial relations, or earlier edits. Previsualization requires these properties to remain consistent across successive edits and observations. Entities may leave the view, undergo changes, and later be revisited; local updates and camera requests must remain compatible with the same evolving world. The underlying challenge is therefore to maintain what persists, specify what changes, and determine how the resulting state is observed.

We formulate generative previsualization as *persistent 3D world-state modeling*. At time $t$, an addressable state records scene entities, their geometry and poses, semantic attributes, relations, and physical constraints. A user instruction updates the relevant entities under task constraints, while camera access produces an observation of the resulting state. This shared state supports successive edits and different views while retaining the effects of prior operations, providing a reusable basis for the previsualization process.

Such a state must satisfy two coupled requirements. *Temporal persistence* preserves identity and accumulated changes across edits, occlusion, and revisitation. *Spatial consistency* maintains coherent spatial relations among entities and ensures that their placement, motion, and camera access respect the geometry and physical constraints of the instantiated world. Existing systems can assemble co-



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

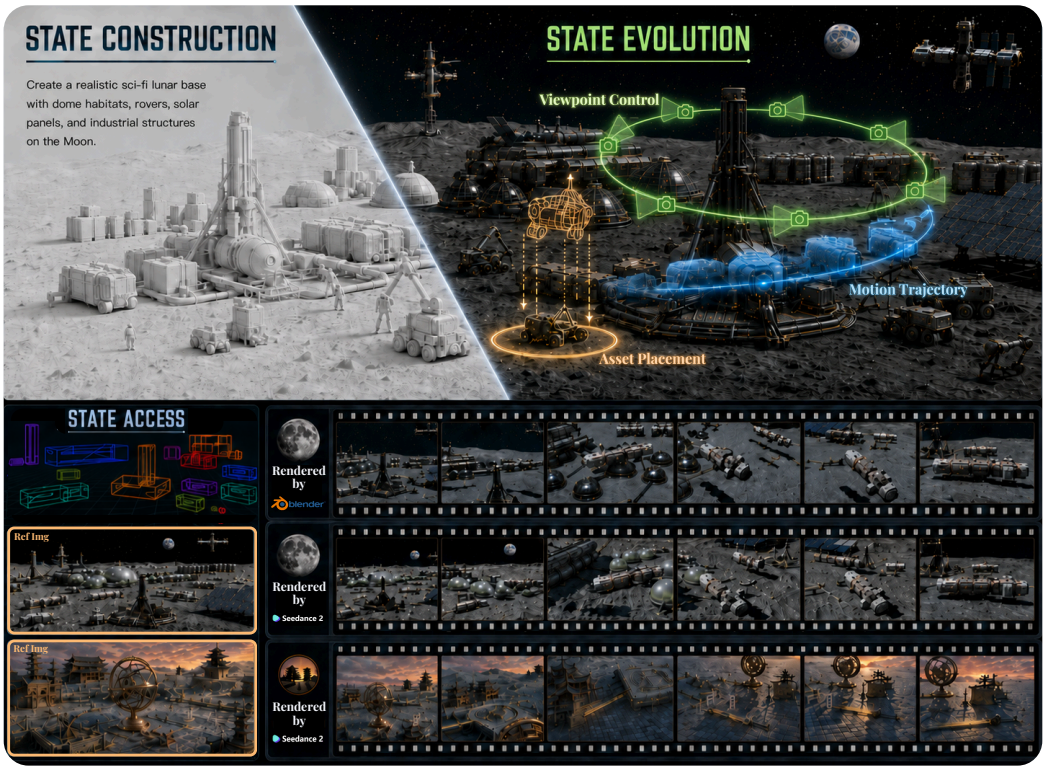


Figure 1: StateFlow integrates State Construction, State Evolution, and State Access to build, edit, and explore 3D worlds for video creation, shot planning, and game-like prototyping.

herent multi-object scenes (Yang et al., 2024b; Feng et al., 2023; Wang et al., 2025; Shi et al., 2025b; Engstler et al., 2025), but usually treat construction as the endpoint rather than supporting selective evolution and repeated, state-consistent access.

We present **StateFlow**, a state-centric framework that operationalizes this lifecycle through **State Construction**, **State Evolution**, and **State Access**. As illustrated in Fig. 1, the three stages address complementary questions: how to initialize a coherent 3D state from conflicting generative evidence, how to apply intent-driven changes while preserving world history, and how to turn semantic camera intent into geometrically feasible observations.

**State Construction** introduces *Conflict-Aware Dual-View Initialization*. A front view supplies appearance and object cues, whereas a bird's-eye view supplies global spatial evidence. Semantic and physical priors resolve layout and grounding errors and object collisions to initialize a coherent, object-centric 3D state. **State Evolution** introduces *Structured State Transition*. It converts an instruction into a selective update plan over addressable entity records, updating geometry, pose, or semantic status while preserving non-targeted content. Applying changes to persistent records allows successive updates to accumulate without resampling unrelated entities. The same interface supports scene expansion, object motion, local style changes, and discrete events within the evolving world. **State Access** introduces *Render-Feedback Camera Planning*. A vision-language model proposes a trajectory, but the instantiated world verifies it through rendering. Visibility, occlusion, framing, collision, and motion errors trigger targeted repairs, closing a proposal–verification loop without training a camera policy.

Together, these stages maintain one world that can be initialized, selectively updated, and repeatedly observed. Off-the-shelf video models (Seedance et al., 2026) may enhance its rendered observations without altering the underlying state. Experiments on cinematic video creation and 3D scene prototyping demonstrate coordinated multi-object motion, adherence to prescribed physical trajectories, and consistent scene revisitation across clips, supporting persistent and controllable creation through successive state changes and observations.

# 2 Related Work

## 2.1 Generative Previsualization and Video Agents

Previsualization uses intermediate representations to validate scene layout, staging, camera motion, and collaboration before production. Prior work has explored real-time engines, immersive interfaces, and rapid video ideation for film, games, and design (Nitsche, 2008; Northam et al., 2012; Ardal et al., 2019; Fei & Liu, 2023; Jo et al., 2024; Ivanov et al., 2022; Yu et al., 2022). These studies establish previsualization as an iterative interface for communicating and testing creative decisions, but typically assume that the underlying scene has been manually authored. Recent video backbones (Blattmann et al., 2023; Yang et al., 2024c; Kong et al., 2024; Seedance et al., 2026; Wan et al., 2025) offer powerful short-clip synthesis, while agentic systems such as MovieAgent, AniMaker, AniME, VideoClaw, Toonflow, and ViMax coordinate scripts, shots, candidates, and production stages (Wu et al., 2025; Shi et al., 2025a; Zhang et al., 2025; HITsz-TMG, 2026; HBAI-Ltd, 2026; Huang et al., 2026). These systems improve long-form planning and clip consistency, but their persistent structure is primarily organized around scripts or rendered shots. StateFlow instead makes an explicit 3D world state the reusable substrate for editing, evolution, and camera access, allowing observations at different times and viewpoints to refer to the same evolving scene.

## 2.2 Editable 3D World Generation

Recent 3D and 4D generators synthesize objects, dynamic assets, and reconstructed worlds from images, text, or video (Tang et al., 2024; Zhao et al., 2025; Xiang et al., 2025; Lin et al., 2025; Ren et al., 2023; Yin et al., 2023; Liang et al., 2024; Xing et al., 2025; Sun et al., 2024; Yang et al., 2025b; Yin et al., 2025; Team et al., 2026; Li et al., 2026; Sun et al., 2025; Zhang et al., 2026). Scene-level methods either use procedural rules and curated assets (Raistrick et al., 2023; 2024), infer layouts before retrieving scene elements (Yang et al., 2024b;a; Feng et al., 2023; Zhu et al., 2025), or directly generate scene content (Team et al., 2025; Wang et al., 2025; Shi et al., 2025b; Huang et al., 2025; Yang et al., 2026; 2025a; Engstler et al., 2025). Although increasingly diverse and realistic, these methods mainly optimize initial construction or a fixed dynamic sequence; procedural and retrieval-based systems are additionally bounded by predefined rules or asset coverage. They do not jointly maintain addressable entities, accumulated state changes, and geometry-grounded camera access. StateFlow connects these capabilities through one persistent, structured 3D world.

# 3 Method

## 3.1 Previsualization Formulation and Overview

StateFlow models previsualization as operations on a persistent world rather than as independent image or video generation. Given input conditions $\mathcal{C}$, the system constructs an initial state, evolves it over time, and accesses it under controllable viewpoints:

$$\mathcal{W}^0 = \mathcal{F}_{\text{build}}(\mathcal{C}), \quad \mathcal{W}^{t+1} = \mathcal{F}_{\text{evolve}}(\mathcal{W}^t), \quad y^t = \mathcal{F}_{\text{access}}(\mathcal{W}^t). \tag{1}$$

Here, $\mathcal{W}^t$ is the reusable world state and $y^t$ is a rendered or interactive observation. Separating the state, its transitions, and its observations allows edits and camera requests to reuse the same world.

We instantiate $\mathcal{W}^t$ as an object-centric structured 3D state:

$$\mathcal{W}^t = \{o_i^t\}_{i=1}^{N_t}, \quad o_i^t = (g_i^t, p_i^t, s_i^t). \tag{2}$$

Each object $o_i^t$ stores geometry $g_i^t$, position and pose $p_i^t$, and editable semantic attributes $s_i^t$. As shown in Fig. 2, State Construction initializes $\mathcal{W}^0$, State Evolution selectively updates its object records, and State Access produces camera-controlled observations without replacing the world.

## 3.2 State Construction

Given input conditions $\mathcal{C}$, State Construction initializes $\mathcal{W}^0$. A single view cannot reliably determine both object appearance and scene layout: a front view provides appearance and state cues but hides depth and occluded objects, whereas a bird's-eye view (BEV) exposes planar layout but provides

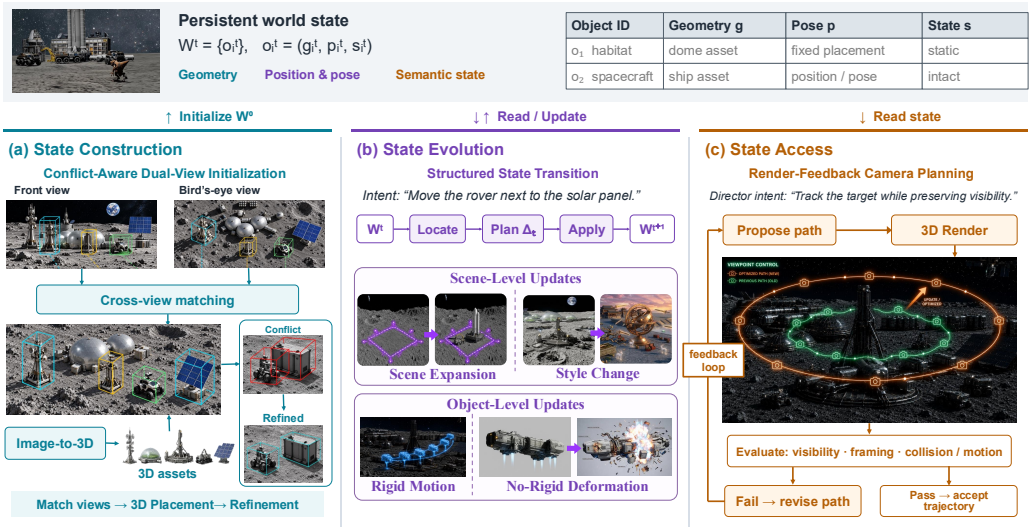


Figure 2: StateFlow constructs, evolves, and accesses a persistent 3D world. Conflict-Aware Dual-View Initialization reconciles appearance, layout, and physical constraints. Structured State Transition supports expansion, restyling, motion, and events, preserving continuity. Render-Feedback Camera Planning improves visibility, framing, and motion feasibility.

weaker appearance evidence. We therefore introduce **Conflict-Aware Dual-View Initialization**, which treats the two views as complementary constraints and resolves their disagreements before committing a 3D state.

We generate a front view followed by a BEV conditioned on the front view and the same prompt. A VLM matches front-view instances to BEV boxes. For each match, the front-view crop drives image-to-3D asset generation (Zhao et al., 2025; Xiang et al., 2025), the BEV box supplies horizontal placement and a coarse footprint, and the generated asset supplies geometry and 3D proportions. The VLM then determines whether the object is grounded or floating and assigns its elevation from the scene context. This division lets each view control only the properties it represents reliably.

The assembled scene is checked against the prompt, scene boundaries, and physical intersections. The verifier identifies the conflicting objects and returns a textual diagnosis; refinement updates only their placements while retaining accepted assets and the coarse layout. This selective repair loop prevents cross-view errors from propagating without regenerating the scene or training a domain-specific 3D detector. The complete procedure is provided in Appendix A.1.

The resulting geometry $g_i^0$, pose $p_i^0$, and semantic attributes $s_i^0$ form the initial world state:

$$\mathcal{W}^0 = \{o_i^0\}_{i=1}^{N_0}, \qquad o_i^0 = (g_i^0, p_i^0, s_i^0). \tag{3}$$

The resulting $\mathcal{W}^0$ is editable, spatially grounded, and ready for subsequent evolution and access.

### 3.3 State Evolution

Given the current world state $\mathcal{W}^t$, State Evolution takes an intent $u^t$ and produces an updated state $\mathcal{W}^{t+1}$. The intent can be a direct user instruction or a director intent board from which the VLM extracts an event. Our objective is to realize the requested change while preserving as much of the existing world as possible. We therefore introduce **Structured State Transition**, which treats evolution as a sparse update to the persistent state table rather than regenerating the entire scene.

The VLM first parses the intent into an event description $e_t$ that specifies what should happen, which entities are involved, and the desired target state. It then locates the corresponding object records and state fields in $\mathcal{W}^t$, producing an affected index set $\mathcal{I}_t$. Based on these records, the VLM predicts a compact transition plan $\Delta_t$, which the system applies only to the selected fields. For example, an instruction to move a spaceship is parsed as a translation event, linked to the spaceship record, and applied to its position and pose state. Its geometry, appearance, and all unrelated object records remain unchanged. At the scene level, an expansion event invokes State Construction only for the new region and merges the new records into the existing state. A style-change event instead updates

the relevant scene- or object-level style attributes while preserving geometry and layout. Additional task instantiations are provided in Appendix A.2.

Formally, the transition follows four steps:

$$\begin{aligned} e_t &= \mathrm{Parse}_{\mathrm{VLM}}(u^t), & \mathcal{I}_t &= \mathrm{Locate}(\mathcal{W}^t, e_t), \\ \Delta_t &= \mathrm{Plan}_{\mathrm{VLM}}(\mathcal{W}^t_{\mathcal{I}_t}, e_t), & \mathcal{W}^{t+1} &= \mathrm{Apply}(\mathcal{W}^t, \Delta_t). \end{aligned} \tag{4}$$

Here, $\Delta_t$ records both the target entities and the required changes to their geometry $g_i^t$, position and pose $p_i^t$, or semantic attributes $s_i^t$. Records outside $\mathcal{I}_t$ are copied directly from $\mathcal{W}^t$ to $\mathcal{W}^{t+1}$. This selective update preserves object identity, accumulated layout, and prior edits while allowing the same world to evolve through a sequence of instructions.

### 3.4 State Access

Given the dynamic world state $\mathcal{W}^t$, the state access stage determines how the constructed world is used for downstream tasks. While a world state can be accessed through editing commands, physical simulation, or user control, camera trajectories provide a general interface for previsualization: they convert the same persistent world into film-like keyframes, explorable videos, and game-like viewpoints without modifying the world itself. However, previous text-only camera generation (Liu et al., 2024) mainly relies on linguistic intent and lacks geometric feedback from the actual scene. To address this, we combine the semantic reasoning of a VLM with geometric evidence from rendered observations, leading to **Render-Feedback Camera Planning**.

We first extract the information needed for camera planning from $\mathcal{W}^t$, including the important objects, their poses and bounding boxes, and the overall scene extent. Canonical rendered views show how these objects appear from different directions. We then divide the user instruction into segment-level director intents $d_i$, where each intent specifies the target object, desired shot style, and camera motion for one segment.

For each segment, the VLM acts as a semantic proposer rather than the final controller. It generates an initial trajectory from the world state, rendered scene observations $V$, and director intent:

$$\pi_i^0 = \mathrm{Propose}_{\mathrm{VLM}}(\mathcal{W}^t, V, d_i). \tag{5}$$

Here, $\pi_i^k$ denotes the $k$-th camera trajectory proposal for segment $i$, represented as a sequence of camera positions, orientations, and viewing parameters. We execute the proposed trajectory in the 3D world using low-cost rendering and evaluate the rendered result:

$$R_i^k = \mathrm{Render}(\mathcal{W}^t, \pi_i^k), \qquad f_i^k = \mathrm{Eval}(R_i^k, d_i, \mathcal{W}^t). \tag{6}$$

The evaluator returns $f_i^k$ as natural-language feedback. It either marks the trajectory as accepted or describes the remaining problems, such as a hidden target, poor framing, a camera collision, or jerky motion. When revision is needed, we provide the current trajectory, its rendered result, and this textual feedback to the VLM, which produces the next trajectory proposal. We repeat the same propose–render–feedback process until the trajectory is accepted or five rounds have been completed. The accepted trajectory, or the latest proposal at the fifth round, is used as $\pi_i^\star$. This simple feedback loop remains training-free and requires no camera trajectory annotations. The final trajectory supports keyframe generation, free-view exploration, and third-person views while remaining grounded in the same persistent 3D world state.

## 4 Experiments

### 4.1 Experimental Setup

Our system takes a single text prompt and automatically completes the workflow from 3D world construction to downstream export. We use Gemini 3.1 (Google, 2026a) for scene understanding, intent parsing, layout reasoning, transition planning in *Structured State Transition*, and camera planning in *Render-Feedback Camera Planning*. We use Nano Banana 2 (Google, 2026b) for front-view and BEV references. Hunyuan3D (Team, 2025) generates object assets. Seedance2.0 (Seedance et al., 2026) enhances the videos. For VBench, five methods each generate one video for each of 30

scene prompts. We evaluate complete videos at native resolution with VBench v0.1.5 (Huang et al., 2024), dividing Imaging Quality by 100 before aggregation. Our user study recruits 50 participants with 12 prompts per setting: 45° BEV fly-around videos for scene evaluation and final videos for video evaluation. Results are anonymized and randomized. We report mean scores. Full Gemini and GPT-5.4 evaluations are provided in the supplementary material.

### 4.2 State Construction: Layout and Physical Validity

We compare StateFlow with representative 3D generation baselines: training-free world generation with SynCity (Engstler et al., 2025), feed-forward image-to-3D generation with SAM3D (Team et al., 2025), and part-aware object generation with PartCrafter (Lin et al., 2025).

Table 1: State Construction: structural validity and visual quality. Q-Align reports quality/aesthetics.

| | Structure and physical validity | | | Visual alignment and quality | | | |
|---|---|---|---|---|---|---|---|
| Method | Layout IoU ↑ | Floating ↓ | Collision ↓ | CLIP-I ↑ | CLIP-T ↑ | HPS V2 ↑ | Q-Align ↑ |
| SAM3D | 50.5% | 65.0% | **0.0%** | 0.580 | 15.481 | 0.055 | 2.262/1.137 |
| PartCrafter | 46.0% | 16.3% | 12.5% | 0.542 | 20.761 | 0.110 | 1.675/1.587 |
| SynCity | 44.9% | **0.0%** | **0.0%** | 0.689 | 22.880 | **0.175** | 3.535/**2.966** |
| **StateFlow** | **93.7%** | **0.0%** | **0.0%** | **0.788** | **30.214** | 0.151 | **3.621**/2.859 |

We match objects by semantic identity to VLM-extracted BEV boxes used as pseudo ground truth. Layout IoU measures mean top-down footprint overlap. Floating is the fraction of grounded boxes not contacting their support surface, and collision is the fraction of non-supporting object pairs with intersecting boxes. StateFlow achieves the highest layout IoU and CLIP-T, with zero floating and collision errors under this protocol (Table 1). Since these examples have no ground-truth 3D scenes, we also render 45° BEV videos and compute CLIP-I/CLIP-T (Radford et al., 2021), HPS (Wu et al., 2023b), and Q-Align (Wu et al., 2023a). Table 1 shows clear advantages in CLIP-I, CLIP-T, and Q-Align quality. We note that HPS and Q-Align tend to exhibit strong preferences toward colorfulness and visual style, which may lead to relatively high scores for SynCity. Figure 3 shows more coherent global spatial organization while maintaining object-level details.

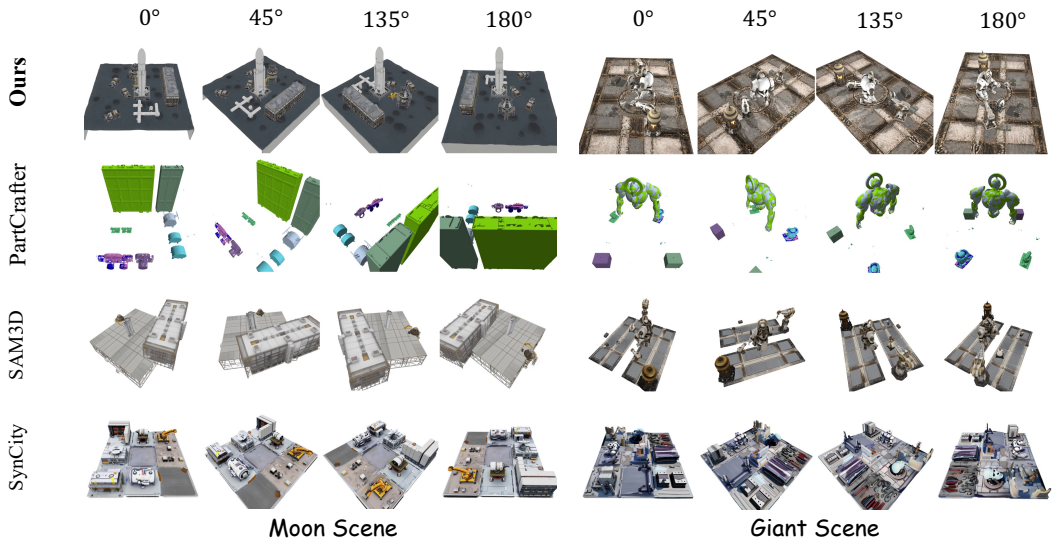


Figure 3: Qualitative comparison on scene generation. StateFlow preserves global spatial organization and object-level details across viewpoints.

**Construction ablations.** Figure 5 visualizes the effects of removing key components. Without BEV layout, object proportions and spatial arrangements become distorted. Without conflict resolution, the assembled scene exhibits object interpenetration and floating artifacts. In the user study (Table 5), layout plausibility decreases from 4.6 to 3.0 and 3.2, respectively. These variants verify the importance of BEV grounding and conflict-aware reconciliation.

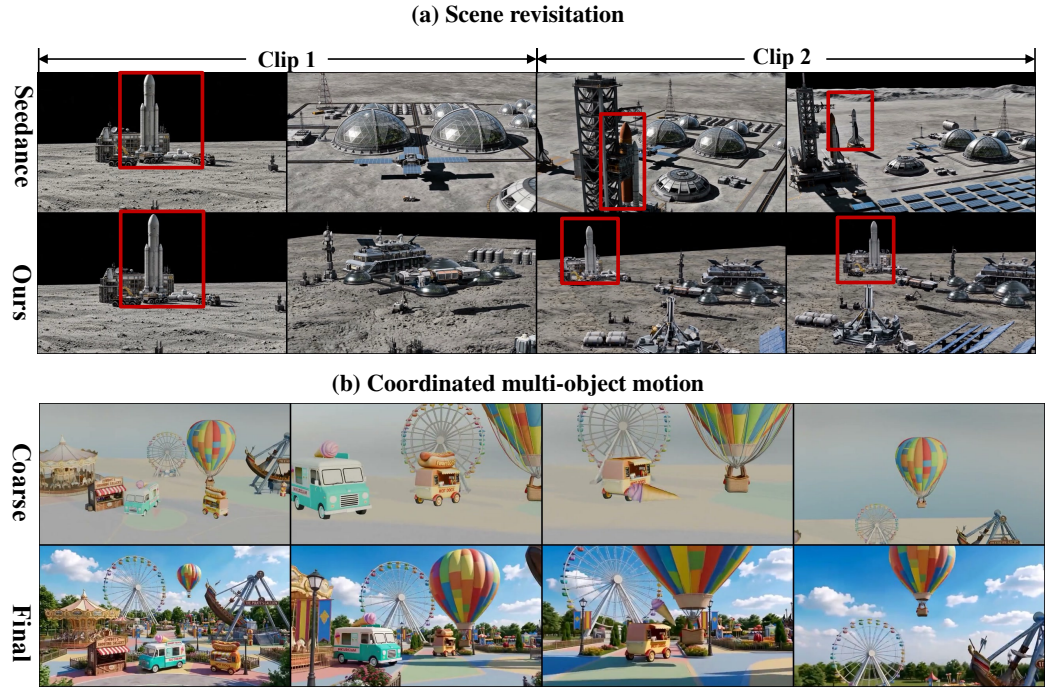


Figure 4: **Persistence and controlled evolution.** (a) Scene revisitation. Highlighted boxes track the rocket across clips: Seedance2.0 exhibits appearance drift, while StateFlow preserves its identity and style. Columns show Scene A, the end of Clip 1 in Scene B, and two later Clip 2 frames returning toward A. (b) Coordinated multi-object motion, with coarse scene renders above final video frames at matched normalized times. Columns proceed from early to late.

### 4.3 State Evolution: Persistence and Controlled Changes

To isolate the effect of the persistent state table, we compare StateFlow with a full-regeneration variant. For each instruction, this variant edits a fresh reference image and reconstructs the entire scene rather than locating the affected record and applying a minimal update. Removing the state table reduces VBench Subject Consistency by 0.28 and Background Consistency by 0.17 (Table 2). Full regeneration can satisfy a new instruction, but also resamples previously accepted content, causing identity and background drift.

Figure 4 illustrates our capabilities. Scene revisitation preserves spatial structure across clips, while coordinated multi-object motion demonstrates how the final video follows the evolving scene state. The physical-motion example in Appendix D applies trajectories computed from prescribed physical equations. Together, these examples illustrate persistent observations and controlled changes within the same world. The state-table ablation tests the role of selective updates across edits.

### 4.4 State Access: Intent Alignment and Camera Feasibility

We compare our render-feedback camera planner with ChatCam (Liu et al., 2024). Intent match is the fraction of video seconds satisfying the requested camera intent, judged by Gemini. Target visibility is the fraction of frames where the target is visible, tracked with SAM2. Collision is the number of colliding camera–object pairs divided by the number of scene objects, computed geometrically. Acceleration measures the trajectory's second derivative, with lower values indicating smoother motion. StateFlow improves every metric (Table 3): each proposed trajectory is executed, inspected in the actual world, and revised from textual render feedback.

**Render-feedback ablation.** *VLM-only Camera* generates trajectories from semantic intent without *Render-Feedback Camera Planning*. Figure 5 shows misdirected framing, trajectories that pass through objects, and oversimplified camera paths. The corresponding StateFlow trajectories keep the intended subjects in view and avoid the illustrated obstructions. The VLM-only variant's camera

Table 2: State Evolution ablation. Changes relative to full StateFlow. Smaller drops are better.

| Method | Δ Subject Cons. | Δ Background Cons. |
|---|---|---|
| StateFlow | 0.00 | 0.00 |
| Full regeneration (w/o state table) | −0.28 | −0.17 |

Table 3: Camera-planning comparison. Render feedback improves intent matching, visibility, collision avoidance, and motion smoothness.

| Method | Intent Match ↑ | Visibility ↑ | Collision ↓ | Accel. ↓ |
|---|---|---|---|---|
| ChatCam | 54.8% | 67.1% | 0.07 | 4.64 |
| **StateFlow** | **89.2%** | **96.2%** | **0.00** | **2.48** |

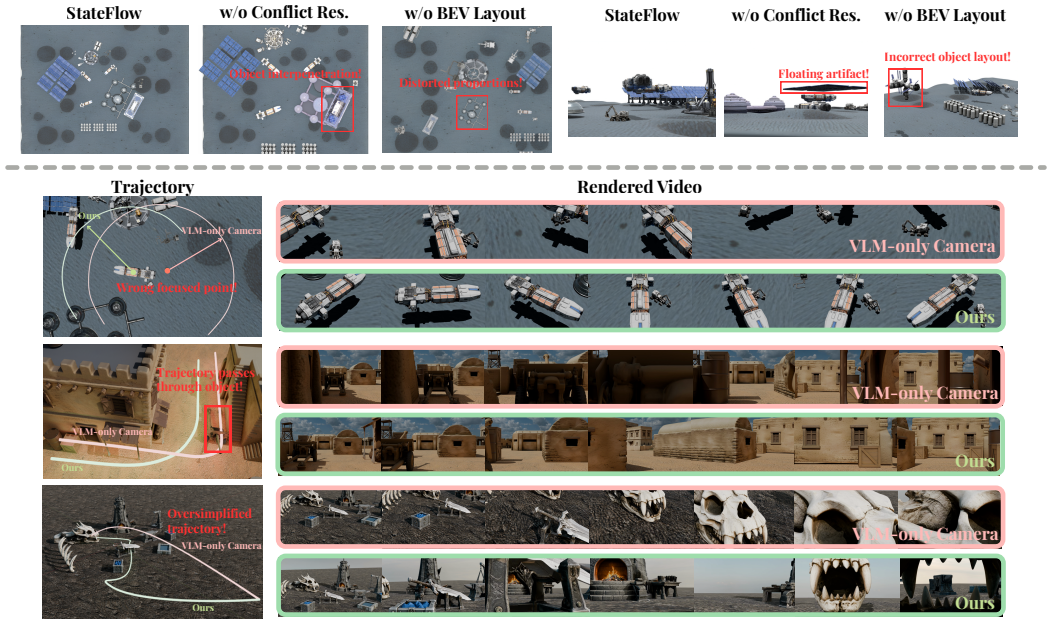


Figure 5: Qualitative ablations of State Construction and State Access.

quality falls from 4.6 to 3.4, and previsualization usefulness from 4.6 to 3.8 (Table 5), supporting the role of rendered feedback in refining trajectories.

### 4.5 Downstream Previsualization Results

We compare StateFlow with AniMaker (Shi et al., 2025a), MovieAgent (Wu et al., 2025), Wan2.2 (Wan et al., 2025), and Seedance2.0 (Seedance et al., 2026), covering narrative-driven animation, agentic multi-shot generation, and text-to-video models. Table 4 reports the highest average score for StateFlow, with the best subject consistency, background consistency, motion smoothness, and flicker scores. Although some baselines achieve slightly higher scores on aesthetic or imaging quality, these metrics are more sensitive to low-level visual preferences such as colorfulness and contrast. In contrast, our method shows stronger performance on structure and temporality related dimensions, which are more critical for controllable scene generation and previsualization.

Table 4: VBench comparison. All metrics are higher-is-better.

| Method | Subject | Background | Aesthetic | Imaging | Motion | Flicker | Average |
|---|---|---|---|---|---|---|---|
| AniMaker | 0.7509 | 0.8343 | **0.6795** | 0.6684 | 0.9582 | 0.9641 | 0.8092 |
| MovieAgent | 0.7533 | 0.8557 | 0.5348 | 0.7260 | 0.9895 | 0.9858 | 0.8075 |
| Wan2.2 | 0.8836 | 0.9422 | 0.5220 | 0.6911 | 0.9852 | 0.9731 | 0.8283 |
| Seedance2.0 | 0.8110 | 0.9329 | 0.6311 | 0.7184 | 0.9828 | 0.9565 | 0.8387 |
| **StateFlow** | **0.9135** | **0.9506** | 0.5257 | 0.7185 | **0.9923** | **0.9902** | **0.8484** |
| Direct render | 0.8367 | 0.9120 | 0.6240 | **0.7627** | 0.9832 | 0.9546 | 0.8455 |

Figure 6 shows more stable scene structure and clearer spatial grounding across viewpoints.

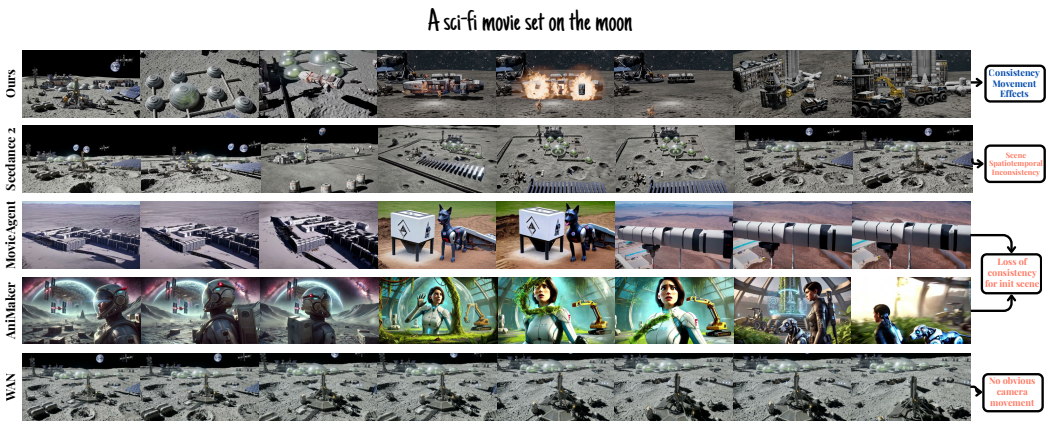


Figure 6: Qualitative comparison on the sci-fi movie set on the moon. StateFlow preserves scene structure across a long sequence and supports controllable camera motion, while video baselines exhibit inconsistency or restricted viewpoints.

**World state and video enhancement.** Seedance2.0 receives the generated front image, Blender-rendered video, and text prompt, with the instruction to keep geometry from the video and appearance from the image. Compared with the standalone Seedance2.0 baseline, the full pipeline improves Subject Consistency from 0.8110 to 0.9135 using the same video backbone. Direct rendering already achieves an average of 0.8455. Enhancement raises it to 0.8484 while lowering aesthetic and imaging scores. Thus, the benefit is strongest in consistency and temporal quality rather than uniform improvement across all visual metrics.

Table 5: User study (50 participants, 12 prompts per setting). Scores range from 1 to 5, with higher values indicating better performance. Selected dimensions are shown. Full user and MLLM scores appear in Table 8. "Useful" denotes previsualization usefulness.

*Scene-level evaluation*

| Method | Layout | Useful | Overall |
|---|---|---|---|
| SynCity | 2.8 | 2.7 | 3.2 |
| SAM3D | 3.1 | 3.1 | 3.5 |
| PartCrafter | 1.9 | 1.8 | 2.0 |
| w/o BEV | 3.0 | 3.1 | 3.3 |
| w/o conflicts | 3.2 | 3.2 | 3.4 |
| **StateFlow** | **4.6** | **4.7** | **4.5** |

*Video-level evaluation*

| Method | Camera | Useful | Overall |
|---|---|---|---|
| Seedance2.0 | 3.5 | 3.7 | 3.6 |
| Wan2.2 | 3.0 | 3.1 | 3.1 |
| AniMaker | 2.5 | 2.4 | 2.5 |
| MovieAgent | 1.6 | 1.5 | 1.7 |
| VLM-only | 3.4 | 3.8 | 4.1 |
| **StateFlow** | **4.6** | **4.6** | **4.5** |

**Perceptual quality and practical usefulness.** StateFlow achieves the highest user-rated overall score and previsualization usefulness in both settings (Table 5). The GPT-5.4 evaluation also gives StateFlow the highest overall score, although it rates Seedance2.0 higher on video camera quality (Table 7). Additional industrial-scene comparisons are provided in the supplementary material. The CS-style demonstration renders a persistent desert environment in Blender. Its first-person gun and hand are composited overlays rather than generated scene content (Appendix D.1).

## 5 Conclusion

StateFlow formulates generative previsualization as persistent 3D world-state modeling and realizes it through an editable object-centric representation. Conflict-Aware Dual-View Initialization reconciles appearance, layout, relations, and physical constraints. Structured State Transition updates affected records while preserving world history. Render-Feedback Camera Planning converts semantic requests into feasible observations. This separation between a reusable world and its rendered outputs supports controllable video creation and interactive 3D prototyping. Experiments across long sequences and multi-object interactions demonstrate improved spatial-temporal consistency, scene coherence, and camera control over direct visual-generation baselines. Runtime remains limited by the latency of third-party models.

### AI USE STATEMENT

Generative AI tools were used to assist with language editing and LaTeX formatting. All AI-assisted changes were reviewed by the authors, who take full responsibility for the final content of this work.

### ETHICS STATEMENT

StateFlow supports creative previsualization but may inherit model biases or enable misleading content. Users should respect asset rights and avoid non-consensual depictions. Our user study reports aggregate ratings with anonymized and randomized method outputs.

### REPRODUCIBILITY STATEMENT

The main paper and appendix describe model components, implementation details, and evaluation protocols. Reproduction requires access to third-party models, whose stochastic outputs and updates may affect the results.

## REFERENCES

Dui Ardal, Simon Alexandersson, Mirko Lempert, and André Tiago Abelho Pereira. A collaborative previsualization tool for filmmaking in virtual reality. In *Proceedings of the 16th ACM SIGGRAPH European Conference on Visual Media Production*, pp. 1–10, 2019.

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

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Marc Christie, Patrick Olivier, and Jean-Marie Normand. Camera control in computer graphics. In *Computer graphics forum*, volume 27, pp. 2197–2218. Wiley Online Library, 2008.

Paul Engstler, Aleksandar Shtedritski, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Syncity: Training-free generation of 3d worlds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27585–27595, 2025.

Guangzheng Fei and Dake Liu. Tape: Tangible augmented previz environment for filmmaking. In *Proceedings of the Eleventh International Symposium of Chinese CHI*, pp. 251–262, 2023.

Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36: 18225–18250, 2023.

Google. Gemini 3.1 Pro Preview. `https://ai.google.dev/gemini-api/docs/models/gemini-3.1-pro-preview`, 2026a. Accessed: 2026-05-12.

Google. Nano Banana 2: Google's latest AI image generation model. `https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/`, 2026b. Accessed: 2026-05-12.

Brett A Halperin and Stephanie M Lukin. From camera-eye to ai: Exploring the interplay of cinematography and computational visual storytelling. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–18, 2025.

HBAI-Ltd. Toonflow-app: Open-source ai tool for animated short drama creation. https://github.com/HBAI-Ltd/Toonflow-app, 2026. GitHub repository, accessed 2026-08-08.

HITsz-TMG. Videoclaw: Ai creative video production agent. https://github.com/HITsz-TMG/VideoClaw, 2026. GitHub repository, accessed 2026-08-08.

Lingxuan Huang, Sizhe He, Hengji Zhou, Liqiang Nie, Lianghao Xia, and Chao Huang. Vimax: Agentic video generation. *arXiv preprint arXiv:2606.07649*, 2026.

Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. Midi: Multi-instance diffusion for single image to 3d scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23646–23657, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Alexander Ivanov, Tim Au Yeung, Kathryn Blair, Kurtis Danyluk, Georgina Freeman, Marcus Friedel, Carmen Hull, Michael Yuk-Shing Hung, Sydney Pratte, and Wesley Willett. One week in the future: Previs design futuring for hci research. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2022.

Hye-Young Jo, Ryo Suzuki, and Yoonji Kim. Collagevis: Rapid previsualization tool for indie filmmaking using video collages. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2024.

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.

Longfei Li, Zhiwen Fan, Wenyan Cong, Xinhang Liu, Yuyang Yin, Matt Foutter, Panwang Pan, Chenyu You, Yue Wang, Zhangyang Wang, et al. Martian world model: Controllable video synthesis with physically accurate 3d reconstructions. *Advances in Neural Information Processing Systems*, 38, 2026.

Hanwen Liang, Yuyang Yin, Dejia Xu, Hanxue Liang, Zhangyang Wang, Konstantinos N Plataniotis, Yao Zhao, and Yunchao Wei. Diffusion4d: Fast spatial-temporal consistent 4d generation via video diffusion models. *arXiv preprint arXiv:2405.16645*, 2024.

Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. *arXiv preprint arXiv:2506.05573*, 2025.

Christophe Lino, Marc Christie, Roberto Ranon, and William Bares. The director's lens: an intelligent assistant for virtual cinematography. In *Proceedings of the 19th ACM international conference on Multimedia*, pp. 323–332, 2011.

Xinhang Liu, Yu-Wing Tai, and Chi-Keung Tang. Chatcam: Empowering camera control through conversational ai. *Advances in Neural Information Processing Systems*, 37:54483–54506, 2024.

Michael Nitsche. Experiments in the use of game technology for pre-visualization. In *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, pp. 160–165, 2008.

Lesley Northam, Joe Istead, and Craig S Kaplan. A collaborative real time previsualization tool for video games and film. In *ACM SIGGRAPH 2012 Posters*, pp. 1–1. 2012.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite photorealistic worlds using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12630–12641, 2023.

Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, Zeyu Ma, and Jia Deng. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21783–21794, June 2024.

Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Haoyuan Shi, Yunxin Li, Xinyu Chen, Longyue Wang, Baotian Hu, and Min Zhang. Animaker: Multi-agent animated storytelling with mcts-driven clip generation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025a.

Yukai Shi, Weiyu Li, Zihao Wang, Hongyang Li, Xingyu Chen, Ping Tan, and Lei Zhang. Scenemaker: Open-set 3d scene generation with decoupled de-occlusion and pose estimation model. *arXiv preprint arXiv:2512.10957*, 2025b.

Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhang, and Yikai Wang. Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion. *arXiv preprint arXiv:2411.04928*, 2024.

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.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *International Conference on Learning Representations*, volume 2024, pp. 33879–33896, 2024.

HY Team, Chenjie Cao, Xuhui Zuo, Zhenwei Wang, Yisu Zhang, Junta Wu, Zhenyang Liu, Yuning Gong, Yang Liu, Bo Yuan, et al. Hy-world 2.0: A multi-modal world model for reconstructing, generating, and simulating 3d worlds. *arXiv e-prints*, pp. arXiv–2604, 2026.

SAM 3D Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. Sam 3d: 3dfy anything in images. 2025. URL https://arxiv.org/abs/2511.16624.

Tencent Hunyuan3D Team. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details, 2025. URL https://arxiv.org/abs/2506.16504.

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.

Dilin Wang, Hyunyoung Jung, Tom Monnier, Kihyuk Sohn, Chuhang Zou, Xiaoyu Xiang, Yu-Ying Yeh, Di Liu, Zixuan Huang, Thu Nguyen-Phuoc, et al. Worldgen: From text to traversable and interactive 3d worlds. *arXiv preprint arXiv:2511.16825*, 2025.

Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090*, 2023a.

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

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2096–2105, 2023b.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

Ke Xing, Hanwen Liang, Dejia Xu, Yuyang Yin, Konstantinos N Plataniotis, Yao Zhao, and Yunchao Wei. Tip4gen: Text to immersive panorama 4d scene generation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 9267–9276, 2025.

Yixuan Yang, Junru Lu, Zixiang Zhao, Zhen Luo, James JQ Yu, Victor Sanchez, and Feng Zheng. Llplace: The 3d indoor scene layout generation and editing via large language model. *arXiv preprint arXiv:2406.03866*, 2024a.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16227–16237, 2024b.

Zhifei Yang, Keyang Lu, Chao Zhang, Jiaxing Qi, Hanqi Jiang, Ruifei Ma, Shenglin Yin, Yifan Xu, Mingzhe Xing, Zhen Xiao, et al. Mmgdreamer: Mixed-modality graph for geometry-controllable 3d indoor scene generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9391–9399, 2025a.

Zhifei Yang, Guangyao Zhai, Keyang Lu, YuYang Yin, Chao Zhang, Zhen Xiao, Jieyi Long, Nassir Navab, and Yikai Wang. Flowscene: Style-consistent indoor scene generation with multimodal graph rectified flow. *arXiv preprint arXiv:2603.19598*, 2026.

Zhongqi Yang, Wenhang Ge, Yuqi Li, Jiaqi Chen, Haoyuan Li, Mengyin An, Fei Kang, Hua Xue, Baixin Xu, Yuyang Yin, et al. Matrix-3d: Omnidirectional explorable 3d world generation. *arXiv preprint arXiv:2508.08086*, 2025b.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024c.

Yuyang Yin, Dejia Xu, Zhangyang Wang, Yao Zhao, and Yunchao Wei. 4dgen: Grounded 4d content generation with spatial-temporal consistency. *arXiv preprint arXiv:2312.17225*, 2023.

Yuyang Yin, HaoXiang Guo, Fangfu Liu, Mengyu Wang, Hanwen Liang, Eric Li, Yikai Wang, Xiaojie Jin, Yao Zhao, and Yunchao Wei. Panoworld-x: Generating explorable panoramic worlds via sphere-aware video diffusion. *arXiv preprint arXiv:2509.24997*, 2025.

Rongrong Yu, Ning Gu, Gun Lee, and Ayaz Khan. A systematic review of architectural design collaboration in immersive virtual environments. *Designs*, 6(5):93, 2022.

Lisai Zhang, Baohan Xu, Siqian Yang, Mingyu Yin, Jing Liu, Chao Xu, Siqi Wang, Yidi Wu, Yuxin Hong, Zihao Zhang, et al. Anime: Adaptive multi-agent planning for long animation generation. In *Proceedings of the SIGGRAPH Asia 2025 Posters*, pp. 1–3. 2025.

Yisu Zhang, Chenjie Cao, Tengfei Wang, Xuhui Zuo, Junta Wu, Jianke Zhu, and Chunchao Guo. Worldstereo: Bridging camera-guided video generation and scene reconstruction via 3d geometric memories. *arXiv preprint arXiv:2603.02049*, 2026.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

Xiaoming Zhu, Xu Huang, Qinghongbing Xie, Zhi Deng, Junsheng Yu, Yirui Guan, Zhongyuan Liu, Lin Zhu, Qijun Zhao, Ligang Liu, et al. Imaginarium: Vision-guided high-quality 3d scene layout generation. *ACM Transactions on Graphics (TOG)*, 44(6):1–24, 2025.

**Conflict-Aware Dual-View Initialization**

**Input:** prompt $\mathcal{C}$; image and image-to-3D generators; VLM; physical checker
**Output:** initial structured world state $\mathcal{W}^0$

Generate front view $I_{\text{front}}$ from $\mathcal{C}$.
Generate BEV $I_{\text{bev}}$ conditioned on $(I_{\text{front}}, \mathcal{C})$.
Detect front-view instances, predict BEV boxes, and match identities across the two views.
**for each** matched object $i$ **do**
    Generate a clean front-view crop and lift it to geometry $g_i^0$.
    Initialize horizontal placement and footprint from the BEV box.
    Infer grounded/floating status and elevation; initialize $p_i^0$ and $s_i^0$.
**end for**
Assemble candidate state $\widehat{\mathcal{W}}^0$.
**repeat**
    Check prompt consistency, scene bounds, grounding, and collisions.
    **if** no violation remains **then break**.
    Produce textual feedback and update only the conflicting poses $p_i^0$.
**until** the candidate state passes verification.
**return** $\mathcal{W}^0 \leftarrow \widehat{\mathcal{W}}^0$.

Figure 7: Pseudocode for State Construction. Dual-view evidence initializes an object-centric 3D state, and conflict resolution selectively repairs invalid placements while retaining accepted content.

# A Implementation Details

## A.1 State Construction Procedure

Fig. 7 details Conflict-Aware Dual-View Initialization. The front view provides object appearance, the BEV provides planar layout, and the generated asset provides 3D geometry and proportions. Verification only repositions conflicting objects; accepted assets and unaffected records remain fixed.

## A.2 State Evolution Task Instantiations

StateFlow supports different evolution tasks through the same structured update interface. Each task follows a common pattern:

$$\text{Intent} \rightarrow \text{Event} \rightarrow \text{Target} \rightarrow \text{State-field update} \rightarrow \mathcal{W}^{t+1}.$$

The task-specific instantiations are summarized below.

- **Rigid-object motion:** motion instruction $\rightarrow$ target object $\rightarrow$ 3D translation, rotation, or $(x, y, z)$ trajectory $\rightarrow$ update $p_i^t$.
- **Character action:** action instruction $\rightarrow$ target character $\rightarrow$ pose or motion trajectory $\rightarrow$ update $p_i^t$ while preserving identity.
- **Appearance or status change:** attribute instruction $\rightarrow$ target object $\rightarrow$ appearance, material, or status descriptor $\rightarrow$ update $s_i^t$.
- **Non-rigid motion:** deformation or topology-changing event $\rightarrow$ target object $\rightarrow$ generate a target-state reference and lift it to a new 3D asset $\rightarrow$ replace $g_i^t$ and update $s_i^t$ while preserving the object's identity and coarse placement.
- **Scene style change:** style instruction $\rightarrow$ affected scene or object scope $\rightarrow$ rewrite style and material descriptors $\rightarrow$ update the corresponding semantic states while preserving geometry and layout.
- **Scene expansion:** expansion instruction $\rightarrow$ new spatial region $\rightarrow$ invoke State Construction for the new region and objects $\rightarrow$ merge the new records into $\mathcal{W}^t$ while retaining the existing world state.

Table 6: Typical end-to-end runtime.

| Stage | Time | Scaling |
|---|---|---|
| Construction: image/reasoning | ∼3 min | per scene |
| Construction: 3D generation | ∼3 min | per object, concurrent |
| State Evolution | ∼5 min | per edit |
| State Access | ∼4 min | per edit |
| Seedance2.0 enhancement | ∼2 min | per 5-second clip |
| Typical total | ∼17 min | representative scene |

**Non-rigid transition through asset replacement.** Explicitly simulating complex non-rigid or topology-changing processes in 3D, such as deformation, fracture, or explosion, requires time-varying geometry, materials, and physical interactions—effectively a 4D state. Instead of solving this difficult physical simulation, we represent the process by two discrete object states. Given the source-state reference and an event instruction, an image generator first produces a target-state reference while preserving the object's semantic identity. The target reference is then lifted to a new 3D asset, yielding source and target geometry $(g_i^{\mathrm{src}}, g_i^{\mathrm{tgt}})$. We render the two states as boundary keyframes and use a keyframe-conditioned video generation model to synthesize the intermediate frames. Thus, the structured world records the editable before-and-after states, while the video model realizes the continuous visual transition between them.

*Example.* For a spaceship explosion, we condition the exploded appearance on an image of the intact spaceship, reconstruct the exploded version as a second 3D asset, and render the intact and exploded states as endpoint keyframes. The video generator completes the explosion between these endpoints.

These task forms differ in which state fields they modify, but they share the same event parsing, target localization, and selective application procedure.

### A.3 Runtime

Table 6 reports typical wall-clock time. Construction uses approximately three minutes for image generation and reasoning, followed by roughly three minutes per object for 3D generation; object assets are generated concurrently. State Evolution and State Access scale with the number of edits, whereas Seedance2.0 scales with output duration.

The total is approximately 17 minutes for a typical scene. Wall-clock time can vary with external API response speed, so the values should be interpreted as representative rather than deterministic.

### A.4 Video Enhancement and VBench Protocol

For the final enhanced video, Seedance2.0 (Seedance et al., 2026) receives three inputs: the generated front image, the Blender-rendered video, and the text prompt. The instruction is "keep geometry from the reference video and appearance from the reference image." Thus, StateFlow supplies explicit rendered geometry as the conditioning signal, while Seedance2.0 enhances appearance and temporal fidelity. The supplementary video provides examples of this workflow.

Table 4 separates the Blender direct render, the full StateFlow output, and the standalone Seedance2.0 baseline. The full method and standalone baseline use the same video backbone, so their difference isolates the contribution of StateFlow's structured world and rendered geometry.

StateFlow obtains the highest average and improves Subject Consistency from $0.8110$ to $0.9135$ over the same-backbone Seedance2.0 baseline. Direct rendering already provides strong geometry and object consistency, while the video backbone improves the visual appearance of the final output.

For the complete VBench evaluation, we use a custom set of 30 scene prompts and five methods, generating one video per prompt per method. Every method receives the same scene-level prompt. We evaluate complete videos at their native resolution using VBench v0.1.5 (Huang et al., 2024). Because the raw Imaging Quality score uses a 0–100 scale, we divide it by 100 before reporting and aggregating it with the other normalized dimensions.

# B Additional Quantitative Evaluation

## B.1 State Construction: Layout and Physical Validity

We compare StateFlow with the object- and scene-generation baselines SAM3D (Team et al., 2025), PartCrafter (Lin et al., 2025), and SynCity (Engstler et al., 2025). We match objects by semantic identity to VLM-extracted BEV boxes used as pseudo ground truth. *Layout IoU* is the mean top-down footprint IoU over matched objects. *Floating rate* is the percentage of grounded objects whose 3D box does not contact its assigned support surface, and *collision rate* is the percentage of non-supporting object pairs with intersecting 3D boxes. CLIP-T measures image–text alignment between the rendered scene and the input prompt.

StateFlow achieves the highest layout IoU and CLIP-T while attaining zero floating and collision errors, showing that its advantage extends from prompt alignment to usable 3D structure.

# C Ablation Studies

This section contains the supplementary ablation studies. We first isolate the persistent state table used by State Evolution and then summarize the role of each major StateFlow component.

## C.1 State Evolution: Persistent State Table

To isolate the role of the persistent state table, we compare StateFlow with a full-regeneration variant. This variant edits a fresh reference image for every instruction and reconstructs the entire scene, instead of locating the affected state record and applying a minimal update. Removing the state table reduces VBench Subject Consistency by $0.28$ and Background Consistency by $0.17$.

The result supports the central motivation of State Evolution: a requested edit should change only the necessary information. Full regeneration can satisfy the new instruction, but it also resamples previously accepted content, causing identity and background drift.

## C.2 Component Ablations and Module Roles

The following leave-one-out correspondences clarify the role of each major component:

- **w/o world state** $\rightarrow$ one-shot generation with a video model such as Wan (Wan et al., 2025) or Seedance2.0 (Seedance et al., 2026): no persistent structure is available for later editing or camera-grounded access.
- **w/o State Construction** $\rightarrow$ raw object or scene generation such as SAM3D (Team et al., 2025): layout and grounding errors arise before evolution and reach the final video.
- **w/o State Evolution** $\rightarrow$ a static world: the scene can be rendered but cannot be selectively edited under new instructions.
- **w/o State Access** $\rightarrow$ a VLM-only camera planner: trajectories are not verified through rendering and therefore suffer from visibility, framing, and collision errors.

# D Additional Qualitative Examples

We provide representative examples of scene revisitation, physically controlled motion, coordinated multi-object interaction, and a challenging industrial scene. Row identities, temporal order, and frame-selection details are specified in the captions.

## D.1 CS-style Game Demonstration

The desert environment in the CS-style demonstration is rendered in Blender from the scene generated by StateFlow. The first-person gun and hand are a composited video overlay used to demonstrate a game-like presentation; they are not part of the generated environment. A rigged first-person asset could be integrated into the same world, but game mechanics and user-interface layers are

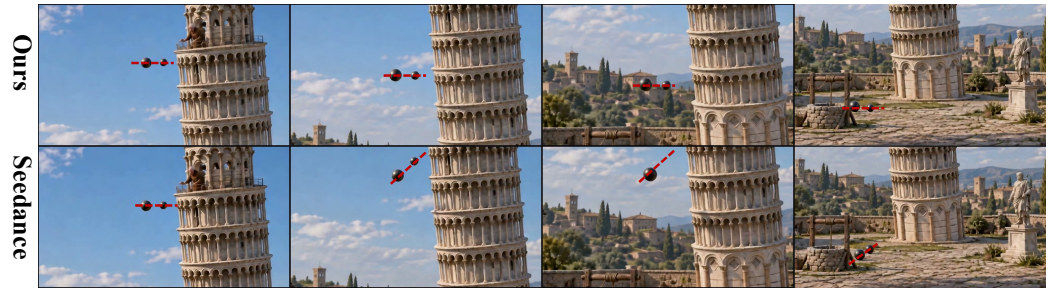


Figure 8: Physical motion in the Leaning Tower of Pisa experiment. The top row shows StateFlow and the bottom row shows Seedance2.0; columns proceed from earlier to later timestamps, with the final column showing the landing moment. The red dashed lines mark the center-of-mass positions of the balls. StateFlow follows a trajectory prescribed by physical equations, whereas Seedance2.0, as a direct video-generation baseline, does not follow the prescribed physical rule.




Figure 9: Coordinated multi-object interaction. The top row shows the scene-level render and the bottom row shows the final video. Columns show matched normalized times from early to late in the sequence. The keyframes illustrate how the final video follows the coordinated motion specified in the scene state while the camera moves through the amusement-park scene.

outside our scope. As in conventional game production, our focus is scene generation and persistent world structure rather than gameplay logic.

# E Additional Evaluation and Analysis

## E.1 State Access: Camera Planning

We compare our render-feedback camera planner with ChatCam (Liu et al., 2024). Intent match is the fraction of video seconds that satisfy the requested camera intent, judged by Gemini. Target visibility is the fraction of frames in which the target is visible, tracked with SAM2. Collision rate is computed geometrically as the number of colliding camera–object pairs divided by the number of scene objects, without a learned model. Smoothness is the second derivative of the trajectory; lower acceleration indicates smoother camera motion.

StateFlow improves every metric. ChatCam relies primarily on language-model reasoning, whereas StateFlow closes the planner–render loop: each proposed trajectory is executed, inspected in the actual world, and revised from textual render feedback.

## E.2 Cross-Model MLLM Evaluation

We conduct a second MLLM-based evaluation with GPT-5.4 using the same criteria as the Gemini evaluation. For scene generation, we compare StateFlow with the best baseline for each dimension. For video, we compare against Seedance2.0.




Figure 10: Video-generation comparison on an industrial alien mining base. StateFlow follows an intricate camera trajectory, preserving scene and object identities. AniMaker and MovieAgent exhibit scene or spatiotemporal inconsistency; WAN and Seedance2 show limited motion.

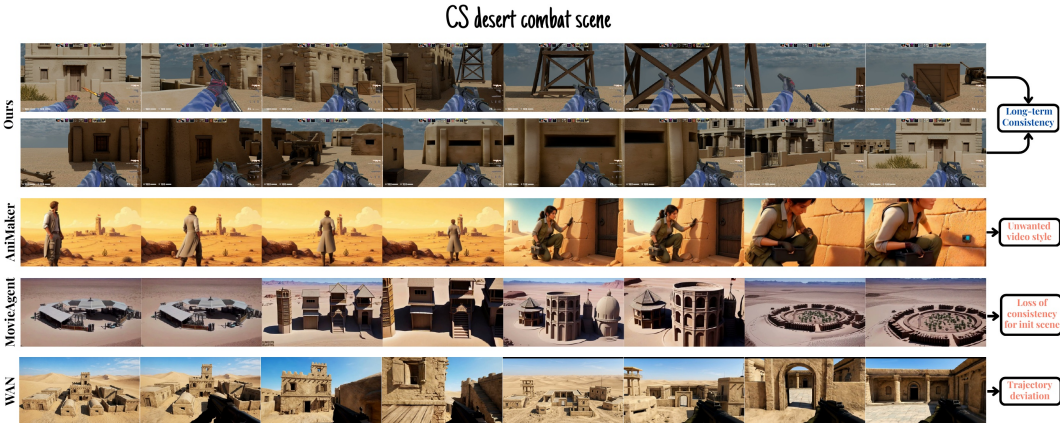


Figure 11: CS-style game prototyping in a generated desert environment. The StateFlow sequence preserves a persistent scene across long-range viewpoint changes, while the compared video-generation methods exhibit unwanted transitions, inconsistent geometry, or trajectory deviation. Our first-person gun and hand are composited overlays illustrating a game-like presentation.

GPT-5.4 ranks StateFlow highest across all scene dimensions and on the overall video score. It also rates StateFlow above Seedance2.0 on every video dimension except camera quality. Agreement between the MLLM evaluations provides cross-model evidence for the reported perceptual trends.

## F Full Perceptual Evaluation

**User-study protocol.** We recruit 50 participants and evaluate 12 prompts in each of two settings. The scene-level setting presents $45^\circ$ BEV fly-around videos, while the video-level setting presents the final rendered videos. Outputs from all methods are anonymized and randomized. Participants rate each method on a five-point Likert scale for the dimensions listed in Table 8, and we report the mean score for each dimension.

Table 7: GPT-5.4 evaluation. The scene baseline is the best competing method for each dimension; the video baseline is Seedance2.0.

| *Scene generation* **Dimension** | **StateFlow** | **Best baseline** |
|---|---|---|
| Prompt alignment | **3.00** | 2.53 |
| Layout plausibility | **3.68** | 3.45 |
| Completeness | **2.88** | 2.20 |
| Multi-view coherence | **4.25** | 4.03 |
| Previsualization usefulness | **3.38** | 2.98 |
| Geometric quality | **3.65** | 3.38 |
| Overall | **3.47** | 3.10 |
| *Video generation* **Dimension** | **StateFlow** | **Seedance2.0** |
| Prompt/story alignment | **3.63** | 3.60 |
| Spatial consistency | **3.93** | 3.73 |
| Identity consistency | **4.03** | 3.53 |
| Motion plausibility | **3.80** | 3.70 |
| Camera quality | 3.80 | **4.03** |
| Previsualization usefulness | **3.73** | 3.70 |
| Overall | **3.82** | 3.72 |

Table 8: User study and MLLM-based evaluation on 3D scene generation (top) and video generation (bottom). The rows *w/o BEV Layout*, *w/o Conflict Res.*, and *VLM-only Camera* are ablation variants of StateFlow. Each cell is reported as *user study / MLLM evaluation*. User study: $N{=}50$ participants, 12 prompts per setting. **Bold** represents the best performance.

*Scene-level study ($45^\circ$ BEV fly-around videos)*

| **Method** | **Prompt Align.** ↑ | **Layout Plaus.** ↑ | **Complete ness** ↑ | **Geo. Quality** ↑ | **Coher. ence** ↑ | **Previs Useful.** ↑ | **Overall Score** ↑ |
|---|---|---|---|---|---|---|---|
| SynCity | 3.5/2.8 | 2.8/2.8 | 3.1/2.6 | 3.9/2.8 | 2.9/4.4 | 2.7/2.6 | 3.2/3.0 |
| SAM3D | 3.9/2.4 | 3.1/2.6 | 3.7/2.2 | 3.9/2.2 | 3.2/4.6 | 3.1/1.2 | 3.5/2.5 |
| PartCrafter | 2.4/2.0 | 1.9/2.0 | 1.9/2.0 | 2.2/2.0 | 2.0/3.2 | 1.8/1.0 | 2.0/2.0 |
| w/o BEV Layout (Ours) | 3.6/3.0 | 3.0/2.6 | 3.4/2.0 | 3.7/2.4 | 3.0/3.6 | 3.1/1.2 | 3.3/2.5 |
| w/o Conflict Res. (Ours) | 3.7/3.0 | 3.2/2.6 | 3.2/2.0 | 3.8/2.6 | 3.1/3.6 | 3.2/1.6 | 3.4/2.6 |
| **StateFlow (Ours)** | **4.4/3.6** | **4.6/3.3** | **4.2/3.7** | **4.2/3.6** | **4.6/4.7** | **4.7/3.1** | **4.5/3.7** |

*Video-level study (final rendered videos)*

| **Method** | **Prompt/Story Align.** ↑ | **Spatial Cons.** ↑ | **Identity Cons.** ↑ | **Motion Plaus.** ↑ | **Camera Quality** ↑ | **Previs Useful.** ↑ | **Overall Score** ↑ |
|---|---|---|---|---|---|---|---|
| Seedance2 | 4.1/3.4 | 3.3/1.5 | 3.2/1.3 | 3.9/2.6 | 3.5/1.4 | 3.7/2.3 | 3.6/2.1 |
| Wan2.2 | 3.5/3.6 | 2.8/2.4 | 2.7/2.2 | 3.4/2.7 | 3.0/1.6 | 3.1/1.5 | 3.1/2.3 |
| AniMaker | 2.8/1.6 | 2.3/1.2 | 2.2/1.5 | 2.8/2.4 | 2.5/2.3 | 2.4/1.4 | 2.5/1.7 |
| MovieAgent | 1.9/1.3 | 1.6/1.1 | 1.5/1.2 | 1.9/2.2 | 1.6/1.4 | 1.5/1.3 | 1.7/1.4 |
| VLM-only Camera (Ours) | 4.3/3.5 | 4.6/3.4 | 4.6/3.7 | 4.1/3.2 | 3.4/2.6 | 3.8/3.5 | 4.1/3.3 |
| StateFlow (Ours) | **4.4/3.8** | **4.7/4.3** | **4.7/3.9** | **4.2/3.6** | **4.6/4.4** | **4.6/4.2** | **4.5/4.0** |