Look-Before-Move: Narrative-Grounded World Visual Attention in Dynamic 3D Story Worlds

Jiaming Bian
Central South University
bianjiaming@csu.edu.cn

Bingliang Li
University of New South Wales
bing.liang.li@outlook.com

Yuehao Wu
University of New South Wales
yuehao.wu@unsw.edu.au

Pichao Wang*
NVIDIA
pichaowang@gmail.com

Zhi Wang
Nanjing University
zhiwang@nju.edu.cn

Hailan Ma
Nanyang Technological University
hailanma0413@gmail.com

Huadong Mo†
University of New South Wales
huadong.mo@unsw.edu.au

Zhenhong Sun†
Australian National University
zhenhong.sun@anu.edu.au

Abstract

As embodied AI and world models increasingly operate in dynamic 3D environments, visual perception must move beyond passively interpreting given observations toward actively deciding what to observe. We study this problem through camera planning in dynamic 3D story worlds, where the camera must not only generate smooth motion, but also decide what visual evidence should be acquired before it moves. We formulate this capability as **Narrative-Grounded World Visual Attention**, where the camera acts as an embodied observer that determines what to observe, how to compose the observation, and how to shift attention over time under narrative intent and physical 3D constraints. To realize this capability, we propose **Look-Before-Move**, a camera planning framework that separates observation specification from motion execution. It first builds a Semantic Observation Contract to convert directorial intent into executable visual constraints, then performs Monte Carlo Viewpoint Search to find narrative-compliant and geometrically feasible viewpoints, and finally applies Semantic Trajectory Grounding to connect selected viewpoints into continuous, collision-aware, and temporally coherent camera motion. We further construct a dynamic 3D Story World Benchmark based on *StoryBlender*, covering 50 stories, 457 scenes, and 1585 shots with animated characters, semantic scene configurations, and executable 3D environments. Experiments show that our framework improves subject perception, intent consistency, and trajectory quality over representative baselines, demonstrating the importance of organizing visual attention before generating camera motion. Project page and code are available at: https://engineeringai-lab.github.io/Look-Before-Move/.

1 Introduction

As embodied AI [11, 47, 32] and world models [25, 20] increasingly operate inside executable 3D environments, visual perception can no longer be treated as a passive input stream. Unlike conventional image or video understanding, where visual evidence is already given, an agent in a native 3D world must actively determine what to observe before planning how to move. This connects

*This work is not related to the author's position at NVIDIA. † Co-corresponding authors

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

to autonomous camera control [22] and camera trajectory planning [6], but shifts the focus from smooth trajectory generation to observation-driven camera planning. This question is especially important in dynamic 3D story worlds, where recent systems have begun to construct editable narrative environments with controllable layouts, assets, character actions, and spatial-temporal dynamics [27, 12, 17]. Here, a 3D story world is an executable, editable narrative environment with geometry, identified characters and objects, spatial layouts, scripted actions, and time-varying states in shared 3D coordinates. A camera in such worlds is not simply a geometric sensor that maximizes visibility or avoids collisions. It acts as the eye of the world: selecting narrative subjects, emphasizing actions, revealing spatial relations, and shifting attention as the story unfolds. For example, when a story describes an assassin pointing a pistol, the camera should not merely capture any person; it should identify the acting subject, preserve the action cue, and compose the scene according to its narrative meaning. We define this capability as **Narrative-Grounded World Visual Attention**: the camera organizes what to observe, how to observe it, and when to transition attention within a physically constrained 3D world.

Realizing this capability requires solving three coupled problems. First, high-level directorial intent must be converted into visual requirements, such as target subjects, semantic relations, visibility conditions, composition preferences, and action cues. Second, the system must search for viewpoints that satisfy both narrative relevance and physical feasibility, since a semantically appropriate view may be invalid due to occlusion, collision, poor scale, or infeasible camera placement. Third, selected viewpoints must be grounded into camera motion that tracks dynamic actors, avoids obstacles, and preserves temporal coherence. These requirements go beyond camera-controllable video synthesis [34, 9, 46] and text-to-camera trajectory generation [49, 18, 5], where the objective is to follow a camera motion, synthesize a trajectory, or maintain visual consistency. Together, they advance camera planning toward the semantic-spatial-temporal organization of visual evidence.

Existing camera planning methods provide foundations, but they often assume that the observation objective is already specified. Rule-based and geometry-driven methods encode cinematographic heuristics, visibility constraints, collision avoidance, and smooth path generation [6, 22]. Recent learning-based [2, 40] and generative methods [34, 9, 46, 41] model camera motion priors or synthesize camera-controllable videos by injecting trajectory representations into video generation models. Several works analyze camera motion and 3D camera control in videos [51, 1, 21], while recent 3D cinematography systems explore auto-regressive trajectory generation [49], diffusion-based camera and scene generation [18], text-to-camera trajectory generation [5], and multi-agent film automation [43]. These methods improve feasible trajectory generation, camera-controllable synthesis, or autonomous cinematography, but they do not address how an intent should be inferred from narrative intent and grounded in a physically executable 3D world. To address this gap, we propose **Look-Before-Move**, a narrative-grounded camera planning framework built on a simple principle: before a camera can move intelligently, it must first know what visual evidence it is supposed to acquire. Instead of directly generating a trajectory from a story instruction, our framework first constructs a Semantic Observation Contract that converts directorial intent into executable visual constraints. It then performs Monte Carlo Viewpoint Search to identify viewpoints that are both narrative-compliant and geometrically feasible, followed by Semantic Trajectory Grounding to connect selected viewpoints into continuous camera motion. This decomposition turns narrative-grounded world visual attention into three executable steps: specifying the intended observation, selecting feasible viewpoints, and grounding them into motion.

A systematic study of this problem also requires an executable evaluation environment. Although recent 3D story-world generation methods provide editable narrative environments, real-world filming is costly and difficult to control. Additionally, 2D video datasets or diffusion-generated videos cannot reliably determine whether a camera plan is physically feasible, whether a better viewpoint exists, or whether a failure results from semantic misunderstanding, geometric occlusion, or motion instability. We therefore construct a dynamic 3D Story World Benchmark based on *StoryBlender* [17]. The benchmark covers 50 stories, 457 scenes, and 1585 shots, and supports controlled evaluation along three dimensions: subject perception, intent consistency, and trajectory quality. Our experiments show that Look-Before-Move improves over representative camera trajectory planning baselines across all three evaluation dimensions. The results support our central claim: in dynamic 3D story worlds, effective camera planning should begin with observation before motion generation.

In summary, our main contributions are as follows:

- We formulate **Narrative-Grounded World Visual Attention** as a camera planning capability in dynamic 3D story worlds, reframing the problem from passively interpreting given observations to intent-driven visual evidence acquisition.
- We propose **Look-Before-Move**, a narrative-grounded camera planning framework that decomposes camera planning into observation specification, viewpoint search, and trajectory grounding.
- We construct a dynamic 3D Story World Benchmark on executable 3D scenes, supporting controlled evaluation of subject perception, intent consistency, and trajectory quality.

2 Related Work

Camera Trajectory Planning. Autonomous camera control has long studied generating feasible and visually pleasing viewpoints or trajectories in virtual environments. Early methods rely on cinematography rules, geometric constraints, visibility reasoning, collision avoidance, and smooth paths [6, 22]. Recent generative methods extend camera control to video generation by injecting trajectory representations, such as extrinsic matrices or Plucker coordinates, into diffusion models [34, 9, 46], or by improving camera-controllable synthesis through multi-view and 3D-aware consistency [41, 40, 2]. Related generation work covers entity guidance [30], diversity [23], video alignment [33], controllable dance generation [48], and video-to-audio generation [16]. However, many approaches [51, 1, 21] operate primarily in pixel space and lack access to executable 3D embodied worlds. Recent 3D cinematography methods explore auto-regressive trajectory generation [49], diffusion-based camera and scene generation [18], and multi-agent film automation [43]. Different from these works, our goal is not only to synthesize a trajectory, but to ground the intended observation from narrative intent before camera motion.

3D Story Worlds. The construction of 3D story worlds provides the foundation for controllable narrative visualization. Related perception methods address human-object interaction detection [44, 45] and 3D human pose estimation [31]. Engine-based methods show that virtual environments can support dynamic storyboard generation and pre-visualization [27]. Recent LLM-based systems use code generation or structured reasoning to synthesize open-domain 3D scenes with editable layouts and assets [11, 47, 25, 20]. Sketch-guided generation [42] also supports interactive 3D scenes. Other works introduce character action scripting and narrative behavior modeling, enabling 3D environments to express story events instead of static layouts [12]. *StoryBlender* further maintains cross-scene asset consistency and produces editable dynamic 3D storyboards with spatial-temporal dynamics [17]. These methods make it possible to construct narrative 3D worlds, but remain open how such worlds should be observed. Our work complements 3D story-world by introducing a camera planning framework that organizes visual attention inside executable story worlds.

LLM Reflection and Agentic Planning. Large language models have been used as agents that reason, reflect, and refine outputs through feedback. Frameworks such as Reflexion and Self-Refine show that language models can evaluate and revise their own generations [29, 26]. Works extend reflection with dynamic instructions, visual-language reasoning, experience accumulation, tree search, and multi-agent collaboration [24, 4, 50, 52, 8]. These approaches are effective in text-centric or interactive environments, including narrative settings [36]. Related work explores instruction-guided medical VQA [38], feature-space knowledge enhancement [39], hierarchical decision-making [10], and multi-round medical reasoning [37]. In dynamic 3D worlds, however, reflection must be grounded in physical constraints: a proposed view may be occluded, the camera may collide with the scene, the subject may leave the frame, or the trajectory may become unstable. Embodied reasoning and visual reflection studies suggest that planning feedback should be tied to visual and geometric evidence [19, 13, 35]. In our framework, reflection is used as physical verification for viewpoint selection and trajectory grounding rather than as a verbal memory mechanism.

3 Method

Problem Statement. Let $\mathcal{W} = (\mathcal{G}, \mathcal{A}, \mathcal{L}, \mathcal{M})$ denote an executable dynamic 3D story world, where $\mathcal{G}$ is the scene geometry, $\mathcal{A}$ contains characters, objects, and semantic attributes, $\mathcal{L}$ describes the spatial layout, and $\mathcal{M}$ records animations and object motions. Given a directorial intent $\mathcal{I}$ and a temporally ordered event sequence $\mathcal{S} = (s_1, \ldots, s_N)$, scene-to-camera trajectory planning aims to produce a continuous camera trajectory for each event:

$$\tau_i = \pi(\mathcal{W}, \mathcal{I}, \mathcal{S}_{\leq i}), \tag{1}$$

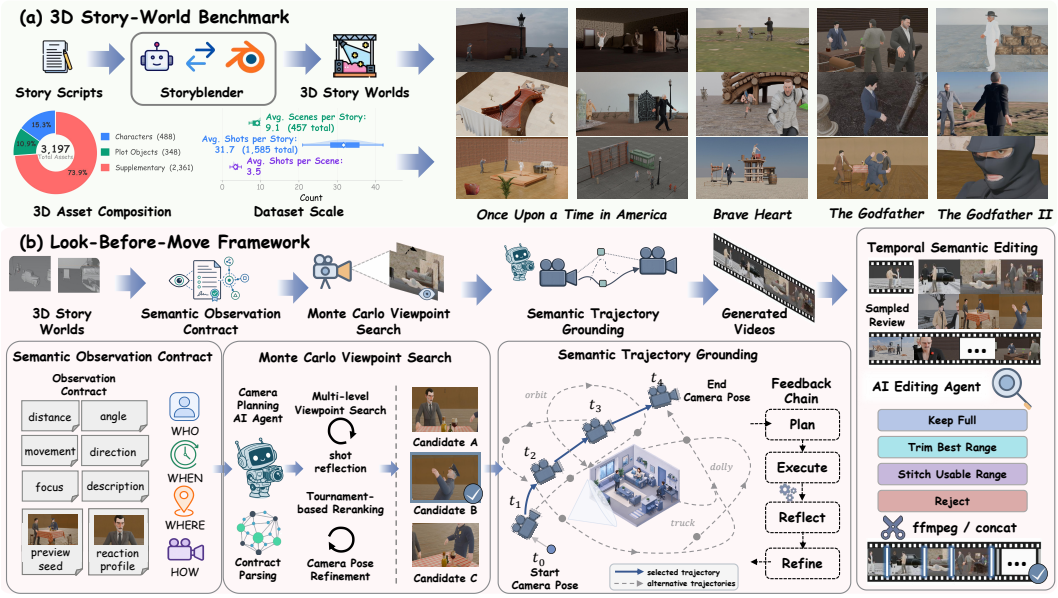


Figure 1: Overview of the benchmark construction and Look-Before-Move framework. (a) We build a dynamic 3D story-world benchmark using *StoryBlender*, covering narrative scripts, controllable scenes, character animations, camera annotations, and dataset statistics. (b) Look-Before-Move converts narrative intent into an observation contract, searches and refines feasible viewpoints in the executable 3D world, and grounds selected views into temporally coherent camera trajectories.

where $\tau_i : [0,1] \to \mathrm{SE}(3)$ is an executable camera motion and $\pi(\cdot)$ is the planning policy. Here, $\mathrm{SE}(3)$ represents the mathematical space of 3D positions and orientations. Executing τ_i in the 3D world produces an observation clip:

$$c_i = \mathcal{R}(\mathcal{W}, \tau_i, s_i), \tag{2}$$

and the final output is an ordered observation sequence $\mathcal{E} = (c_1, \ldots, c_N)$.

Unlike static viewpoint prediction or low-level trajectory generation, this task aims to organize *world visual attention* by deciding what should be observed, emphasized, and revealed over time. It requires solving two coupled problems: *spatial viewpoint composition*, which selects intent-compliant and visible viewpoints under geometric constraints, and *temporal motion coordination*, which aligns these viewpoints with feasible camera motion relative to dynamic subjects and narratives.

To address this task, we propose **Look-Before-Move**, a collaborative multi-agent camera planning framework consisting of three components. First, **3D Story World Benchmark** provides executable dynamic scenes with geometry, semantic entities, animations, shot-level configurations, and rendering feedback for controlled camera planning evaluation. Second, **Look** addresses *spatial viewpoint composition* by establishing an Observation Contract to search for safe, visible, and intent compliant viewpoints before any camera motion is committed. Third, **Move** addresses *temporal motion coordination* through grounding selected viewpoints into continuous camera trajectories that follow dynamic subjects, avoid collisions, and seamlessly align with narrative progression.

3.1 3D Story World Benchmark

To study narrative-grounded camera planning, we require a benchmark where story intent, dynamic events, character identities, and physical 3D geometry are jointly controllable and verifiable.

3D World Building. Existing 2D video datasets or diffusion-generated scenes often suffer from spatial hallucinations, identity drift, and a lack of physical grounding, severely limiting their utility for rigorous camera planning. To overcome these limitations, we construct our executable benchmark using *StoryBlender* [17], an engine-verified 3D generation framework that guaranties strict *inter-shot consistency* and *explicit editability*. Unlike 2D pixel-space representations, our virtual world relies on a persistent, physically grounded simulation. Formally, this world is represented as $\mathcal{W} =$

$(\mathcal{G}, \mathcal{A}, \mathcal{L}, \mathcal{M})$, corresponding to scene geometry, canonical character assets, spatial layout, and dynamic action sequences. By materializing entities in a unified 3D coordinate space, the environment provides the camera with precise geometric boundaries and semantic perception capabilities. Building upon this foundation, we integrate these engine-verified scenes, animations, and temporal events to formalize our dataset as $\mathcal{D} = \{(\mathcal{W}_j, \mathcal{I}_j, \mathcal{S}_j, \mathcal{Y}_j^{\text{scene}})\}_{j=1}^{M}$, where $\mathcal{I}_j$ denotes the directorial intent, $\mathcal{S}_j$ represents the event progression, and $\mathcal{Y}_j^{\text{scene}}$ stores the comprehensive metadata necessary for programmatic evaluation. This establishes a reliable mapping between narrative intent and physical rendering, ensuring that camera decisions receive accurate visual and geometric feedback.

Benchmark. Addressing the lack of dynamic evaluation, we propose an Executable 3D World Benchmark for world visual attention. Spanning 50 stories, the dataset comprises 457 dynamic scenes and 1585 shots. To support narrative environments, it includes 3197 3D assets (836 plot-related assets and 2361 supplementary assets for scene decoration), featuring an average of 9.76 characters per story and 1.44 character actions per shot. In the evaluation, given the 3D world $\mathcal{W}$, intent $\mathcal{I}$, and event process $\mathcal{S} = (s_1, \ldots, s_N)$, the model must output a continuous camera trajectory $\tau_i : [0, 1] \to \mathrm{SE}(3)$ for each unit s_i. Because our dataset is built on 3D environments rather than static pose predictions, it allows models to make spatial-temporal decisions and undergo closed-loop engine verification. We conduct quantitative evaluations across three dimensions: (1) **Subject Perception**, assessing the ability to capture targets and avoid physical occlusions; (2) **Intent Consistency**, verifying whether viewpoints adhere to cinematic semantics; and (3) **Trajectory Quality**, examining motion smoothness, spatial-temporal dynamics, and execution feasibility.

3.2 Look for Spatial Viewpoint Composition

Before the camera can move, it must first determine what the story requires it to see. The *Look* stage addresses this problem by converting high-level directorial intent into an adaptive observation contract that specifies the subjects, actions, spatial relations, visibility requirements, and composition preferences to be satisfied by candidate viewpoints. To this end, a group of specialized perception and planning agents collaborate over narrative instructions and rendered scene evidence, establishing a semantic-geometric perception interface between the story world and the camera.

Semantic Observation Contract. In complex 3D dynamic environments, the system first needs an accurate understanding of the current physical and visual state. To achieve this, we introduce the Perception Agent. By interacting with the 3D world, this agent samples and renders multi-view preview images around the target character to obtain intuitive environmental feedback. These multi-view previews not only capture the character's pose and spatial position but also reveal potential environmental occlusions, thus extracting scene state representations rich in visual and geometric information from the observation unit s_i. This intelligent perception mechanism based on real rendering establishes a solid foundation for subsequent semantic intent parsing and planning.

Faced with abstract high-level directorial intent, the camera struggles to execute it directly, necessitating a precise perception-based mapping mechanism. To this end, the Planning Agent, upon receiving the perception results, proposes an Observation Contract. Through semantic parsing Φ, this agent converts the abstract intent $\mathcal{I}$ and the scene state of the current observation unit s_i into a concrete observation contract $\mathcal{O}_i$:

$$\mathcal{O}_i = \{a^*, C_{\text{spatial}}, T_{\text{action}}\} = \Phi(\mathcal{I}, s_i) \tag{3}$$

where a^* denotes the observable target subject, C_{spatial} represents spatial composition requirements, and T_{action} is the temporal action constraint. This structured contract not only specifies "what to look at" and "how to look at it" for the camera but also effectively narrows the state space for subsequent viewpoint search, successfully establishing a mapping between semantic intent and 3D geometry to provide prior guidance for precise camera control.

Monte Carlo Viewpoint Search. Merely establishing observation targets is insufficient to guaranty the usability of the framing; the camera must also identify the optimal visual landing point in the executable 3D world $\mathcal{W}$. To achieve this, we design the Camera Reflection mechanism, a self-evaluation process based on multi-agent collaboration, viewpoint search, and fine-tuning. First, based on the observation contract $\mathcal{O}_i$, the system drives multiple Camera Agents to perform Monte Carlo-based Multi-level Viewpoint Search, generating a large number of candidate shot boards in the large space. Subsequently, the Evaluation Agent executes Tournament-based Reranking on these candidate viewpoint sets $\mathcal{V}$, progressively eliminating low-quality shots through competitive scoring

to select the winning initial optimal viewpoint v^*_{init}:

$$v^*_{\text{init}} = \arg\max_{v \in \mathcal{V}} \left(\lambda_1 \text{Vis}(v, a^*) + \lambda_2 \text{Comp}(v, C_{\text{spatial}}) - \lambda_3 \text{Occ}(v, \mathcal{W})\right) \tag{4}$$

where $\text{Vis}(\cdot)$ evaluates subject visibility, $\text{Comp}(\cdot)$ measures composition rationality, and $\text{Occ}(\cdot)$ calculates physical occlusion caused by the environment.

Following this competitive selection, the Reflection Agent refines the winning viewpoint through localized camera-pose adjustment. Starting from v^*_{init}, it applies small perturbations to the camera position, orientation, and focal parameters, and re-renders the scene to check whether the target subject remains visible, the action cue is preserved, and the composition contract is satisfied. Refinements that introduce collision, severe occlusion, excessive subject truncation, or unstable framing are rejected. Through this render-evaluate-adjust loop, the system corrects minor visual imperfections, such as off-center subjects, poor scale, partial occlusion, or weak spatial relation expression, and obtains the final viewpoint v^*_i.

In summary, the *Look* stage converts narrative intent into a physically executable spatial observation decision. It first builds a semantic observation contract from narrative intent and rendered scene evidence, then searches and refines candidate viewpoints through Monte Carlo sampling, tournament reranking, and render-based reflection. The resulting viewpoint v^*_i grounds "what should be observed" in the executable 3D world and serves as the visual anchor for subsequent trajectory grounding in the *Move* stage.

3.3 Move for Temporal Motion Coordination

Once the observation contract and initial optimal viewpoint are determined, the system transitions from the "searching" for shots to the "executing" motion stage of camera trajectory planning. At this stage, the planning and editing agents work collaboratively to generate continuous, stable, and executable motion trajectories in the 3D scene, a process we term **Semantic Trajectory Grounding**. The whole process of Semantic Trajectory Grounding can be divided into Trajectory Planning, Trajectory Reflection, and Temporal Semantic Editing.

Trajectory Planning. After acquiring a sequence of high-quality discrete shots (keyframes) $\{v^*_t\}$, generating smooth, continuous trajectories that adapt to dynamic subject movements is the core of achieving high-quality visual presentation. The Planning Agent uses the discrete shots as key nodes to solve for the optimal continuous trajectory τ^*_i in the joint spatial-temporal space:

$$\tau^*_i = \arg\min_{\tau} \sum_t \left(\|\tau(t) - v^*_t\|^2 + \gamma \text{Cost}_{\text{smooth}}(\dot{\tau}(t), \ddot{\tau}(t))\right) \tag{5}$$

This objective function aims to minimize the deviation between the continuous trajectory $\tau(t)$ and the discrete key shots v^*_t, while strictly constraining the camera's kinematic properties (such as velocity $\dot{\tau}(t)$ and acceleration $\ddot{\tau}(t)$) through the penalty term $\text{Cost}_{\text{smooth}}$. Through this relay strategy of first generating discrete shots by the Camera Agent, then generating continuous trajectories by the Planning Agent, the system can plan camera trajectories that both stably track the target and exhibit excellent smoothness.

Trajectory Reflection. Although the initially planned trajectories are theoretically feasible, they may still encounter unexpected physical obstacles or intent deviations in complex dynamic scenes, necessitating strict closed-loop verification. To address this, the Evaluation Agent intervenes again to trigger the Trajectory Reflection mechanism. It simulates execution and comprehensively scores $R(\tau_i)$ the entire generated trajectory τ_i in a real 3D rendering environment:

$$R(\tau_i) = \alpha P_{\text{subj}}(\tau_i) + \beta I_{\text{intent}}(\tau_i, \mathcal{I}) + \delta Q_{\text{traj}}(\tau_i) \tag{6}$$

The formula comprehensively evaluates the trajectory from three dimensions: subject perception P_{subj}, intent consistency I_{intent}, and motion quality Q_{traj}. Only trajectory segments with scores exceeding a threshold are retained. This agent-driven closed-loop verification mechanism actively identifies and corrects sub-optimal or unexecutable actions, greatly enhancing the robustness of camera movements and ensuring the absolute reliability of trajectories in complex environments.

Temporal Semantic Editing. When dealing with complex scenes containing multiple event streams, a single viewpoint is often insufficient to fully convey all information, making reasonable shot scheduling particularly important. At this stage, the Editing Agent takes over and schedules all

observation clip sets $\{c_i\}$ in the temporal dimension. It screens and splices them to generate the final observation sequence $\mathcal{E}^*$:

$$\mathcal{E}^* = \arg\max_{\mathcal{E}\subset\{c_i\}} \left(\sum_{c_i\in\mathcal{E}} U(c_i) - \eta \sum_{j=1}^{|\mathcal{E}|-1} D_{\text{trans}}(c_j, c_{j+1}) \right) \tag{7}$$

where $U(c_i)$ measures the information transmission utility and semantic concentration of a single clip, and $D_{\text{trans}}(\cdot)$ calculates the visual incoherence penalty for adjacent viewpoint switches. By balancing clip quality and switching smoothness, the Editing Agent condenses discrete observation clips into a high-quality visual sequence, thereby conveying high-level directorial intent as accurately and vividly as possible.

In summary, the *Move* stage transforms selected viewpoints into temporally coordinated camera behavior. It first connects discrete observation anchors into smooth continuous trajectories, then verifies their physical feasibility and narrative consistency through closed-loop trajectory reflection. Finally, temporal semantic editing schedules verified clips into a coherent observation sequence. As a result, the system converts "how to observe" into executable camera motion that remains stable, feasible, and aligned with the evolving story.

4 Experiment

4.1 Implementation Details

Benchmark Metrics. We evaluate camera planning from three complementary dimensions: **Subject Perception (SP)**, **Intent Consistency (IC)**, and **Trajectory Quality (TQ)**. SP focuses on whether the generated camera behavior can correctly capture the intended subjects under dynamic scene conditions. It measures target visibility, framing completeness, and robustness to physical occlusions. IC evaluates whether the resulting views are semantically aligned with the given filming intent, including shot scale, narrative focus, action relevance, and consistency with the event description. TQ assesses the execution quality of the generated camera motion, including trajectory smoothness, motion stability, transition coherence, and temporal continuity. All metrics are normalized to a 0–100 scale, where higher scores indicate better performance. Detailed metric definitions and computation protocols are provided in the **Appendix C**.

Setup and Evaluator Models. Our rendering and physics validation environment is built on Blender, which provides executable 3D scenes, camera control, collision checking, and frame-level rendering for evaluation. We use Gemini-3-Flash-Preview [7] for agent reasoning, reflection, and instruction generation. For subject perception evaluation, we use YOLO11x [15] together with ORB [28] to assess target detection, visibility, and frame-level consistency. For intent consistency, we use Qwen2.5-VL-7B-Instruct [3] to perform blind text-video alignment scoring between the rendered camera video and the corresponding filming instruction. To ensure a fair comparison, all methods are evaluated using the same story segments, prompts, renderer, evaluator models, and segment-count weighting protocol. All videos are rendered at 24 FPS on a workstation equipped with one NVIDIA RTX 5090 GPU. Details are provided in the **Appendix C** .

Datasets. We evaluate all methods on the cinematic test set introduced in this work, which covers indoor and outdoor scenes, multi-character interactions, object-centric events, and temporally evolving story segments. Each scene is paired with natural language filming instructions specifying the intended subject, narrative focus, action cue, and shot-level visual requirements, enabling zero-shot evaluation of text-to-3D camera scheduling. This setting tests whether a method can translate high-level directorial intent into executable camera behavior that decides both what to observe and how to organize motion over time; details are provided in **Appendix C**.

4.2 Main Results

Based on the benchmark protocol above, we compare Look-Before-Move with representative camera planning and trajectory generation baselines, including Cinematographic Camera Diffusion [14], GenDoP [49], and Exceptional Trajectories [5]. We report results from two complementary perspectives: namely quantitative comparison and qualitative visual analysis. FilmAgent-adapted preserves FilmAgent's [43] cinematographer–director negotiation, shot proposals, and iterative review, using

Table 1: Main quantitative comparison with segment-count weighting. Scores include subject perception, intent consistency, trajectory quality, user study, and overall; higher is better. first best and second best mark the strongest two entries.

Set	Methods	Subject perception				Intent consistency				Trajectory quality				User ↑	Overall ↑
		SP1 ↑	SP2 ↑	SP3 ↑	Mean ↑	IC1 ↑	IC2 ↑	IC3 ↑	Mean ↑	TQ1 ↑	TQ2 ↑	TQ3 ↑	Mean ↑	Study	
Baseline	CCD	88.45	66.72	69.87	75.01	58.41	85.71	**24.86**	55.96	40.01	86.05	72.88	66.15	3.21	65.62
Baseline	GenDoP+RGBD	82.74	57.80	67.27	69.27	74.30	76.52	22.08	57.30	32.64	80.99	73.29	62.00	2.92	62.75
Baseline	GenDoP+noRGBD	83.75	59.14	67.00	69.96	72.53	78.17	24.17	57.91	36.13	81.25	**73.94**	63.41	3.01	63.63
Baseline	FilmAgent-adapted	75.47	55.24	65.92	65.54	78.18	59.90	24.77	54.28	40.08	73.82	63.04	59.09	—	59.64
Director	ET+Director A	59.69	36.82	59.72	52.08	**85.92**	63.17	10.80	52.92	40.81	56.96	67.64	55.00	2.28	53.28
Director	ET+Director B	61.34	38.60	60.43	53.46	85.90	62.98	17.20	54.98	39.69	61.11	67.18	55.87	2.35	54.71
Director	ET+Director C	57.40	43.54	51.22	50.72	79.20	69.80	19.63	55.00	38.86	57.51	67.59	54.55	2.25	53.26
Ours	Look-Before-Move	**90.02**	**66.97**	**72.59**	**76.53**	78.15	**90.30**	23.26	**63.90**	**62.44**	**87.73**	73.92	**74.65**	**4.08**	**71.70**

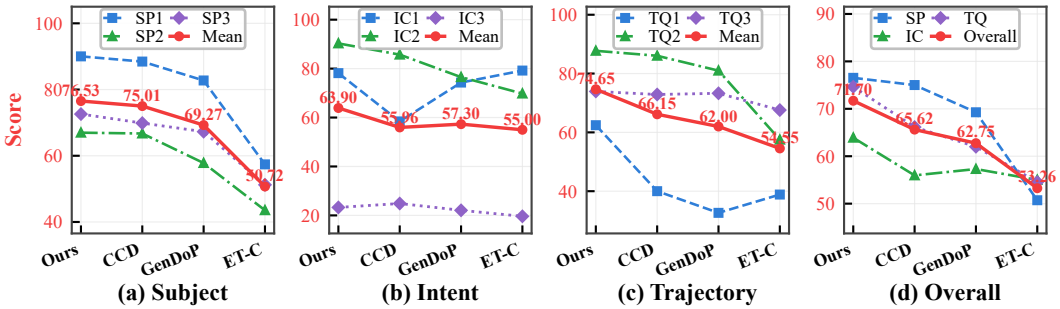


Figure 2: Fine-grained quantitative comparison across subject perception, intent consistency, trajectory quality, and overall performance. Look-Before-Move improves the segment-weighted score while maintaining stronger motion-related metrics than the compared camera planning baselines. The panel exposes where each method succeeds or fails beyond the single aggregate score.

a deterministic adapter to convert its observation plan into Blender-executable trajectories without manual selection or correction. Under the same stories, rendering pipeline, and evaluation protocol, Look-Before-Move achieves 71.70 Overall versus 59.64 for FilmAgent-adapted, with gains of 10.99 SP, 9.62 IC, and 15.56 TQ points.

Quantitative Evaluation. Table 1 and Figures 2 and 5 show that Look-Before-Move achieves the best overall performance under the unified segment-weighted evaluation protocol. Our method obtains an Overall score of 71.70, outperforming the strongest complete baseline CCD by 6.08 points, with clear gains in both intent consistency and trajectory quality. These results show that dynamic 3D camera planning requires more than keeping subjects visible or generating smooth motion. Existing baselines often address only one aspect of the problem, such as subject perception, shot-size matching, or trajectory generation, but struggle to jointly satisfy narrative intent, physical feasibility, and temporal coherence. By first identifying narrative-compliant viewpoints and then grounding them into executable trajectories, Look-Before-Move provides a more complete solution to world visual attention. The results support our central argument: camera planning should be formulated as observation-driven visual evidence organization rather than direct trajectory generation.

Qualitative Evaluation. Figure 3 shows qualitative results across story segments. In the assassin case, Look-Before-Move identifies the intended character, renders the subject at close-up scale, and preserves the action cue. In the car and hospital scenes, it selects viewpoints that capture interactions while maintaining stable framing. In the bedroom wide shot, the generated view keeps both the bed and the sleeping character within the narrative focus, preparing the subsequent plot where the character awakens. These examples show that our method does not merely generate plausible frames, but selects views that are physically valid and narratively specific. Figure 4 further illustrates how the framework maintains narrative-grounded visual attention over time. When the main subject moves across shots, the camera continues to track the protagonist, preserve action cues, and adapt its viewpoint to the evolving scene context. This behavior is enabled by converting the instruction into an observation contract and validating candidate views in the executable 3D world before trajectory grounding. As a result, Look-Before-Move reduces viewpoint drift and translates textual intent into stable visual evidence before camera motion is executed.

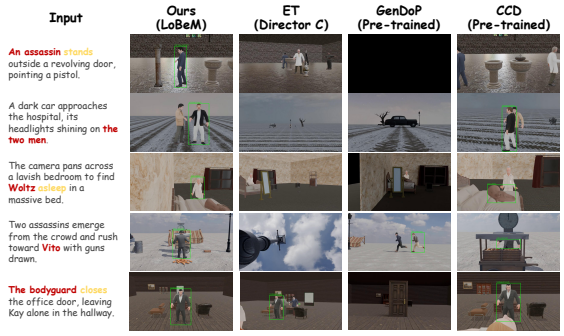


Figure 3: Qualitative comparison on representative story segments. Rows pair narrative instructions with outputs from our method and baselines, testing subject identity, action cues, spatial context, and story-relevant framing.

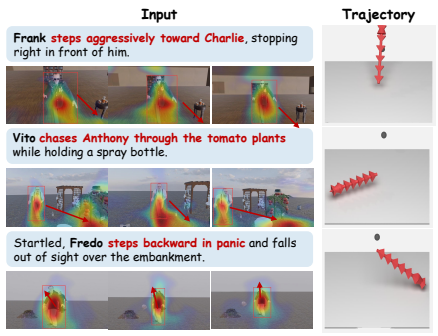


Figure 4: Qualitative visualization of world visual attention. Attention maps and trajectories show narrative intent grounded in evidence before execution.

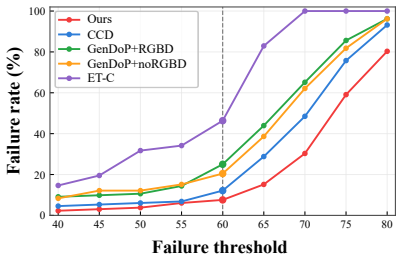


Figure 5: Failure-case counts under unified evaluation. Lower curves indicate fewer segments below each threshold.

Table 2: Ablation results with segment-count weighting between different variants. Each row removes one component under the same evaluator, showing its impact on SP, IC, TQ, and overall score.

Variant	SP ↑	IC ↑	TQ ↑	Overall ↑
Ours	**76.53**	**63.90**	**74.65**	**71.70**
w/o MLS (Fast)	67.54	53.28	60.91	60.52
w/o VLM-R	75.30	60.82	71.86	69.11
w/o TG	69.05	53.24	60.19	60.74
w/o SHA	68.28	53.57	60.23	60.54
w/o PCSJ	64.24	55.00	59.90	59.45

4.3 Additional Results

Ablation Study. Table 2 evaluates five components under shared dataset, prompts, renderer, evaluator, and metrics. MLS, VLM-R, TG, SHA, and PCSJ denote Multi-level Monte Carlo Search, VLM Reflection, Trajectory Grounding, Semantic Height Adjust, and Pre-continuity Story Judge. The Fast variant is *w/o MLS*, where Monte Carlo search and reranking are disabled. The results show that viewpoint search and temporal grounding are the sources of improvement. Removing MLS causes the large drop, reducing the overall score from 71.70 to 60.52, confirming the importance of candidate expansion and reranking before motion planning. VLM-R improves intent consistency through visual-language feedback, while TG, SHA, and PCSJ are crucial for maintaining trajectory quality and temporal coherence. Overall, the ablation validates the core design of Look-Before-Move: narrative-compliant viewpoint selection must be coupled with executable motion grounding.

Contract Analysis. Figure 7(a) compares Direct Trajectory Generation (Direct), Observation Contract + Trajectory Generation (Contract), and LoBeM-full (LoBeM) under identical inputs, model settings, rendering, and evaluation; full results appear in Table 4 (Appendix H.1). The first two use the same trajectory generator and differ only in whether the Observation Contract is used. Compared with direct generation, the Observation Contract improves Overall by 3.36 points, SP by 4.13 points, and IC by 7.35 points, isolating the benefit of structuring what should be observed before motion planning. LoBeM-full further improves Overall by 6.55 points, demonstrating the aggregate benefits of search, geometric verification, grounding, and reflection.

Evaluation Validation. Figure 7(b) places LoBeM first under the three independent IC3 evaluators and third under the original Qwen, with identical inputs and protocols (Appendix I). Human IC also favors LoBeM (4.09), above CCD (3.25), GenDoP (3.10), and Director (2.53) (Figure 7(c); Appendix J).

Trajectory Visualization. Figure 6 provides an additional quantitative view of the trade-off between semantic quality and trajectory quality. The comparison complements the main table by separating whether a method succeeds because it chooses story-relevant evidence, because it produces smooth

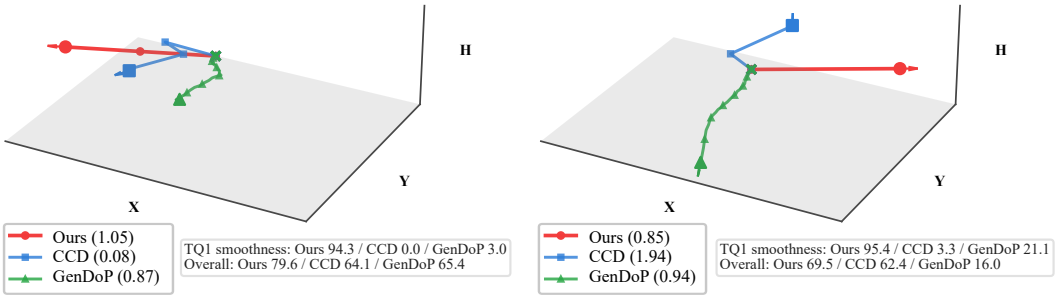


Figure 6: Additional comparison of semantic trajectory quality. Look-Before-Move achieves stronger semantic alignment without sacrificing executable motion quality.

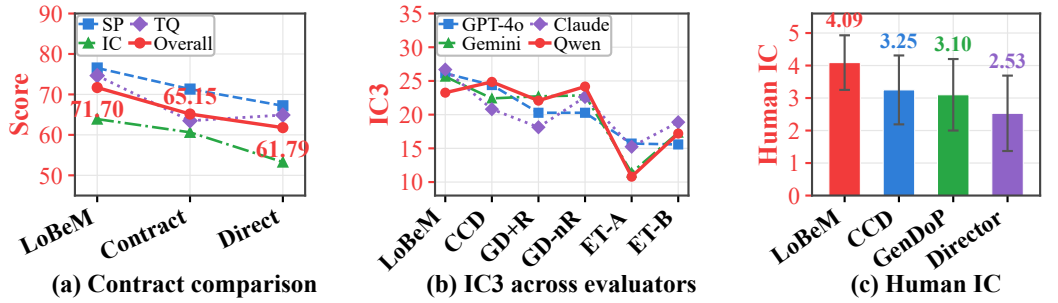


Figure 7: Contract comparison (a), IC3 across evaluators (b), and human IC with SD error bars (c). Full results are in Appendices H–J.

executable motion, or because it balances both. Look-Before-Move is designed to improve this balance: the Look stage improves semantic evidence acquisition, while the Move stage prevents high-quality still viewpoints from becoming unstable motion plans.

Abstract Intent Recovery. Using only executable Contract fields, abstract-intent recovery reaches 77.78% for Abstract-A and 73.15% for its paraphrase Abstract-B, compared with 16.67% for uniform six-way prediction (Table 9). These results show that abstract intents leave recoverable information in executable observation constraints after free-form intent text is removed, while concrete instructions are easier to recover (87.04%). Detailed controls and examples appear in Appendix K.

Efficiency. Separate end-to-end profiling on one RTX 3090 with Blender 4.5.5 LTS gives 26.50 seconds per effective video second for Look-Before-Move versus 18.16 for CCD, including rendering and evaluation (Table 10). This is a $1.46\times$ runtime ratio for a 6.08-point Overall gain; detailed cost profiling appears in Appendix L.

5 Conclusion

We introduced narrative-grounded world visual attention as a novel paradigm for camera planning within dynamic 3D environments, moving beyond passive trajectory generation toward the active organization of visual evidence. To realize this capability, we proposed Look-Before-Move as a collaborative planning framework that effectively translates abstract directorial intent into executable camera behaviors by decoupling spatial viewpoint composition and temporal motion coordination. To facilitate systematic research, we constructed an executable 3D story-world benchmark that provides rigorous geometric and semantic feedback for closed-loop verification. Extensive experiments validate that our approach significantly outperforms existing methods across subject perception, intent consistency, and trajectory quality. Ultimately, this research bridges the critical gap between high-level cinematic intent and foundational physical execution, paving the way for fully autonomous and intelligent visual agents in complex virtual worlds, especially in long-horizon narrative generation and embodied world modeling. These results position camera control as active perception, requiring agents to reason jointly about narrative evidence, 3D observability, and executable motion.

Acknowledgements and Disclosures

We received no funding or other third-party support for this work during the past 36 months. We declare no competing interests and no financial relationships outside the submitted work that could be perceived as influencing this research.

References

[1] Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B Lindell, and Sergey Tulyakov. Ac3d: Analyzing and improving 3d camera control in video diffusion transformers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22875–22889. IEEE, 2025.

[2] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, et al. Vd3d: Taming large video diffusion transformers for 3d camera control. In *International Conference on Learning Representations*, volume 2025, pages 66712–66737, 2025.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[4] Kanzhi Cheng, Yantao Li, Fangzhi Xu, Jianbing Zhang, Hao Zhou, and Yang Liu. Vision-language models can self-improve reasoning via reflection. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8876–8892, 2025.

[5] Robin Courant, Nicolas Dufour, Xi Wang, Marc Christie, and Vicky Kalogeiton. Et the exceptional trajectories: Text-to-camera-trajectory generation with character awareness. In *European conference on computer vision*, pages 464–480. Springer, 2024.

[6] Zahra Dehghanian, Pouya Ardehkhani, Amir Vahedi, Hamid Beigy, and Hamid R Rabiee. Camera trajectory generation: A comprehensive survey of methods, metrics, and future directions. *Computer Vision and Image Understanding*, page 104872, 2026.

[7] Google DeepMind. Gemini 3 Flash – model card. Model card. `https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Flash-Model-Card.pdf`, December 2025. URL `https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Flash-Model-Card.pdf`. API model identifier: `gemini-3-flash-preview`. Accessed 2026-09-30.

[8] Zikang Guo, Benfeng Xu, Xiaorui Wang, and Zhendong Mao. Mirror: Multi-agent intra-and inter-reflection for optimized reasoning in tool learning. *arXiv preprint arXiv:2505.20670*, 2025.

[9] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

[10] Zican Hu, Wei Liu, Xiaoye Qu, Xiangyu Yue, Chunlin Chen, Zhi Wang, and Yu Cheng. Divide and conquer: Grounding LLMs as efficient decision-making agents via offline hierarchical reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 24570–24590. PMLR, 2025. URL `https://proceedings.mlr.press/v267/hu25q.html`.

[11] Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A Ross, Cordelia Schmid, and Alireza Fathi. Scenecraft: An llm agent for synthesizing 3d scenes as blender code. In *ICML*, pages 19252–19282, 2024.

[12] Yuzhou Huang, Yiran Qin, Shunlin Lu, Xintao Wang, Rui Huang, Ying Shan, and Ruimao Zhang. Story3d-agent: Exploring 3d storytelling visualization with large language models. *arXiv preprint arXiv:2408.11801*, 2024.

[13] Pu Jian, Junhong Wu, Wei Sun, Chen Wang, Shuo Ren, and Jiajun Zhang. Look again, think slowly: Enhancing visual reflection in vision-language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9251–9270, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.470. URL https://aclanthology.org/2025.emnlp-main.470/.

[14] Hongda Jiang, Xi Wang, Marc Christie, Libin Liu, and Baoquan Chen. Cinematographic camera diffusion model. In *Computer Graphics Forum*, volume 43, page e15055. Wiley Online Library, 2024.

[15] Glenn Jocher and Jing Qiu. Ultralytics yolo11. https://github.com/ultralytics/ultralytics, 2024. Version 11.0.0.

[16] Bingliang Li, Fengyu Yang, Yuxin Mao, Qingwen Ye, Hongkai Chen, and Yiran Zhong. Tri-Ergon: Fine-grained video-to-audio generation with multi-modal conditions and LUFS control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4616–4624, 2025. doi: 10.1609/aaai.v39i5.32487. URL https://ojs.aaai.org/index.php/AAAI/article/view/32487.

[17] Bingliang Li, Zhenhong Sun, Jiaming Bian, Yuehao Wu, Yifu Wang, Hongdong Li, Yatao Bian, Huadong Mo, and Daoyi Dong. Storyblender: Inter-shot consistent and editable 3d storyboard with spatial-temporal dynamics. In *European Conference on Computer Vision*, pages 130–147. Springer, 2026.

[18] Xinyang Li, Zhangyu Lai, Linning Xu, Yansong Qu, Liujuan Cao, Shengchuan Zhang, Bo Dai, and Rongrong Ji. Director3d: Real-world camera trajectory and 3d scene generation from text. *Advances in neural information processing systems*, 37:75125–75151, 2024.

[19] Bingqian Lin, Yunshuang Nie, Ziming Wei, Jiaqi Chen, Shikui Ma, Jianhua Han, Hang Xu, Xiaojun Chang, and Xiaodan Liang. Navcot: Boosting llm-based vision-and-language navigation via learning disentangled reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5945–5957, 2025.

[20] Guying Lin, Kemeng Huang, Michael Liu, Ruihan Gao, Hanke Chen, Lyuhao Chen, Beijia Lu, Taku Komura, Yuan Liu, Jun-Yan Zhu, et al. Pat3d: Physics-augmented text-to-3d scene generation. In *International Conference on Learning Representations*, volume 2026, pages 150281–150301, 2026.

[21] Zhiqiu Lin, Siyuan Cen, Daniel Jiang, Jay Karhade, Hewei Wang, Chancharik Mitra, Yu Tong Tiffany Ling, Yuhan Huang, Rushikesh Zawar, Xue Bai, et al. Towards understanding camera motions in any video. *Advances in Neural Information Processing Systems*, 38, 2026.

[22] Christophe Lino and Marc Christie. Intuitive and efficient camera control with the toric space. *ACM Transactions on Graphics (TOG)*, 34(4):1–12, 2015.

[23] Jinmei Liu, Haoru Li, Zhenhong Sun, Chaofeng Chen, Yatao Bian, Bo Wang, Daoyi Dong, Chunlin Chen, and Zhi Wang. Beyond the dirac delta: Mitigating diversity collapse in reinforcement fine-tuning for versatile image generation. *arXiv preprint arXiv:2601.12401*, 2026.

[24] Liping Liu, Chunhong Zhang, Likang Wu, Chuang Zhao, Zheng Hu, Ming He, and Jianping Fan. Instruct-of-reflection: Enhancing large language models iterative reflection capabilities via dynamic-meta instruction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9956–9978, 2025.

[25] Xinhang Liu, Chi-Keung Tang, and Yu-Wing Tai. Worldcraft: Photo-realistic 3d world creation and customization via llm agents. *arXiv preprint arXiv:2502.15601*, 2025.

[26] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

[27] Anyi Rao, Xuekun Jiang, Yuwei Guo, Linning Xu, Lei Yang, Libiao Jin, Dahua Lin, and Bo Dai. Dynamic storyboard generation in an engine-based virtual environment for video production. In *ACM SIGGRAPH 2023 Posters*, pages 1–2. 2023.

[28] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[29] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

[30] Zhenhong Sun, Junyan Wang, Zhiyu Tan, Daoyi Dong, Hailan Ma, Hao Li, and Dong Gong. Eggen: Image generation with multi-entity prior learning through entity guidance. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6637–6645, 2024.

[31] Jiong Wang, Fengyu Yang, Bingliang Li, Wenbo Gou, Danqi Yan, Ailing Zeng, Yijun Gao, Junle Wang, Yanqing Jing, and Ruimao Zhang. FreeMan: Towards benchmarking 3D human pose estimation under real-world conditions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21978–21988. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02075. URL `https://openaccess.thecvf.com/content/CVPR2024/html/Wang_FreeMan_Towards_Benchmarking_3D_Human_Pose_Estimation_under_Real-World_Conditions_CVPR_2024_paper.html`.

[32] Xin Wang, Wenhao Wu, Menghao Zhang, Zhi Wang, Kun Shao, Jian Luan, Yang Li, Qing Li, Shangding Gu, Huichi Zhou, et al. Harnesspai: An evolving harness for physical ai. *arXiv preprint arXiv:2609.29166*, 2026.

[33] Yibin Wang, Zhiyu Tan, Junyan Wang, Xiaomeng Yang, Cheng Jin, and Hao Li. Lift: Leveraging human feedback for text-to-video model alignment. *arXiv preprint arXiv:2412.04814*, 2024.

[34] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[35] Yana Wei, Liang Zhao, Kangheng Lin, En Yu, Yuang Peng, Runpei Dong, Jianjian Sun, Haoran Wei, Zheng Ge, Xiangyu Zhang, et al. Perception in reflection. *arXiv preprint arXiv:2504.07165*, 2025.

[36] Hongqiu Wu, Weiqi Wu, Tianyang Xu, Jiameng Zhang, and Hai Zhao. Towards enhanced immersion and agency for llm-based interactive drama. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11166–11182, 2025.

[37] Wenhao Wu, Zhentao Tang, Yafu Li, Shixiong Kai, Mingxuan Yuan, Zhenhong Sun, Chunlin Chen, and Zhi Wang. From conflict to consensus: Boosting medical reasoning via multi-round agentic RAG. In *Forty-third International Conference on Machine Learning*, 2026. URL `https://openreview.net/forum?id=S7lpdz7NAW`.

[38] Zhongze Wu, Hongyan Xu, Yitian Long, Shan You, Xiu Su, Jun Long, Yueyi Luo, and Chang Xu. Detecting any instruction-to-answer interaction relationship: Universal instruction-to-answer navigator for med-vqa. In *Forty-first International Conference on Machine Learning*, 2024.

[39] Zhongze Wu, Xiu Su, Feng Yang, Shan You, Jun Long, and Yueyi Luo. Injection without distortion: Geometrically constrained knowledge enhancement for vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 10897–10905, 2026.

[40] Dejia Xu, Yifan Jiang, Chen Huang, Liangchen Song, Thorsten Gernoth, Liangliang Cao, Zhangyang Wang, and Hao Tang. Cavia: Camera-controllable multi-view video diffusion with view-integrated attention. *arXiv preprint arXiv:2410.10774*, 2024.

[41] Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. Camco: Camera-controllable 3d-consistent image-to-video generation. *arXiv preprint arXiv:2406.02509*, 2024.

[42] Yongzhi Xu, Yonhon Ng, Yifu Wang, Inkyu Sa, Yunfei Duan, Zhenhong Sun, Yang Li, Pan Ji, and Hongdong Li. Sketch2scene: Automatic generation of interactive 3d game scenes from user's casual sketches. *arXiv preprint arXiv:2408.04567*, 2024.

[43] Zhenran Xu, Longyue Wang, Jifang Wang, Zhouyi Li, Senbao Shi, Xue Yang, Yiyu Wang, Baotian Hu, Jun Yu, and Min Zhang. Filmagent: A multi-agent framework for end-to-end film automation in virtual 3d spaces. *arXiv preprint arXiv:2501.12909*, 2025.

[44] Jie Yang, Bingliang Li, Fengyu Yang, Ailing Zeng, Lei Zhang, and Ruimao Zhang. Boosting human-object interaction detection with text-to-image diffusion model. *arXiv preprint arXiv:2305.12252*, 2023. doi: 10.48550/arXiv.2305.12252. URL `https://arxiv.org/abs/2305.12252`.

[45] Jie Yang, Bingliang Li, Ailing Zeng, Lei Zhang, and Ruimao Zhang. Open-world human-object interaction detection via multi-modal prompts. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16954–16964. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01604. URL `https://openaccess.thecvf.com/content/CVPR2024/html/Yang_Open-World_Human-Object_Interaction_Detection_via_Multi-modal_Prompts_CVPR_2024_paper.html`.

[46] Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. Direct-a-video: Customized video generation with user-directed camera movement and object motion. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–12, 2024.

[47] Yandan Yang, Baoxiong Jia, Shujie Zhang, and Siyuan Huang. Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent. *Advances in neural information processing systems*, 38:140319–140351, 2026.

[48] Siyue Yao, Mingjie Sun, Bingliang Li, Fengyu Yang, Junle Wang, and Ruimao Zhang. Dance with you: The diversity controllable dancer generation via diffusion models. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8504–8514, 2023. doi: 10.1145/3581783.3612046. URL `https://doi.org/10.1145/3581783.3612046`.

[49] Mengchen Zhang, Tong Wu, Jing Tan, Ziwei Liu, Gordon Wetzstein, and Dahua Lin. Gendop: Auto-regressive camera trajectory generation as a director of photography. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18229–18239. IEEE, 2025.

[50] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642, 2024.

[51] Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *European Conference on Computer Vision*, pages 523–542. Springer, 2022.

[52] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023.

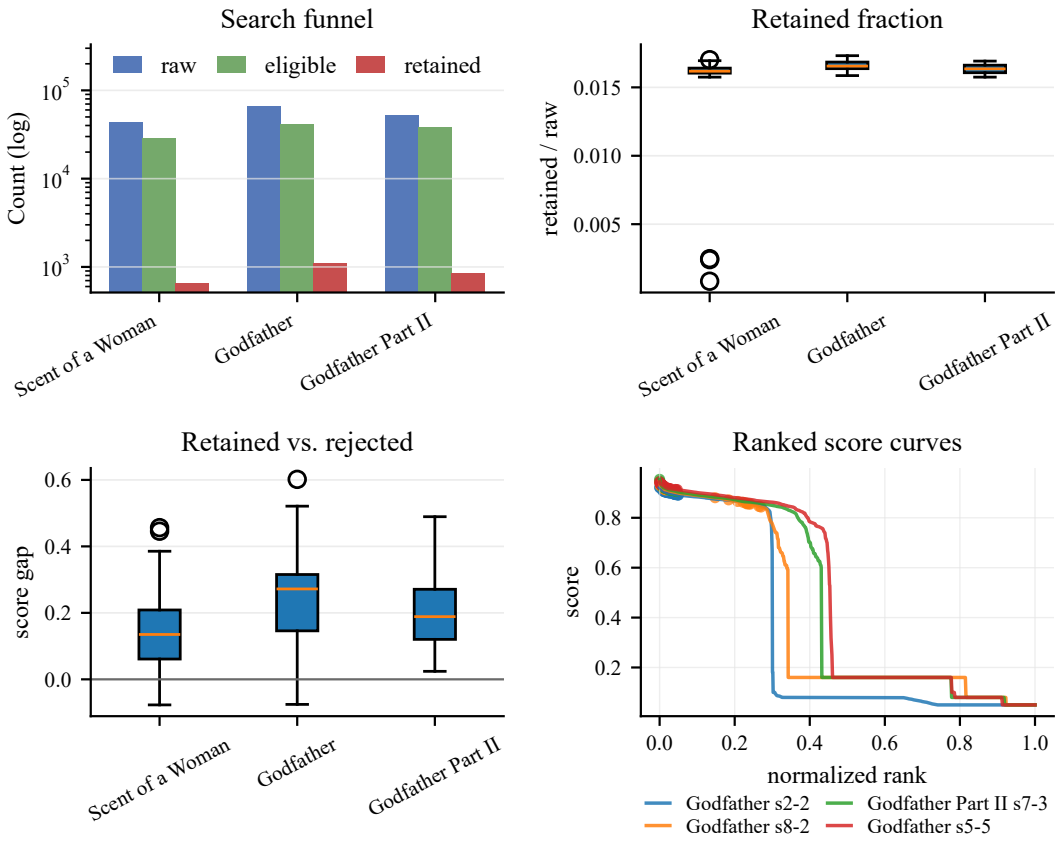


Figure 8: Monte Carlo viewpoint-search summary. The search funnel, retained-fraction distribution, retained-versus-rejected score gap, and candidate ranking curve show how large raw candidate pools are filtered into compact high-quality viewpoint sets.

A Implementation Details

Observation contract. For each shot, Look-Before-Move first parses the directorial intent into a semantic observation contract. The contract specifies the primary acting subject, the expected semantic target region, visibility and occlusion requirements, composition preferences, and temporal action cue. This contract is used to constrain the subsequent candidate search, so the system does not optimize camera motion before deciding what evidence the scene should provide.

Monte Carlo viewpoint search. The Look stage samples a large raw camera-pose pool around the current action context, rejects geometrically invalid or semantically weak views, and retains a compact candidate set for reranking. Figure 8 reports the search statistics over the available candidate-board records. Across 132 shots with visualized candidate boards, the search produces 160,693 raw candidates, keeps 107,478 eligible candidates after geometric and semantic filtering, and retains 2,587 high-quality candidates for final reranking. The small retained fraction indicates that the method is not selecting a viewpoint from a single heuristic anchor; it is searching a large feasible space and compressing it into a small set of narrative-compliant alternatives.

Trajectory grounding and editing. After viewpoint selection, the Move stage grounds discrete viewpoints into continuous camera trajectories in the executable 3D world. Trajectory grounding follows the selected subject, avoids infeasible camera placement, and smooths motion across keyframes. The temporal editing step then schedules observation clips by balancing local information utility against transition inconsistency, which is especially important when consecutive shots focus on different subjects or semantic targets.

B Motivation and Problem Setup

Figure 9 summarizes the motivation behind narrative-grounded world visual attention. Traditional camera control often starts from a specified target, a coarse cinematic rule, or a desired trajectory pattern. In contrast, camera planning inside an executable story world must first determine which visual evidence is narratively relevant: the acting subject, the semantic region, the surrounding context, and the temporal relation to adjacent events. This difference makes the problem more than trajectory synthesis. The camera must first look for an observation that satisfies the story intent and the physical constraints of the 3D world, and only then move through a trajectory that preserves that observation over time.

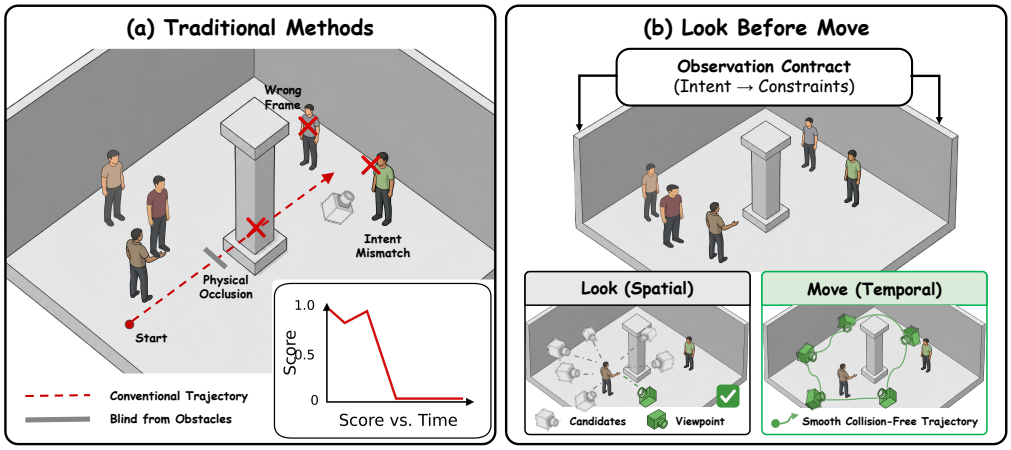
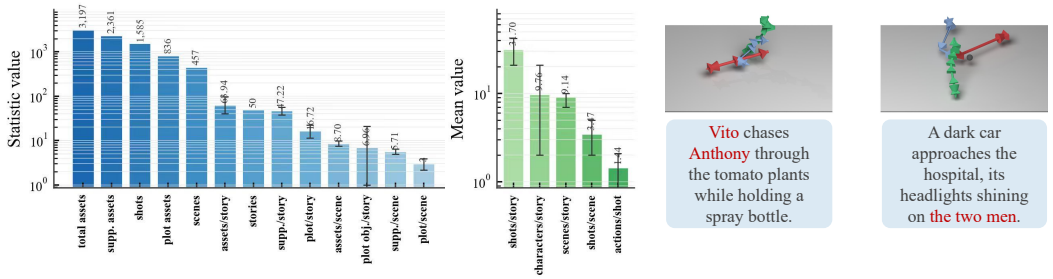


Figure 9: Motivation and task contrast for narrative-grounded world visual attention. The task requires the agent to convert high-level directorial intent into observable subjects, semantic regions, feasible viewpoints, and executable camera motion in a dynamic 3D story world.

C Benchmark Details



(a) Statistics of the CineBoard benchmark **(b) Camera trajectory visualization**

Figure 10: Statistics of the 3D Story World benchmark and representative trajectory visualizations. The left panels summarize benchmark scale, asset composition, character density, and shot-level structure across executable story worlds. The right panels show how natural-language filming instructions are converted into concrete 3D camera paths for dynamic narrative events.

Data hierarchy. Our benchmark is organized as a hierarchy of stories, scenes, and shots. Each story contains temporally ordered narrative events; each scene provides an executable 3D environment with persistent characters, objects, layout, and animations; each shot specifies the local acting subject, semantic target, action cue, and directorial requirement that the camera must satisfy. The benchmark contains 50 stories, 457 scenes, and 1585 shots, with 3197 explicit 3D assets including 836 plot-related assets and 2361 supplementary scene assets. This structure allows us to evaluate camera plans with closed-loop rendering feedback rather than treating viewpoint prediction as a static pose-regression task.

Table 3: Metric groups used in the benchmark. The same definitions are used throughout the main and appendix results.

Group	Sub-metric	Evaluation focus
Subject Perception	SP1	Target capture and sufficient on-screen visibility of the intended subject.
Subject Perception	SP2	Subject identity consistency in multi-character or visually ambiguous scenes.
Subject Perception	SP3	Occlusion robustness when the subject is blocked by geometry, props, or other actors.
Intent Consistency	IC1	Agreement with requested shot scale, camera angle, and composition preference.
Intent Consistency	IC2	Accuracy of semantic target framing, such as face, hands, feet, or full body.
Intent Consistency	IC3	Consistency with the local event description and narrative action cue.
Trajectory Quality	TQ1	Motion smoothness and avoidance of abrupt camera displacement.
Trajectory Quality	TQ2	Executability, stability, and collision-aware spatial behavior in the 3D scene.
Trajectory Quality	TQ3	Temporal continuity between adjacent shot tails and heads.

Evaluation dimensions. We evaluate all methods with the same rendered evidence, evaluator models, and segment-weighted aggregation. The metrics are grouped into Subject Perception (SP), Intent Consistency (IC), and Trajectory Quality (TQ), all normalized to a 0–100 scale. SP measures whether the intended subject remains visible, correctly identified, and robust to occlusion. IC measures whether the rendered view follows the requested shot scale, semantic target, and event description. TQ measures whether the final camera motion is smooth, stable, and temporally coherent across adjacent shots.

Metric computation. For each method, we render the planned trajectory into a sequence of frames. Let shot i contain T_i frames $\{F_{i,t}\}_{t=1}^{T_i}$ and let the observation contract specify the target subject a_i^*, semantic target z_i, shot-size request q_i, and event description e_i. The evaluator extracts a target box $b_{i,t}$ with confidence $c_{i,t}$, area ratio $r_{i,t} = |b_{i,t}|/|F_{i,t}|$, normalized center distance $d_{i,t}$, and a validity flag $m_{i,t}$ indicating whether the intended target is detected. We use $\mathrm{clip}(x) = \min(1, \max(0, x))$ and map every sub-score to 0–100. Invalid or inapplicable terms are removed from the denominator; in particular, TQ3 is not computed for the first shot of a story segment.

$$\mathrm{SP1}_i = \frac{100}{\sum_t m_{i,t} + \epsilon} \sum_{t=1}^{T_i} m_{i,t}\, c_{i,t}\, \mathrm{clip}\left(\frac{r_{i,t}}{\rho_{\min}}\right)(1 - d_{i,t}), \tag{8}$$

$$\mathrm{SP2}_i = \frac{100}{\sum_t m_{i,t} + \epsilon} \sum_{t=1}^{T_i} m_{i,t}\, \mathrm{clip}\left(\frac{M(b_{i,t}, G(a_i^*)) - \max_{a \neq a_i^*} M(b_{i,t}, G(a))}{\gamma_{\mathrm{id}}}\right), \tag{9}$$

$$\mathrm{SP3}_i = 100\, \mathrm{clip}\left(1 - \lambda_m L_i^{\mathrm{miss}} - \lambda_e L_i^{\mathrm{edge}} - \lambda_a \frac{1}{T_i - 1} \sum_{t=2}^{T_i} \left|\log \frac{r_{i,t} + \epsilon}{r_{i,t-1} + \epsilon}\right|\right). \tag{10}$$

Here, SP1 measures effective subject coverage by combining detection confidence, visible area, and centering. SP2 measures identity consistency using ORB feature matching $M(\cdot, \cdot)$ between the detected subject crop and the reference gallery $G(a)$ for each character; it rewards a positive margin over visually competing identities. SP3 penalizes target disappearance L_i^{miss}, edge truncation L_i^{edge}, and abrupt visible-area changes, which are common symptoms of occlusion or unstable framing.

$$\mathrm{IC1}_i = \frac{100}{\sum_t m_{i,t} + \epsilon} \sum_{t=1}^{T_i} m_{i,t}\, \mathrm{clip}\left(1 - \frac{\left|\log \frac{r_{i,t}+\epsilon}{\rho(q_i)+\epsilon}\right|}{\tau_{\mathrm{size}}}\right), \tag{11}$$

$$\mathrm{IC2}_i = \frac{100}{\sum_t m_{i,t} + \epsilon} \sum_{t=1}^{T_i} m_{i,t}\, S_{\mathrm{region}}(F_{i,t}, b_{i,t}, z_i), \tag{12}$$

$$\mathrm{IC3}_i = 100\, S_{\mathrm{vlm}}(e_i, \{F_{i,t}\}_{t \in \mathcal{K}_i}). \tag{13}$$

IC1 compares the observed subject area ratio with the canonical ratio $\rho(q_i)$ for the requested shot size. IC2 evaluates whether the requested semantic region, such as face, hands, feet, or full body, is actually emphasized inside the target crop. IC3 is a blind visual-language judgment over keyframes $\mathcal{K}_i$, where the evaluator checks whether the rendered evidence conveys the event description and directorial intent.

$$\mathrm{TQ1}_i = 100\ \mathrm{clip}\big(1 - \alpha_f \overline{\Delta\phi}_i - \alpha_p \overline{\mathrm{MAD}}_i\big)\,, \tag{14}$$

$$\mathrm{TQ2}_i = 100\ \mathrm{clip}\left(1 - \beta_c \frac{1}{T_i - 1}\sum_{t=2}^{T_i} \|p_{i,t} - p_{i,t-1}\|_2 - \beta_r \frac{1}{T_i - 1}\sum_{t=2}^{T_i} \left|\log \frac{r_{i,t} + \epsilon}{r_{i,t-1} + \epsilon}\right| - \beta_o C_i\right), \tag{15}$$

$$\mathrm{TQ3}_i = 100\ \mathrm{clip}\left(\eta_s\, \mathrm{sim}(F_{i-1,T_{i-1}}, F_{i,1}) \;+\; \eta_p\left(1 - \frac{\|p_{i-1,T_{i-1}} - p_{i,1}\|_2}{\tau_{\mathrm{cut}}}\right)\right). \tag{16}$$

TQ1 measures visual smoothness using Farneback optical-flow acceleration $\overline{\Delta\phi}_i$ and mean absolute pixel difference $\overline{\mathrm{MAD}}_i$. TQ2 measures tracking stability through target-center drift $p_{i,t}$, target-area jumps, and an execution penalty C_i for collisions or invalid camera placements. TQ3 measures cut continuity between adjacent shots by combining image similarity and the jump of the target center from the previous tail frame to the current head frame. The final group scores are simple means over valid sub-metrics,

$$\mathrm{SP}_i = \frac{\mathrm{SP1}_i + \mathrm{SP2}_i + \mathrm{SP3}_i}{3}, \tag{17}$$

$$\mathrm{IC}_i = \frac{\mathrm{IC1}_i + \mathrm{IC2}_i + \mathrm{IC3}_i}{3}, \tag{18}$$

$$\mathrm{TQ}_i = \frac{\mathrm{TQ1}_i + \mathrm{TQ2}_i + \nu_i \mathrm{TQ3}_i}{2 + \nu_i}, \qquad \nu_i = \mathbb{1}[i > 1]. \tag{19}$$

and the shot-level overall score is

$$\mathrm{Overall}_i = \frac{\mathrm{SP}_i + \mathrm{IC}_i + \mathrm{TQ}_i}{3}. \tag{20}$$

For a method m, we report segment-count weighted results over story segments $\mathcal{S}$:

$$\mathrm{Score}(m) = \frac{\sum_{s\in\mathcal{S}} n_s \left(\frac{1}{n_s}\sum_{i\in s} \mathrm{Score}_i(m)\right)}{\sum_{s\in\mathcal{S}} n_s}, \tag{21}$$

where n_s is the number of evaluated shots in segment s. The same aggregation is used for every sub-metric, group mean, and overall score in the main paper.

D Additional Ablation Analysis

Figure 11 expands the ablation table in the main paper with fine-grained sub-metric behavior and pairwise method similarity. Multi-level Monte Carlo Search (MLS) mainly enlarges the feasible viewpoint set and improves both subject perception and intent consistency. VLM Reflection (VLM-R) provides targeted semantic correction, especially for cases where the target region or narrative focus is visually subtle. Trajectory Grounding (TG) turns selected viewpoints into executable motion, Semantic Height Adjust (SHA) improves the vertical alignment of semantic targets such as face or full body, and Pre-continuity Story Judge (PCSJ) reduces temporally inconsistent viewpoint switches before final editing. The counterfactual cases further show that the components fail in different ways rather than contributing an interchangeable score boost. When the search component is weakened, the planner often keeps a plausible subject in view but loses the best semantic region or chooses a viewpoint with fragile visibility. When reflection is removed, errors are more likely to appear as fine-grained semantic mismatches, because the candidate view may be geometrically valid while still emphasizing the wrong body part, object, or interaction cue. When trajectory grounding is removed, the initial still view can remain reasonable, but the rendered clip becomes unstable once the subject or camera begins to move. This separation is useful because it indicates which part of the pipeline should be improved for each error type.

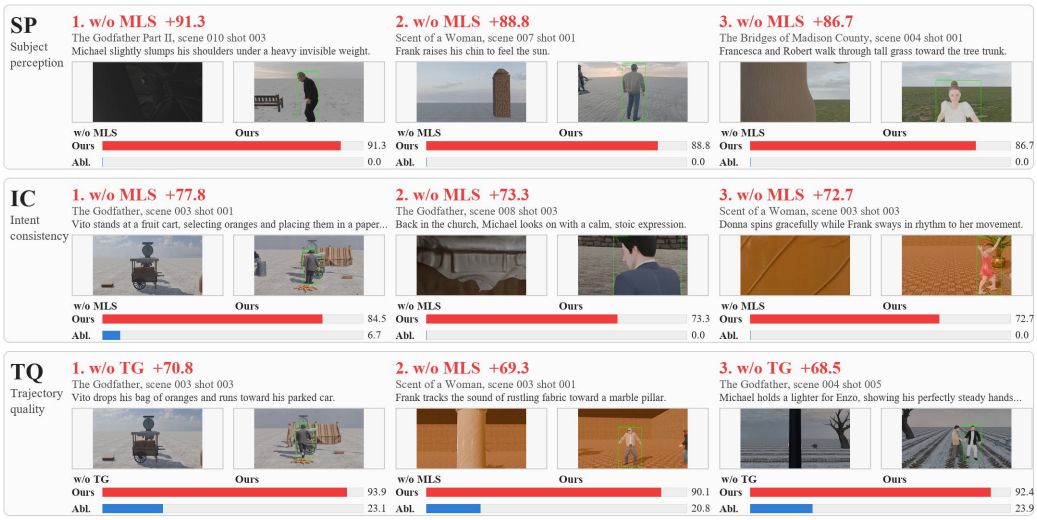


Figure 11: Counterfactual ablation visualization. The metric panels show how each component affects SP, IC, and TQ sub-metrics under controlled component removals.

E Additional Qualitative Results

The qualitative comparisons in the main paper show representative final renderings, while the candidate boards in Figures 14–19 expose the intermediate viewpoint alternatives considered before final selection. These boards are useful because many sampled views can satisfy a coarse visibility constraint, but only a smaller subset simultaneously preserves subject identity, semantic target region, action cue, occlusion constraints, and composition. Each page corresponds to a group of shots and shows the retained candidate set after geometric validation, semantic filtering, deduplication, and reranking.

The boards also make the role of the Look stage visually explicit. For close-up or region-specific requests, the retained views often concentrate around the requested semantic region while still keeping enough body or scene context for identity and action understanding. For wide or motion-oriented requests, the retained views spread over a larger feasible region so that the Move stage can later choose a trajectory that is both stable and narratively informative. This intermediate evidence helps distinguish Look-Before-Move from methods that directly commit to a single camera anchor or trajectory template. The additional boards also clarify how the same narrative instruction can admit several visually acceptable solutions. Some candidates prioritize a stable view of the acting subject, while others reveal more context or preserve a clearer action cue. Keeping these alternatives until reranking gives the planner room to trade off subject scale, semantic target visibility, occlusion risk, and trajectory continuity. This is particularly important for dynamic scenes, where a viewpoint that is optimal at one frame may become unreliable a few frames later. The board evidence therefore complements the final renderings by showing not only what was selected, but what competing observations were rejected.

F Discussion

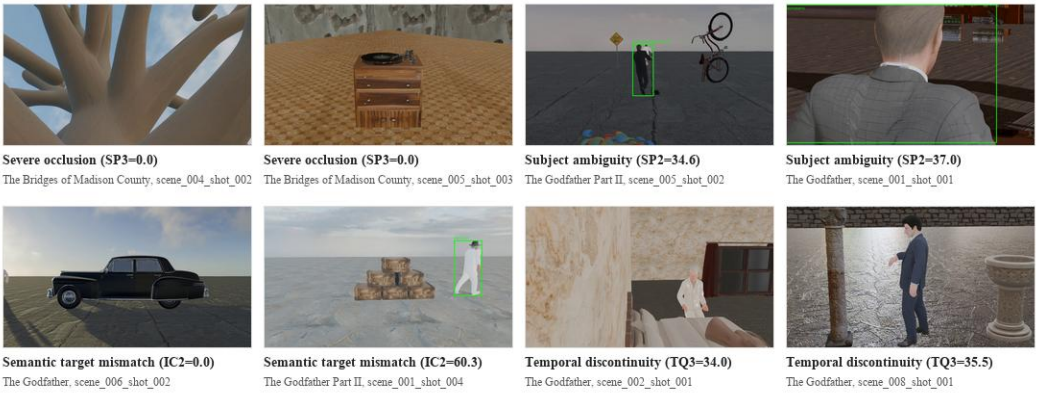


Figure 12: Representative failure cases. Severe occlusion, subject ambiguity, semantic target mismatch, and temporal discontinuity correspond to drops in SP3, SP2, IC2, and TQ3, respectively.

Failure Cases. Figure 12 summarizes representative failure cases according to the most affected metric dimension. Severe occlusion reduces SP when the acting subject is blocked by walls, furniture, props, or other characters. Subject ambiguity reduces SP when the camera locks onto the wrong person or fails to maintain identity in crowded scenes. Semantic target mismatch reduces IC when the requested visual region, such as the face, hands, feet, or full body, is framed imprecisely. Temporal discontinuity reduces TQ when the visual state at the end of one shot is poorly aligned with the beginning of the next shot. These failures are often coupled: an early identity error can lead to later target-region mismatch, while a correct still viewpoint may become invalid once the subject moves behind geometry or another character enters the line of sight. This motivates reporting failures with affected sub-metrics rather than treating them as generic visual artifacts.

Future Work. The observed failures suggest several directions for improvement. First, future planners should maintain stronger scene-level reasoning and uncertainty estimates over target identity, especially in multi-character scenes. Second, viewpoint search should anticipate future occlusions rather than only repairing them after rendering, enabling more reliable long-horizon camera motion. Third, semantic targets should be treated as temporally extended visual evidence rather than local keyframe regions. For example, a face, hand, or full-body request may be visible at one keyframe but become less legible during camera transition. A stronger planner should therefore evaluate whether the requested evidence remains interpretable across the whole clip. In addition, future systems could return structured failure explanations together with rendered clips, allowing the camera agent to revise the specific failed constraint instead of repeating the entire planning process. Longer story arcs also introduce higher-level constraints such as screen direction, repeated object viewpoints, delayed reveals, and story-level visual memory, which remain important extensions of narrative-grounded world visual attention.

Societal Impact. This work may have positive impact by enabling more controllable and interpretable camera planning for virtual production, animation, embodied AI, simulation, and educational storytelling. By grounding camera behavior in explicit narrative intent and executable 3D verification, the proposed framework can reduce manual effort in scene visualization and make cinematic content creation more accessible to users without advanced technical skills. At the same time, more autonomous camera planning systems may be misused to generate misleading, manipulative, or low-quality synthetic visual narratives at scale. Therefore, future deployment should include human oversight, provenance tracking, and appropriate safeguards when the generated camera behavior is used in public-facing media or high-stakes simulation contexts.

G User Study

We conducted a subjective user study to assess the perceptual quality from a human perspective. A total of 95 participants rated camera motion clips generated by Look-Before-Move and other methods (CCD, GenDoP, and Director) across four different stories, with three clips per story presented. To reduce prior bias, amera motion generation method names were hidden from participants during evaluation; within each story, clips from different methods were presented in randomized order as Clip 1–3. Participants rated each clip on three dimensions: subject perception, intent consistency, and trajectory quality, using an integer scale from 0 (extremely poor) to 5 (excellent). The final score per method was averaged across all participants and clips.

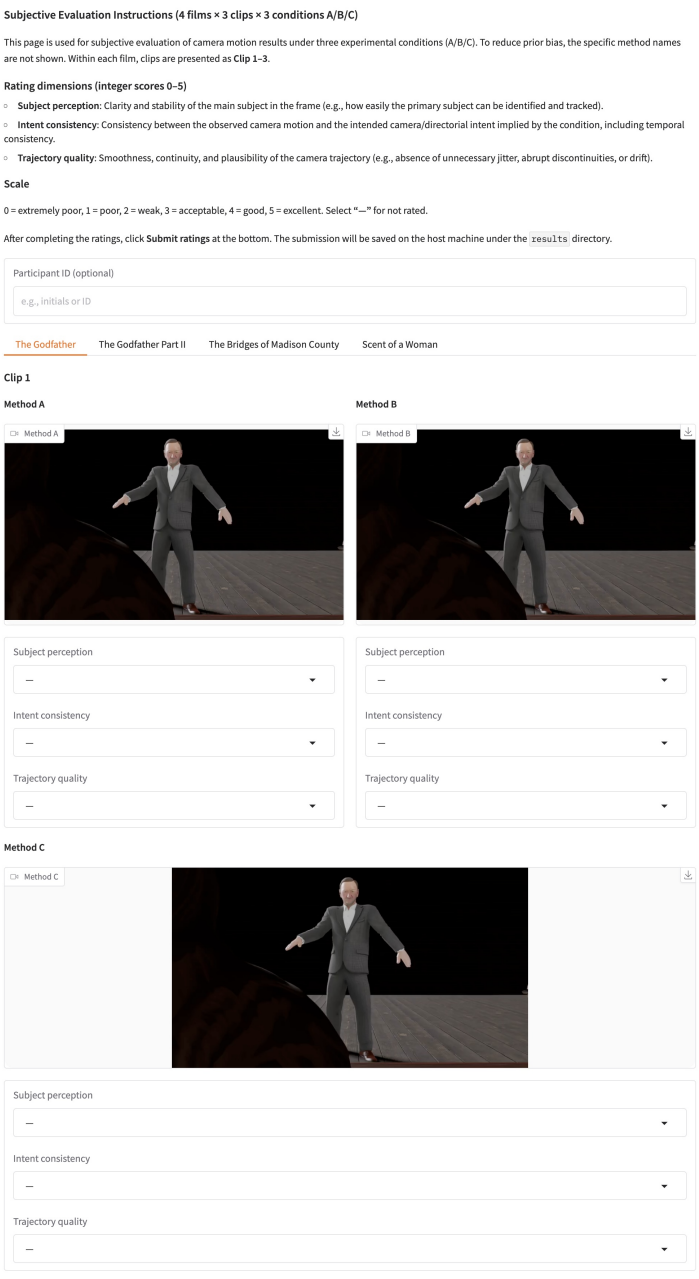


Figure 13: User study interface

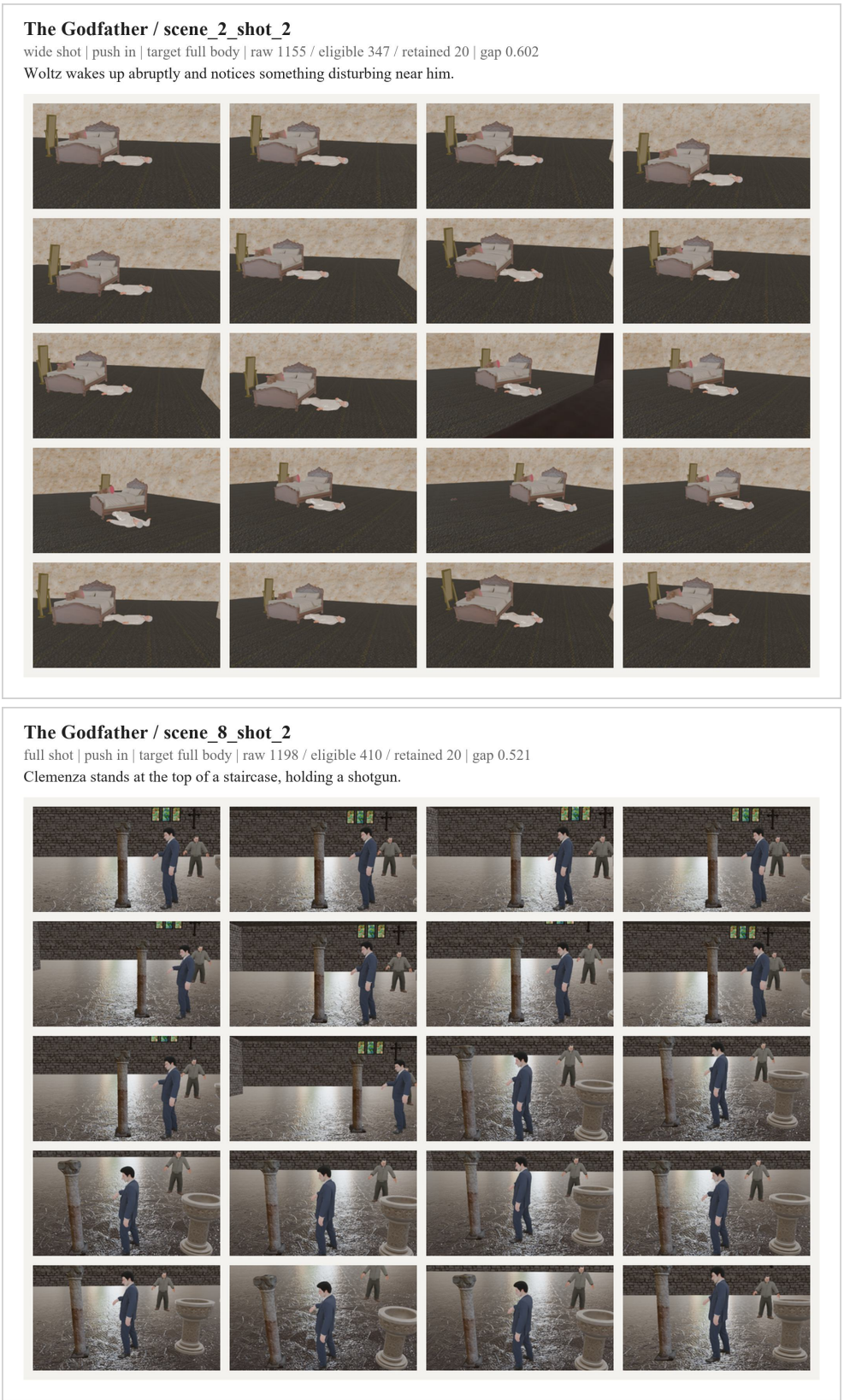


Figure 14: Monte Carlo viewpoint-search candidate boards. Each board reports retained candidate views after sampling, geometric and semantic validation, deduplication, and reranking.

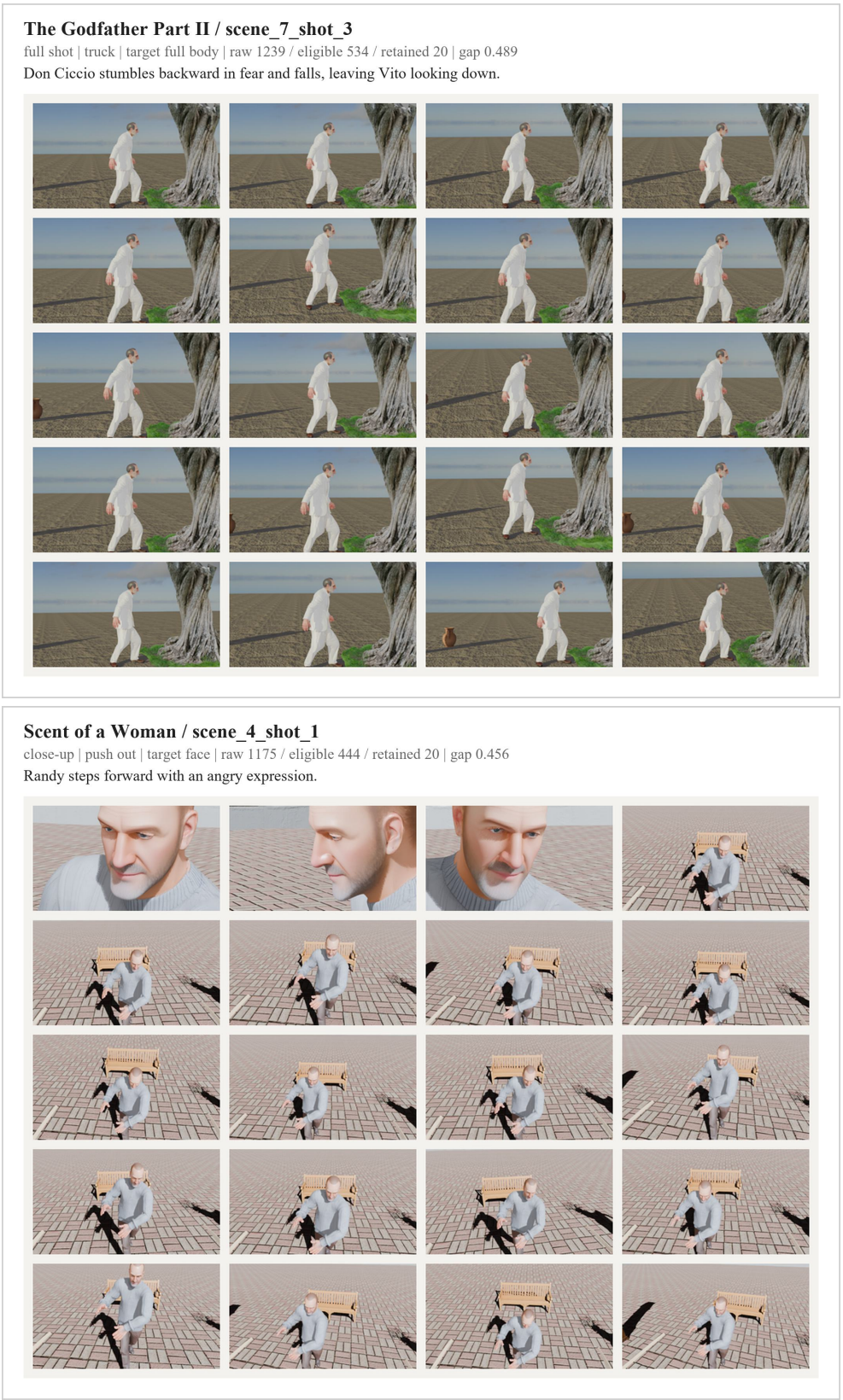


Figure 15: Monte Carlo viewpoint-search candidate boards. Each board reports retained candidate views after sampling, geometric and semantic validation, deduplication, and reranking.

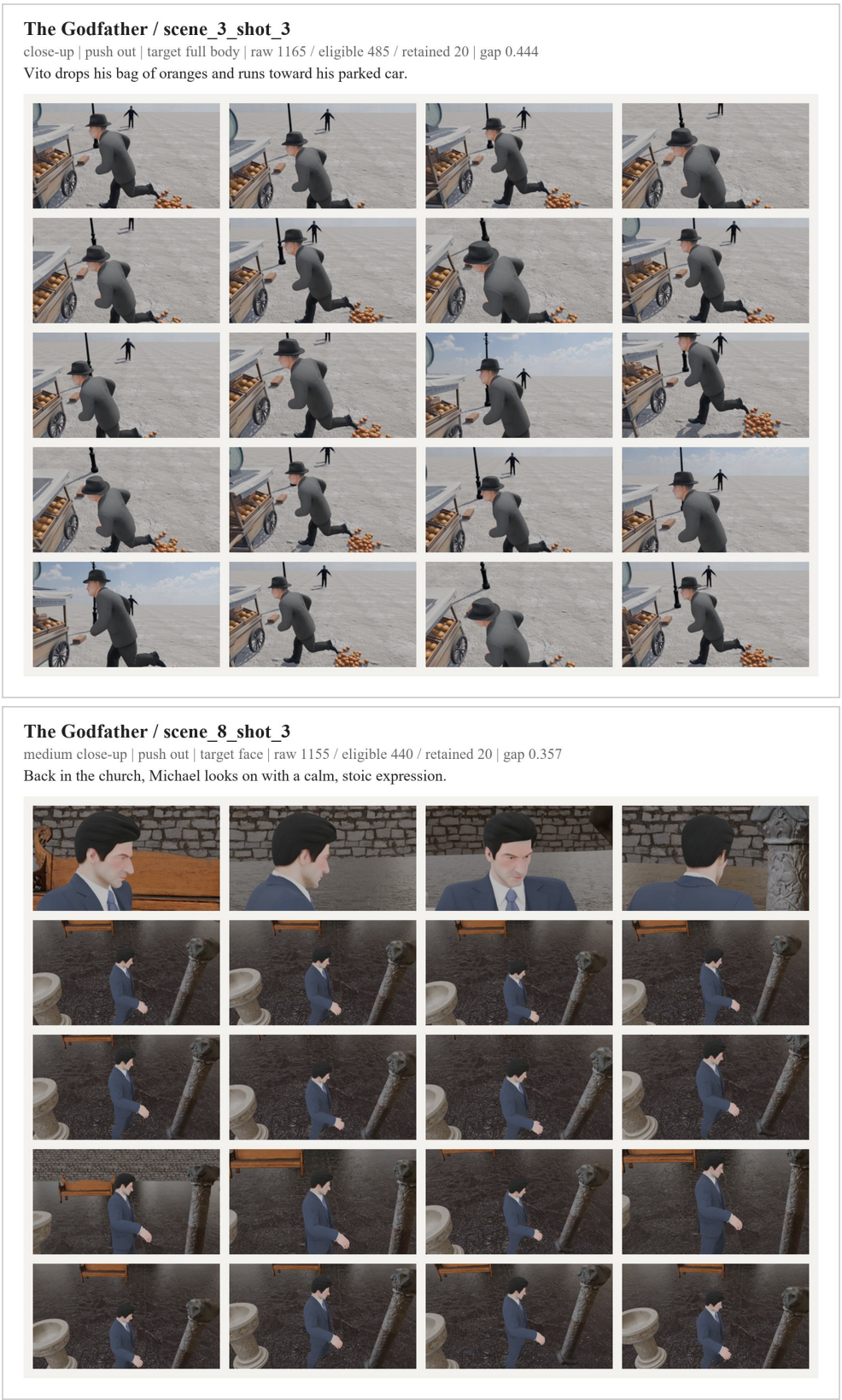


Figure 16: Monte Carlo viewpoint-search candidate boards. Each board reports retained candidate views after sampling, geometric and semantic validation, deduplication, and reranking.

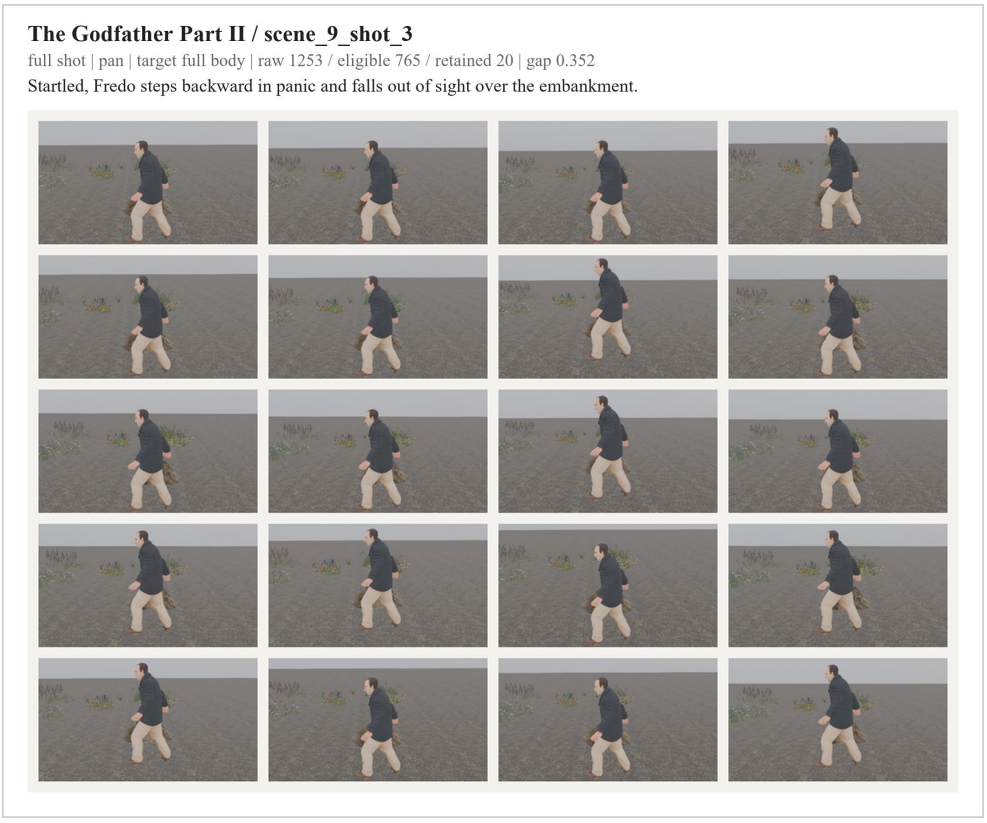


Figure 17: Monte Carlo viewpoint-search candidate boards. Each board reports retained candidate views after sampling, geometric and semantic validation, deduplication, and reranking.

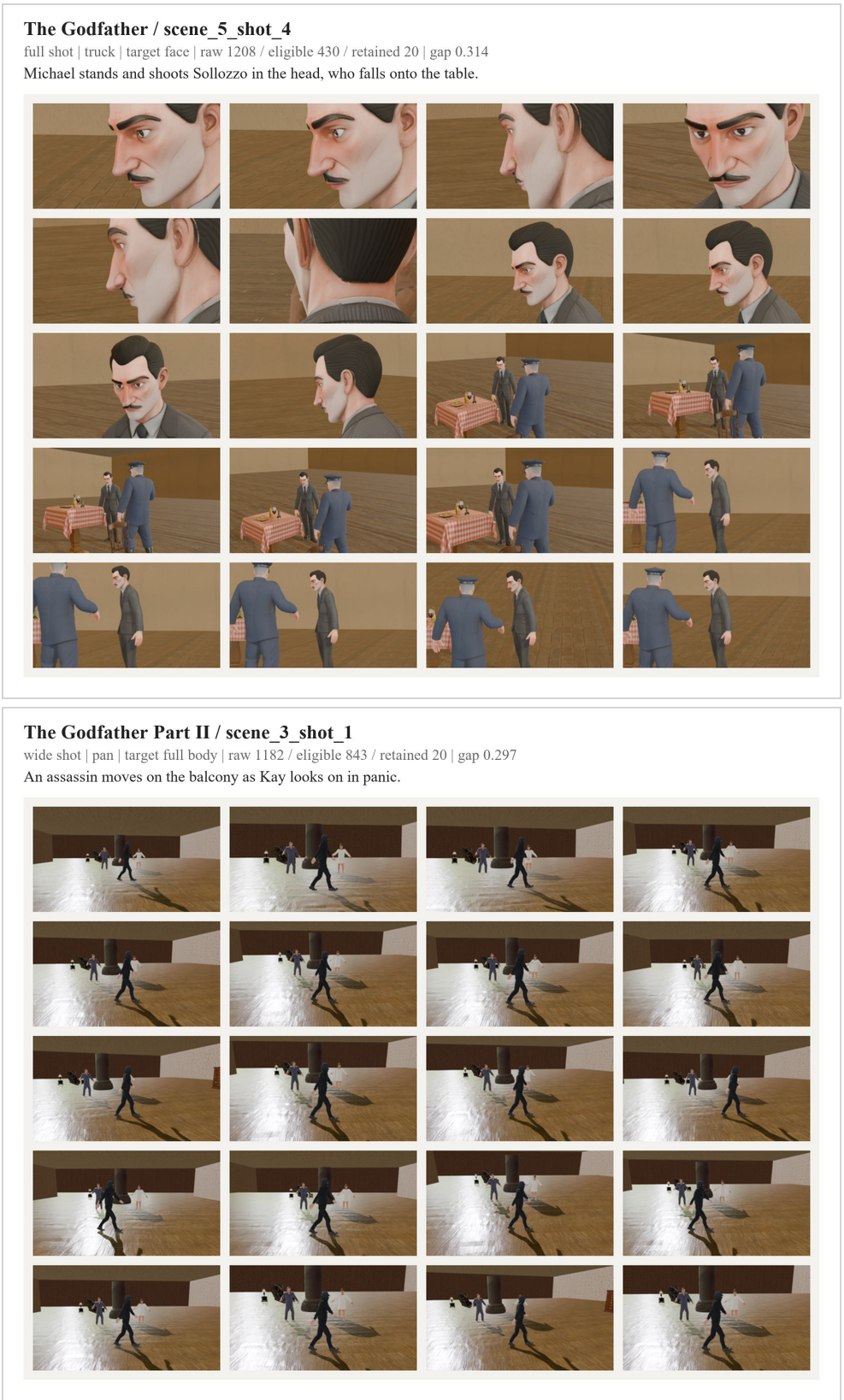


Figure 18: Monte Carlo viewpoint-search candidate boards. Each board reports retained candidate views after sampling, geometric and semantic validation, deduplication, and reranking.

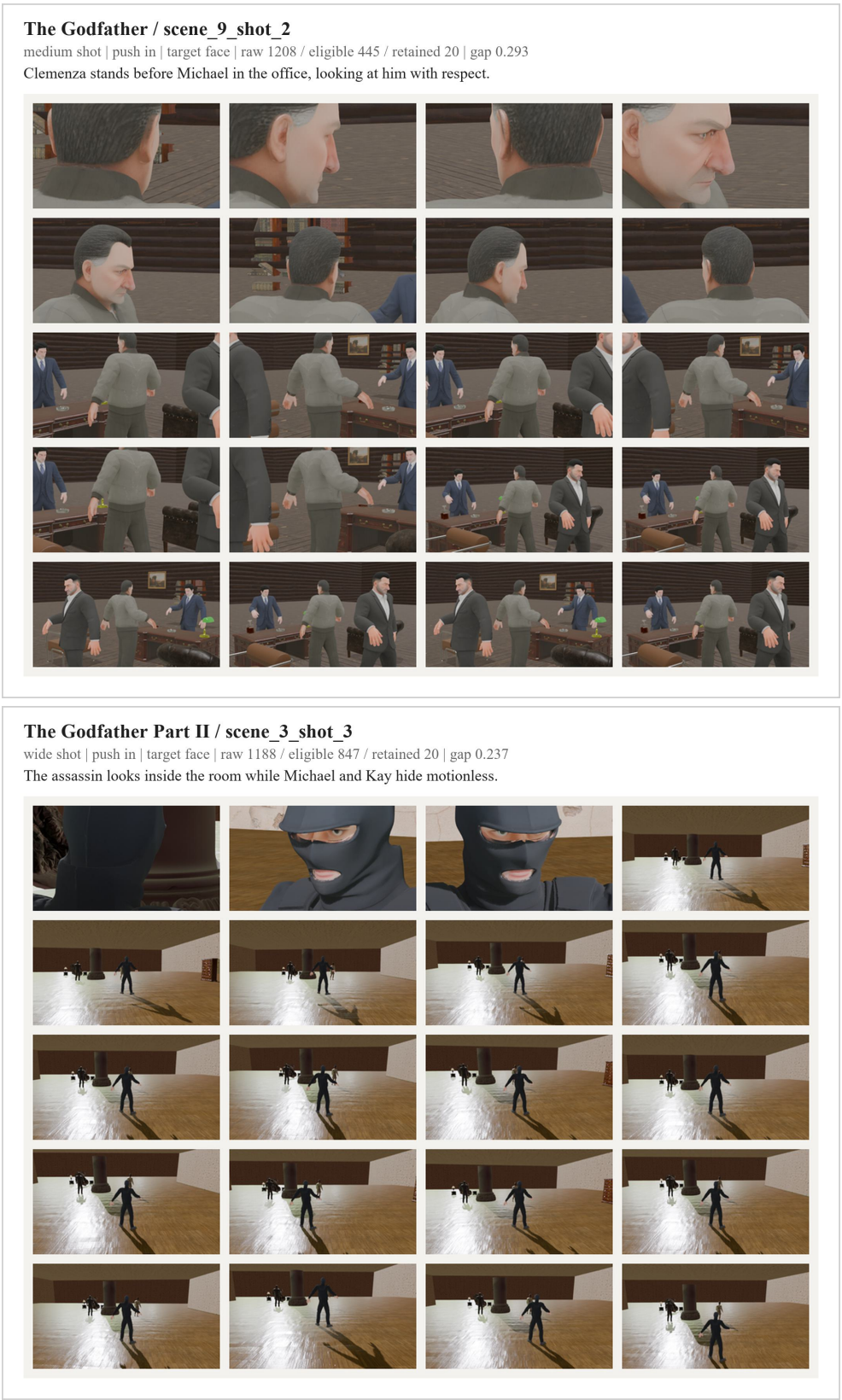


Figure 19: Monte Carlo viewpoint-search candidate boards. Each board reports retained candidate views after sampling, geometric and semantic validation, deduplication, and reranking.

H Observation Contract and Candidate-Budget Analyses

H.1 Observation Contract and Direct Trajectory Generation

The Observation Contract converts each instruction into explicit requirements for the target subject, composition, semantic region, and action cue before trajectory generation. To isolate its contribution, we compare Direct Trajectory Generation, Observation Contract + Trajectory Generation, and LoBeM-full under identical inputs, model settings, rendering, and evaluation. The first two use the same trajectory generator and differ only in whether the Observation Contract is used.

Table 4: Controlled comparison of direct trajectory generation, Observation-Contract-conditioned generation, and the full framework.

Method	SP ↑	IC ↑	TQ ↑	Overall ↑
LoBeM-full	76.53 ± 14.23	63.90 ± 17.79	74.65 ± 15.55	$\mathbf{71.70 \pm 11.94}$
Observation Contract + Trajectory Generation	71.32 ± 23.76	60.63 ± 22.51	63.49 ± 16.68	$\mathbf{65.15 \pm 18.41}$
Direct Trajectory Generation	67.19 ± 23.46	53.28 ± 21.12	64.91 ± 15.79	$\mathbf{61.79 \pm 16.40}$

Compared with direct generation, the Observation Contract improves Overall by 3.36 points, SP by 4.13 points, and IC by 7.35 points, isolating the benefit of structuring what should be observed before motion planning. LoBeM-full further improves Overall by 6.55 points, demonstrating the aggregate benefits of search, geometric verification, grounding, and reflection.

H.2 Candidate-Budget Analysis

We disabled LLM-based reranking and reflection and varied only the number of sampled candidates.

Table 5: Candidate-budget analysis with LLM-based reranking and reflection disabled. Candidates denotes the number of sampled candidates per camera.

Candidates	Overall ↑	IC ↑	Search time (s/shot) ↓
8	66.719	53.868	9.644
16	69.917	55.354	10.662
32	70.064	55.445	12.624
50	**70.319**	**56.271**	13.710

Performance rises from 8 to 16 candidates and largely plateaus over 16–50: 16→50 gains only 0.402 Overall and 0.917 IC while search time rises 28.6%. Thus, the full-framework gain is not explained by pose-pool size alone.

I Independent IC3 Evaluation

We recomputed IC3 on the complete test set using GPT-4o, Gemini-2.5-Flash, and Claude-Sonnet-4. All evaluators received the same keyframes, event descriptions, scoring prompt, and required output format. Therefore, the evaluator model was the only experimental variable.

Table 6: IC3 results under independent evaluator models and the original Qwen evaluator. Gemini, Claude, and Qwen denote Gemini-2.5-Flash, Claude-Sonnet-4, and Qwen2.5-VL-7B-Instruct, respectively.

Method	GPT-4o	Gemini	Claude	Qwen
Look-Before-Move	**26.13**	**25.64**	**26.69**	23.26
CCD	24.39	22.41	20.81	**24.86**
GenDoP+RGBD	20.28	22.72	18.13	22.08
GenDoP-noRGBD	20.28	22.86	22.62	24.17
ET+Director A	15.71	11.40	15.24	10.80
ET+Director B	15.58	17.33	18.88	17.20

Under identical inputs and scoring protocols, these results indicate that LoBeM did not receive a preferential advantage from the original Qwen evaluator. Its relative advantage persists when evaluators from different model families are used, indicating that our conclusion does not result from adaptation to a single evaluator or from self-evaluation bias.

J Human IC Ratings and Automatic-Evaluation Alignment

The blinded, randomized user study involved 95 participants who rated SP, IC, and TQ. Under balanced assignment, each method received 285 valid IC ratings. Table 7 reports the human IC rating distributions and summary statistics.

Table 7: Human IC rating distributions and summary statistics.

Method	Score 0	Score 1	Score 2	Score 3	Score 4	Score 5	Human IC
Look-Before-Move	1	2	10	35	145	92	4.09 ± 0.84
CCD	4	11	45	105	90	30	3.25 ± 1.06
GenDoP	6	14	55	105	80	25	3.10 ± 1.10
Director	15	35	85	95	45	10	2.53 ± 1.16

Look-Before-Move achieves the highest human IC score, exceeding CCD, GenDoP, and Director by 0.84, 0.99, and 1.56 points, respectively. This indicates that human observers consider LoBeM's shots more accurate in presenting target subjects, key actions, and narrative intent.

We compared human and automatic IC at the method-family level using the primary LoBeM and CCD results, the average RGBD and noRGBD results for GenDoP, and the average Director variants.

Table 8: Human and automatic IC at the method-family level.

Method family	Human IC	Automatic IC
Look-Before-Move	4.09	63.90
CCD	3.25	55.96
GenDoP	3.10	57.61
Director	2.53	54.30

Human and automatic IC scores show strong method-family-level agreement (Pearson $r = 0.947$; Spearman $\rho = 0.800$). Both rank LoBeM first and Director last. Only CCD and GenDoP exchange positions, with close scores. Thus, automatic IC reflects relative human judgments of narrative-intent presentation and supports LoBeM's advantage.

K Abstract Narrative Intent in Executable Contracts

The Observation Contract should not be interpreted as directly quantifying artistic quality or replacing human aesthetic judgment. Instead, its purpose is to translate abstract narrative intent into executable visual requirements that can guide camera planning. JSON itself does not quantify or understand artistic intent; it is only a storage format and could be replaced by Markdown, YAML, or another structured representation without changing our method.

To test whether executable Contract fields preserve abstract narrative information, we conducted a controlled experiment where all free-form intent descriptions, rationales, notes, and answer metadata were removed. Evaluators only received executable fields, including subject identity, semantic target, shot scale, screen region, visibility requirements, and camera-motion constraints.

K.1 Abstract Intent Recovery

We evaluated whether executable Contract fields preserve six abstract narrative conditions: Isolation, Intimacy, Dominance, Suspense, Urgency, and Vulnerability. Each sample belongs to one of these six intent categories. Abstract-A and Abstract-B are two different textual expressions of the same

abstract narrative intent, while Concrete represents explicit cinematographic instructions describing the intended observation. Random denotes uniform six-way prediction.

Table 9 reports the complete recovery results.

Table 9: Recovery of abstract narrative intent from executable Contract fields.

Input condition	Description	Accuracy	Macro-F1
Abstract-A	Abstract narrative intent expressed with wording A	77.78%	0.786
Abstract-B	Paraphrased abstract narrative intent	73.15%	0.746
Concrete	Explicit cinematographic instruction	87.04%	0.894
Random	Uniform six-way random prediction	16.67%	0.167

Even after removing all textual intent descriptions, executable Contract fields still allow the underlying narrative condition to be recovered substantially above chance. As expected, concrete instructions are easier to recover, but abstract intents also leave recoverable information in executable observation constraints.

K.2 Beyond Scene Bias

To verify that this signal does not come only from scene content, we compared an Abstract Contract with a scene-only Contract under the same scene. The Abstract Contract is preferred over the scene-only Contract in **88.89%** of decisive comparisons (102 decisive pairs, decisive win rate 91.18%, exact binomial test $p = 4.99 \times 10^{-19}$). This indicates that the Contract does not simply memorize scene semantics; rather, it captures additional observation-level decisions induced by narrative intent.

K.3 Isolation Example

For an "isolation" intent, the Contract does not store an explicit emotion label such as "loneliness". Instead, it translates this abstract narrative requirement into executable visual constraints, including maintaining a single primary subject without secondary focus, using wider framing (LS $\rightarrow$ LS $\rightarrow$ ELS), reducing subject image area, placing the subject at the bottom-right of the screen, and applying outward camera motion (`dolly_out`). These structured decisions allow the abstract intent to influence camera composition and motion through executable parameters. Thus, abstract narrative intent is represented through observable choices in composition, visibility, and camera behavior rather than through an explicit emotion token. Together, these results support our intended claim: the Observation Contract is not an artistic-quality evaluator, but a structured interface that converts abstract narrative intent into executable visual constraints before camera motion generation.

L Efficiency–Performance Trade-off and Runtime Profiling

We profile all methods under identical rendering and evaluation settings using Blender 4.5.5 LTS, with each method run on one RTX 3090 GPU.

L.1 End-to-End Runtime

We report wall-clock runtime, including scene loading, model or LLM latency, viewpoint search, trajectory generation, method-specific rendering, and evaluation, to reflect realistic operation. Runtime per effective second is the total elapsed time divided by the total duration of the final rendered videos. LoBeM is approximately $1.46\times$ slower than CCD, the strongest baseline in Overall performance, and $2.17\times$ slower than the fastest ET variant.

Table 10 reports the runtime results.

L.2 LoBeM Cost Breakdown

LoBeM makes 1.25 LLM calls per effective video second on average, with a mean latency of 6.78 seconds per call. Candidate-viewpoint search requires 8.03 seconds per effective video second,

Table 10: End-to-end runtime under identical profiling settings.

Method	Runtime (s/effective video second)
LoBeM	26.50
CCD	18.16
GenDoP + noRGBD	14.68
GenDoP + RGBD	16.17
ET + Director A	12.21
ET + Director B	12.26
ET + Director C	12.31

approximately 30% of the total runtime. Render-based verification, trajectory execution, final rendering, and evaluation are included in the total. The component measurements are not strictly additive because LLM latency is averaged per request, whereas total runtime is measured over the complete pipeline.

L.3 Performance–Efficiency Trade-off

The additional cost primarily arises from Monte Carlo viewpoint search, LLM-based planning and reflection, and closed-loop render verification. Relative to CCD, LoBeM increases runtime by approximately 46% while improving Overall by 6.08 points, Intent Consistency by 7.94 points, and Trajectory Quality by 8.50 points.

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: Our claims have been clearly made in the abstract and introduction, which reflect our contribution and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: We organized our limitations in the Appendix F, including occlusion handling, multi-character identity ambiguity, semantic target mismatch, and temporal discontinuity.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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 does not present formal theoretical results such as theorems, propositions, or lemmas.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

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: We disclose the evaluated models, benchmark, prompting format, evaluation protocol, and hardware setup to support reproduction of the main results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: [Yes]

Justification: The evaluation data are based on publicly available benchmarks. We will release the code and datasets upon publication, together with detailed implementation instructions, to support faithful reproduction of the main experimental results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

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: We disclose the benchmark construction, baselines (CCD, GenDoP, ET/Director), metric computation protocols (SP, IC, TQ), segment-count weighted aggregation, and hardware/software setup. Details are provided in Section 4 and Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

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: [Yes]

Justification: All metrics are reported under a unified segment-count weighted evaluation protocol. Results are shown in Table 1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: All videos are rendered at 24 FPS on a workstation equipped with one NVIDIA RTX 5090 GPU, as described in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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: This research is based on virtual 3D scenes using publicly available story scripts and executable 3D environments, without collecting real personal data, and fully conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper includes a discussion of potential impacts in the Appendix F.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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: [N/A]

    Justification: The paper does not release pretrained models, image generators, or scraped data with high misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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: We properly cite the original sources of all tools and models used, including StoryBlender, YOLO11x, ORB, Qwen2.5-VL, and the Gemini model series.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We construct and introduce the dynamic 3D Story World Benchmark (CineBoard), covering 50 stories, 457 scenes, 1585 shots, and 3197 3D assets. Construction details are provided in Section 3.1 and Appendix C.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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: [Yes]

    Justification: We conducted a user study with 95 voluntary participants rating camera clips generated by different methods on subject perception, intent consistency, and trajectory quality. Shown in Appendix G

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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: The user study only involves subjective ratings of rendered video clips without collecting personal data or conducting high-risk experiments, and therefore does not require IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Gemini-3-Flash-Preview is used for semantic parsing, observation contract generation, viewpoint reflection, and trajectory planning across multiple agents. Qwen2.5-VL-7B-Instruct is used for blind intent consistency scoring. Details are provided in Section 4.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.