Evaluating Spatial World Modeling in Video Generators via 3D Camera Trajectory Generation

Tianlong Wang
tianlongwang@virginia.edu
University of Virginia

Lehan Yang
lehan.yang@virginia.edu
University of Virginia

Yuheng Liu
yuhenl22@uci.edu
University of California, Irvine

Hanzhang Yuan
ynx9zm@virginia.edu
University of Virginia

Wenhao Zhang
bdu8us@virginia.edu
University of Virginia

Pinqiao Wang
wug7aj@virginia.edu
University of Virginia

Yifan Li
liyifa11@msu.edu
Michigan State University

Yu Kong
yukong@msu.edu
Michigan State University

Sheng Li
shengli@virginia.edu
University of Virginia

Abstract

Recent progress in world models raises a central question: can video generators, as candidate world models, reason about spatial structure rather than only produce plausible motion? Existing evaluations often miss key spatial functions or test them in simplified settings such as mazes, grids, and toy motion patterns. To address this gap, we introduce 3D Camera Trajectory Creation, a floor-plan-conditioned task where a model must generate both a plan-style video and a camera pose sequence under indoor structural constraints. We build two datasets for this task: a single-target dataset where each path visits only one item and a more realistic multi-task dataset for long tour-like behavior. We introduce a score engine that measures trajectory quality score for diagnostic evaluation. Our analysis shows that video generators learn visually regular spatial cues, especially local free-space perception. However, this ability does not reliably compose into room-level planning. Models still struggle with doorway traversal, correct-room selection, target grounding, and target-facing orientation, suggesting partial spatial world-model behavior without a reliable topological abstraction of space.

1. Introduction

Recent work on world models spans several directions, including future video generation (Liu et al., 2026; Brooks et al., 2024; Wang et al., 2024a), interactive environment simulation (Bruce et al., 2024; Yang et al., 2023; Hong et al., 2025b), future-state prediction (Hafner et al., 2023; Hansen et al., 2023; Assran et al., 2025), embodied planning (Bar et al., 2025; Serpiva et al., 2026), and robot rollout prediction (Zhu et al., 2025a; Shen et al., 2025). These directions share an implicit assumption: world models, or their base models, i.e., the video models, genuinely perceive the world and can perform path planning tasks from that perception. Yet few studies have directly examined whether this assumption holds. In this paper, we ask whether the path-planning ability, exhibited by video models, reflects a natural reasoning pipeline: understand the space, abstract the space into discrete regions, and understand the topological connectivity between them rather than a reflection of the pixel-level priors learned during training. We find that current video models demonstrate non-trivial local perception, reliably identifying open and occupied regions. However, they fail to consistently exhibit the higher-level abilities required to reason about abstractions and connectivity. When tasks demand this structural reasoning, the models continue to apply their perception-level prior, which cannot solve them consistently and reliably.

Recent benchmarks have begun to evaluate spatial reasoning in video and world models (Wang et al., 2026a;b). These works isolate spatial reasoning as an explicit evaluation target. For the path-planning thread, however, limitations in

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

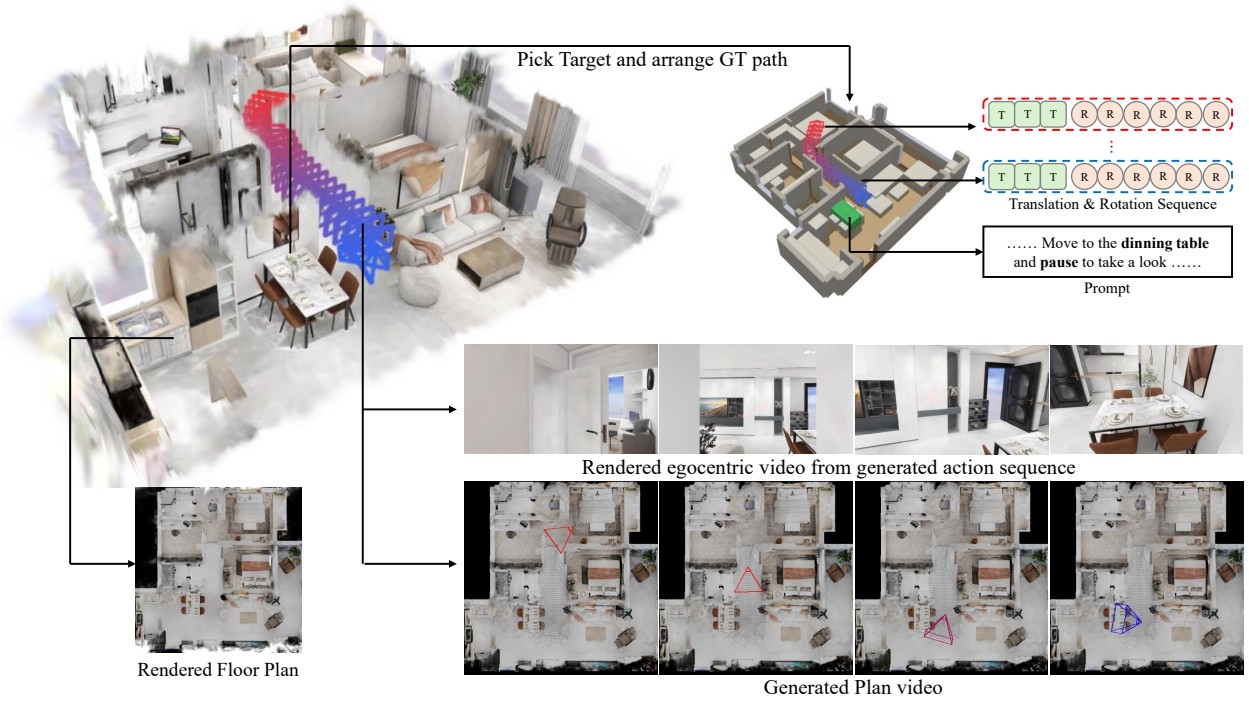


Figure 1. Overview of 3D Camera Trajectory Creation. From a reconstructed indoor scene, we render a top-down floor plan and define target-driven camera paths; the model then generates a floor-plan video with a moving camera marker and the paired translation and rotation sequence.

both task design and evaluation make it difficult to diagnose the source of the observed spatial reasoning. First, the tasks themselves lack the structure needed to require spatial abstraction or connectivity reasoning. Take maze solving as an example: regularly shaped open spaces may allow the model to arrange a route from start to end by relying on patterns learned from training priors, reducing the task to local free-space perception. Second, the evaluation mostly focuses on task completion. A success rate alone, without analyzing where the model fails, can obscure the difference between genuine reasoning and pattern-matched local navigation.

To address this gap, we introduce a 3D camera trajectory planning task, shown in Figure 1. Given a top-down RGB floor plan, a start location drawn on the plan, and a target item described in the prompt, the model generates a trajectory video representing the agent's movement through the space. This task requires spatial abstraction and reasoning about how rooms are connected. The trajectory output supports fine-grained analysis of where and how models fail. The dataset is built from InteriorGS (SpatialVerse Research Team, 2025). The main evaluated model starts from Wan 2.1 1.3B (Wan et al., 2025) and keeps the video backbone largely unchanged. Following prior work on token concatenation in unified video backbones (Shen et al., 2025; Chen et al., 2026b), we enable joint video–pose generation by concatenating camera-pose tokens with video tokens in the same backbone. This paper makes four contributions:

- **Task formulation.** We introduce floor-plan conditioned 3D camera trajectory generation as a functional test of spatial world-model ability.
- **Dataset construction.** We build a single-task dataset to reduce evaluation noise and a multi-task dataset to test long-horizon planning.
- **Evaluation pipeline.** We extend the evaluation with a score engine that provides diagnostic evaluation of spatial reasoning ability.
- **Analysis findings.** Our analysis reveals a clear gap between local spatial perception and route-level reasoning in current video generators.

2. Related Work

Generative Planning and Video World Models. A growing body of work frames decision making as generative modeling. This line includes diffusion planners for trajectory synthesis (Janner et al., 2022; Yoon et al., 2025; Xie et al., 2025), diffusion-based robot motion planners (Carvalho et al., 2023; Shaoul et al., 2024; Xu et al., 2026), and camera trajectory generators that synthesize realistic traversals of 3D scenes from text or cinematic intent (Li et al., 2024; Courant et al., 2024; Zhang et al., 2025c). Recent video world models extend this idea by using video generators to simulate future observations (Hong et al., 2025b; Bruce et al., 2024; Zhu et al., 2025b; Bar et al., 2025). Autonomous-driving systems further use generative world models to produce multimodal futures that align with road layout and scene context (Gao et al., 2024; Wang et al., 2024b; Liao et al., 2025; Xiong et al., 2026). These approaches show that generative models can synthesize coherent futures and action-conditioned rollouts. However, many of them reduce the spatial reasoning burden by either omitting explicit spatial inputs or conditioning on prescribed actions, routes, or controls.

Spatial Reasoning and Navigation in Embodied AI. Navigation methods use several forms of spatial supervision and control. Many vision-language navigation methods predict actions from observations and language instructions (Sridhar et al., 2024; Hong et al., 2025a; Qi et al., 2025; Wang et al., 2025b). Other methods provide floor maps or semantic maps to support navigation decisions (Chen et al., 2026a; Li et al., 2025; Zhang et al., 2025b). These approaches often treat navigation as action prediction rather than video generation. A separate line uses world models for navigation (Liu et al., 2025; Serpiva et al., 2026; Bar et al., 2025). These methods involve video generation, but they often provide detailed routing information, waypoints, or actions as direct input. This design improves controllability, but it leaves open whether a video generator can infer a valid route from scene structure alone.

Evaluating Spatial Understanding in Video Generators. As video world models become more common, recent benchmarks have begun to test whether video generators can reason rather than only render. These benchmarks measure physical consistency (Wang et al., 2026a; Luo et al., 2025; Zhang et al., 2025a; Ziakas et al., 2026), semantic understanding (Wang et al., 2026b; Chen et al., 2025), and goal-directed behavior (Wang et al., 2025a; Newman et al., 2026). However, many evaluations use synthetic scenes, simplified visual layouts, or text-mediated prompts instead of real-world spatial inputs. Our setting asks a different question: can a video generator read spatial structure from a floor plan and produce a valid tour trajectory, both as a video and as a camera-pose sequence, that reaches a specified target?

3. Methods

3.1. 3D Camera Trajectory Creation

We design our task as a diagnostic instrument to separate each layer of spatial reasoning in evaluation. The input is an orthographic top-down floor plan with a frustum rendered at the starting position as the camera, with the target specified in the text prompt (Figure 2). The front face of the frustum indicates the camera's facing direction. From this input, the model generates a trajectory in which the frustum moves and rotates across the floor plan to reach the target. We use floor plans because they contain multiple discrete rooms connected by doorways, providing the structural complexity required for abstraction and connectivity reasoning. The granularity of the trajectory exposes where and how a path breaks down, enabling fine-grained analysis of the source of reasoning. We evaluate three output-format variants—video only, pose sequence only, and joint generation—as a robustness check across configurations common in world-model and planning literature; our primary analyses use the single-task dataset, with a multi-task dataset included as an extension to longer-horizon planning.

Single-task dataset. A 2D RGB floor plan is first rendered from the 3DGS scene, then given a prompt, each record picks a good-view-vantage point nearby the target item. Appendix D.1 provides more details about sampling the vantage points. The path arranger arranges an 81-frame clip path at 24 fps to reach that vantage point. Arrange details are in Appendix D.2. Finally, the path overlay drawn over the floor plan serves as the ground truth video. The single-task corpus contains $49{,}031$ trajectory segments across 707 scenes, with 684 scenes for training and $2{,}763$ segments for testing. To perform evaluation, we set a 60° field-of-view camera at 1.5 m height. Trajectories average 6.55 ± 3.68 m in length, and 75.8% cross at least one room boundary. Table 1 summarizes the task-type composition and cross-room rate. More detailed target room and label composition is reported in Appendix D.

Multi-task dataset. The multi-task dataset extends the single-task setting to longer tour-like behavior, via considerations like distance to target and natural camera rotation. Each clip uses the same basic rendering setup as the single-task

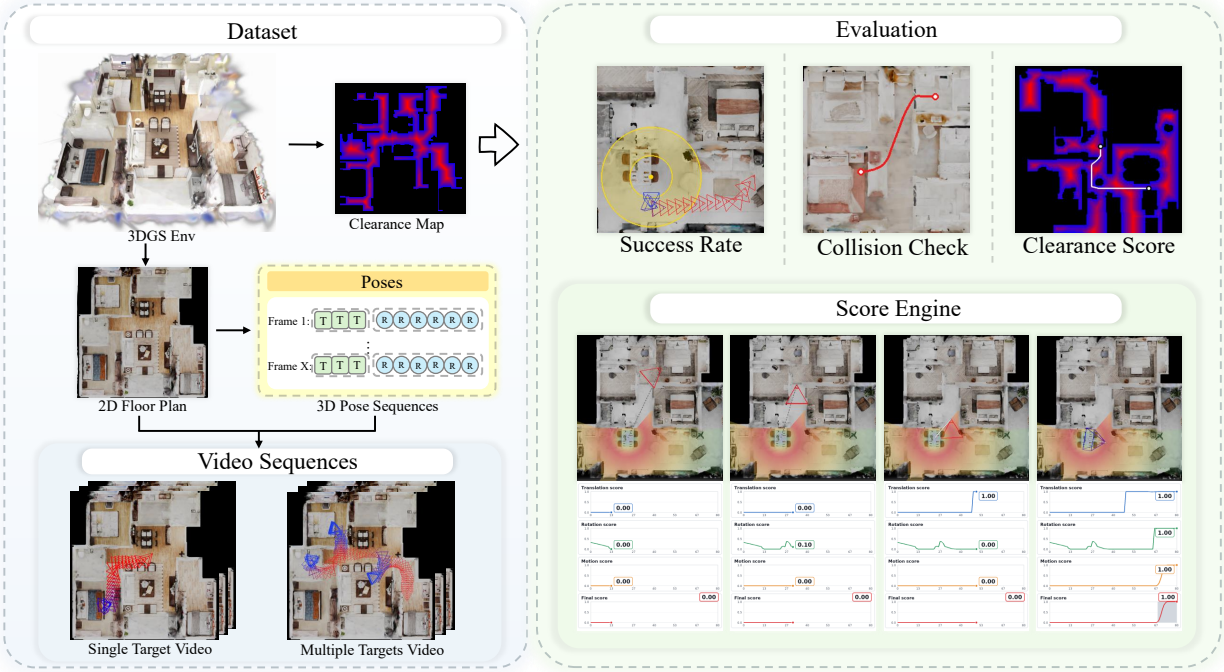


Figure 2. Data-creation and evaluation pipeline. We render a 2D floor plan from a 3DGS scene, generate camera-pose sequences, and overlay each path on the floor plan to create single-target and multi-target tour videos. The evaluation pipeline measures task success, wall collision, clearance, and score-engine trajectory quality over time.

Table 1. Task-type composition of the single-task dataset. Train and Test report segment counts. Cross-room reports the fraction of each task's segments whose target room differs from the spawn room.

Task type	Train	Test	Cross-room
`stop_and_look`	20,000	1,056	63.0%
`room_span`	15,000	1,135	98.9%
`pass_and_look`	11,268	572	66.9%
Total	46,268	2,763	75.8%

Table 2. Chain-length buckets for the multi-task dataset. As chain length grows, the composition shifts from being `pass_and_look`-heavy to being `stop_and_look`-heavy.

Sub-tasks	Clips	% pass	% stop	% span
1–3	19,270	70.5%	8.3%	21.2%
4–6	28,547	48.1%	32.7%	19.1%
7–10	9,426	26.4%	61.0%	12.6%
11–13	332	9.9%	82.6%	7.4%
All clips	57,575	45.8%	36.7%	17.5%

dataset. The dataset contains 57,575 clips across 723 scenes. Sub-task chains range from 1 to 13 entries, with a mean of 4.60 ± 1.97 tasks per record. Tab. 2 summarizes the chain-length buckets and sub-task composition. Appendix D reports the duration of each sub-task type, and lists the ten most frequent item targets for `stop_and_look` and `pass_and_look`.

3.2. Model Design

Model Details. We use Wan 2.1 1.3B (Wan et al., 2025) as the video diffusion backbone. We evaluate three variants. The video-only model uses the base video generator without architectural changes. The pose-only model reuses the main transformer blocks as a diffusion planner over camera-pose tokens. As shown in Figure 3, the video+pose model concatenates video tokens and pose tokens into one sequence. This simple yet effective methods allows the two streams to effectively exchange information between the floor-plan video trajectory and the camera-pose trajectory.

Loss. We simplify the original Wan 2.1 training recipe to a shared flow-matching noise schedule. The video head predicts the flow-matching velocity in VAE-latent space. The pose head predicts the clean camera trajectory directly. Each pose contains a 3D translation vector and a 6D rotation representation. We supervise translation and rotation with

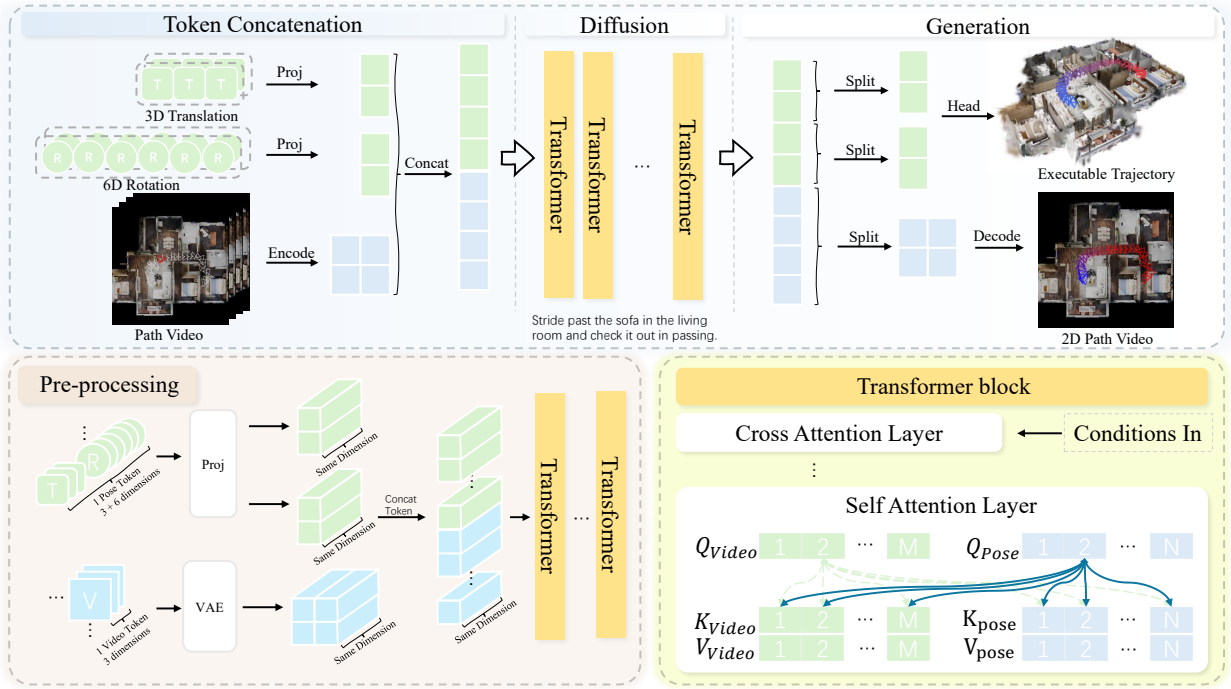


Figure 3. Model pipeline. We concatenate pose tokens with video tokens in a shared transformer backbone, then decode both an executable camera trajectory and a 2D path video. Within each transformer block, self-attention allows video and pose tokens to exchange information.

MSE, and add a geodesic rotation loss after mapping the 6D rotation back to $SO(3)$:

$$\begin{aligned}\mathcal{L} =& \lambda_{\text{vid}}\, \mathbb{E}_{z_0,\epsilon_z,t}\left[\|v_\theta(z_t,t,c)-(\epsilon_z-z_0)\|_2^2\right] \\ &+ \lambda_{\text{tr}}\, \mathbb{E}\left[\|\hat{p}_\theta - p_0\|_2^2\right] + \lambda_{\text{rot}}\, \mathbb{E}\left[\|\hat{r}_\theta - r_0\|_2^2\right] \\ &+ \lambda_{\text{geo}}\, \mathbb{E}\left[d_{\text{geo}}\left(R(\hat{r}_\theta), R(r_0)\right)\right].\end{aligned} \tag{1}$$

Here $z_t = (1-\sigma_t)z_0 + \sigma_t \epsilon_z$ is the noisy video latent, c denotes the text and floor-plan conditioning, $p_0 \in \mathbb{R}^3$ is the ground-truth translation, and $r_0 \in \mathbb{R}^6$ is the ground-truth 6D rotation. The map $R(\cdot) : \mathbb{R}^6 \to SO(3)$ recovers an orthonormal rotation matrix with Gram–Schmidt orthogonalization, and d_{geo} denotes the geodesic angle between two rotation matrices. We set $\lambda_{\text{vid}} = \lambda_{\text{tr}} = \lambda_{\text{rot}} = 1$ and $\lambda_{\text{geo}} = 0.1$ because the geodesic term is measured in radians and ranges up to π. The video-only model uses only the video loss term, while the pose-only model uses only the pose loss terms.

Training Details. We train the video-only and pose-only models for 30K steps with global batch size 8. We initialize the video+pose model from the trained video-only checkpoint and train it for another 30K steps with the same global batch size. At roughly 1 second per step on H200 GPU, a 30K-step run takes 30,000 seconds, or 8.33 wall-clock hours. Since each run uses 4 GPUs, one run costs about 33.33 H200 GPU-hours.

3.3. Evaluations Methods

To evaluate key spatial reasoning abilities, we use three types of metrics: binary success rates, a quality-oriented score engine, and auxiliary measures such as collision and clearance. We extract translation and rotation sequences from the generated videos and account for acceptable trajectory variants in the evaluation pipeline. Appendix C.1 provides more details about the extraction.

Success Rate. Success rate is a coarse, binary check on translation alone: it asks only whether the camera ever reached the target position and performed the requested action. We treat a frame-to-frame step shorter than $\epsilon = 0.02\,\text{m}$ as a stop and any longer step as a move. A `stop_and_look` hit requires two consecutive stop frames near the target item.

Table 3. Results under the rotation-perfect protocol. X denotes the success rate, Y denotes the success rate after filtering wall collisions, and Score measures path quality. The best result in each column is bolded, the second-best result is underlined, and the worst result is italicized.

Model Type	Variant	Modality	X	Y	Score	Wall Coll
Video only	Single Task	Video	**0.712**	**0.608**	**0.293**	<u>0.156</u>
	Mixed Tasks	Video	0.199	0.183	0.266	**0.080**
Video + Pose	Single Task	Video	<u>0.678</u>	<u>0.533</u>	0.252	0.216
		Pose	0.550	0.249	*0.112*	0.535
	Mixed Tasks	Video	0.216	0.180	<u>0.275</u>	0.168
		Pose	0.133	0.105	0.214	0.243
Pose only	Single Task	Pose	0.338	0.081	0.070	0.758
	Mixed Tasks	Pose	*0.095*	*0.007*	0.182	*0.878*

The `pass_and_look` variant requires at least one move step. A `room_span` hit requires two consecutive stop frames inside the target room. Each segment scores $X = 1$ on a hit and 0 otherwise.

Score engine. For each task instance, the score engine returns a continuous final score in $[0, 1]$. The score combines three terms: a translation score s_{tr}, a rotation score s_{rot}, and a motion confidence term m. Fig. 2 shows how the score changes over time. Appendix C.2 shows the detailed definition.

The translation score s_{tr} measures how close the camera is to a useful distance from the target. We define the useful distance as a target-size-dependent annulus around the target: the score is highest inside this band and decays as the camera moves too close or too far away. A line-of-sight gate then sets s_{tr} to zero when a wall blocks the target. The rotation score s_{rot} measures how well the camera faces the target. It decays as the target bounding box moves away from the camera field of view. The motion confidence term m depends on the task type. For `stop_and_look`, it rewards low speed near the target. For `pass_and_look`, it rewards forward motion along a locally straight path. For `room_span`, it combines stop confidence with a stable directional yaw or pitch sweep. We combine these terms frame by frame, smooth the resulting signal over time, and aggregate the best positive event window for each task slot:

$$f_t^v = \mathrm{smooth}(s_{\mathrm{tr}}^v(t)\, s_{\mathrm{rot}}^v(t))\, m(t), \qquad \texttt{segment_final_score} = \max_{v \in \mathcal{V}} \frac{1}{|\mathcal{E}_v|} \sum_{e \in \mathcal{E}_v} \langle f_t^v \rangle_{W_e} . \tag{2}$$

Here, $\mathcal{V}$ contains all valid variants for the task instance. The set $\mathcal{E}_v$ contains the selected positive event windows for variant v, and $\langle \cdot \rangle_{W_e}$ denotes a weighted average over event window W_e.

Wall collision. We mark a record as a full wall collision when the camera pose falls inside a solid wall in any frame. We apply this test to poses extracted from generated videos and to poses predicted by pose-stream models. The test uses wall geometry from the original scene annotations and 3D bounding boxes.

Clearance. Clearance measures how far a trajectory stays from walls and floor-level obstacles. For each scene, we build a geodesic clearance map that assigns each floor-plan pixel a normalized distance-to-obstacle score in $[0, 1]$. Obstacles and outside-map regions receive score 0. At each frame, we project the camera's world XY position onto this map and sample a clearance value c_t. The segment clearance score is the mean of c_t over all frames.

4. Discussion

4.1. Aggregated Result

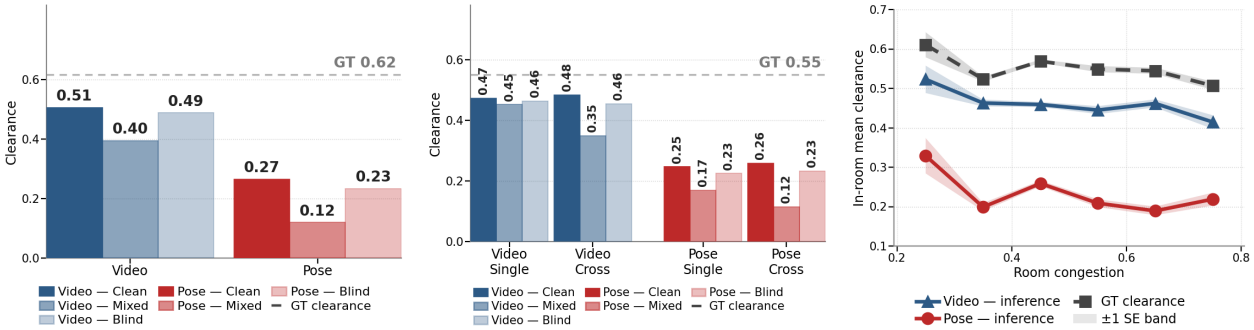
Tables 3 and 4 report aggregate performance across model variants under two evaluation protocols: a rotation-perfect protocol that removes orientation penalties, and a rotation-honest protocol that scores orientation directly. Across both protocols, the single-task video-only model performs best, reaching 71.2% success under rotation-perfect evaluation. Pose-only models perform worst, with the lowest target-reaching rates and the highest wall-collision rates. The video+pose model falls between these cases. Mean scores remain modest in all configurations, around 0.3 under

Table 4. Results under the rotation-honest protocol. X denotes the success rate, X_{LoS} applies the line-of-sight gate, and Y_{LoS} further requires no wall collision. This table focuses on rotation quality in the single-task setting. The best result is bolded.

Model Type	Variant	Modality	X	X_{LoS}	Y_{LoS}	Score
Video only	Single Task	Video	**0.712**	**0.439**	**0.386**	**0.195**
Video + Pose	Single Task	Video	0.690	0.390	0.310	0.161
		Pose	0.533	0.064	0.028	0.005
Pose only	Single Task	Pose	0.335	0.030	0.003	0.003

Table 5. Per-motion-kind success and failure under rotation-perfect and rotation-honest evaluation. `stop_and_look` and `pass_and_look` show a bimodal score tendency.

	rot-perfect		rot-honest	
Motion kind	$\% \leq 0.15$	$\% \geq 0.75$	$\% \leq 0.15$	$\% \geq 0.75$
`room_span`	47.4	1.4	47.4	1.4
`stop_and_look`	36.9	31.8	53.6	16.2
`pass_and_look`	30.1	37.9	45.1	12.9
Overall	40.0	20.1	49.3	9.3



(a) Overall *(b)* By path type *(c)* Vs. congestion

Figure 4. Perception clearance signals. Clean averages collision-free segments, Mixed assigns 0 to full-collision segments, and Blind averages all segments. (a) reports overall clearance; the video stream reaches scores comparable to ground truth. (b) reports clearance by single-room and cross-room tasks; task type has limited effect on clearance. (c) reports clearance by room congestion; all streams follow the ground-truth clearance trend as congestion increases.

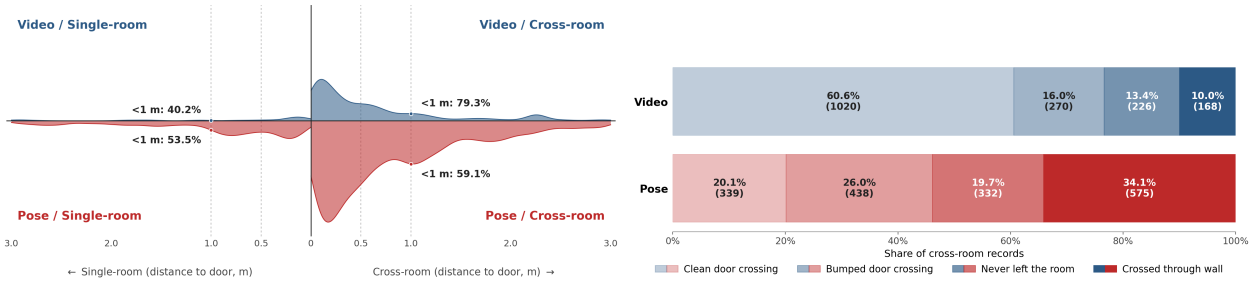
rotation-perfect evaluation and around 0.2 under rotation-honest evaluation. Table 5 shows that the score is bimodal: the model often produces either a high-quality orientation score or a near-failure score, especially on `stop_and_look` and `pass_and_look` segments. The aggregate results show that the models are functional but still far from solving the task consistently. The remainder of this section answers which aspects of spatial reasoning succeed or fail, and where the failures come from, through five lines of evidence.

4.2. Main Evidence

Models perform local floor-plan perception. We measure clearance, defined as the distance from each generated frustum to the nearest wall, on collision-free segments. The video stream achieves a mean clearance of 0.506, compared with the ground-truth value of 0.617 in Figure 4a. This pattern holds across path types, with 0.473 on single-room paths and 0.484 on cross-room paths, compared with matching ground-truth values of 0.554 in Fig. 4b. It also tracks ground-truth patterns across congestion levels, declining as congestion rises and staying within roughly 0.10 of ground truth across bins, as shown in Fig. 4c. These signals indicate that the model has consistent local awareness of free space, the most basic perceptual layer of path planning. This is precisely the capability that pixel-level visual priors are expected to deliver.

Models struggle with room-level reasoning. Cross-room paths require the model to abstract space into discrete rooms and reason about which rooms to traverse. For the video+pose model in single-task evaluation, Figure 8a shows that 24% of cross-room test records end in the wrong room. When we zoom in on failure cases, including collision records, wrong-room failures account for nearly 60% of cross-room failures, as shown in Fig. 8b. This is the dominant failure mode, meaning that the model often arranges the camera path into the wrong room. These results show a deficit in room-level abstraction and raise the question of whether the model's perceptual ability reliably transfers to this higher-level reasoning ability.

Models use doors, but when they fail, they collide right at the door. Cross-room tasks significantly raise the



(a) Density plot of collision distance *(b)* Door crossing breakdown

Figure 5. (a) Area under the curve gives the full-collision rate in each modality. In the video stream, full collision occurs in 4.5% of single-room records and 27.7% of cross-room records; nearly 80% of cross-room collisions occur within 1m of a doorway. (b) Bar length gives the ratio of each cross-room outcome. Clean stands for passing the door and no collision for entire path. Bumped stands for passing the door but collision along the path. Never left the room stands for the camera never leave the spawn room. Crossed through wall stands for camera ends in a different room but has a collision without using the door. For video stream, 76.6% use the door cleanly in total.

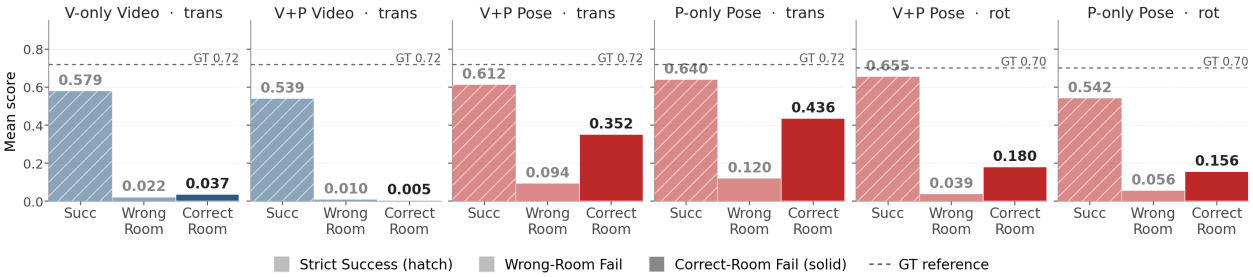


Figure 6. Quality score by outcome bucket for the single task models. Only Pose streams in both models has a reasonable translation score: model arrange the camera relatively closer to the target. All others consideration, translation for video stream, or rotation for pose streams receive low score.

wall-collision rate across modalities. The rate rises from 4.5% to 27.7% for the video stream, and from 31.4% to 55.7% for the pose stream, as shown in Figure 5a. We further find that the video stream often arranges door transitions well. As shown in Figure 5b, clean door crossings and bumped door crossings together account for 75% of cross-room tasks. Yet, among the remaining wall-crossing cases, 80% occur within 1 m of a door, as shown in Figure 5a. This pattern, where the model often crosses doors but usually fails right next to them, shows both the reach and the limit of perception-only path generation. The model can identify regions around doors as visually distinctive openings and thread paths through them. However, perception treats every location on the floor plan in the same way: it does not abstract the door as a required clean passage. It arranges paths near doors as it would through any other free space. This also explains why, when perception misfires, the model collides right next to the door instead of anywhere along the wall.

Target grounding remains shallow. Correct-room failures, where paths find the right destination room but fall short of the target itself, isolate the part of the task that requires fine-grained spatial precision. Figure 6 separates final translation and rotation scores by event type. The video stream's translation score on correct-room failures is capped at 0.022: the camera reaches the right room but stays far from the target. Pose-stream paths move closer, with distance score ≈ 0.5, but rotate poorly, with rotation score ≈ 0.15. Both modalities thus show shallow target grounding even when room-level reasoning succeeds. Full statistics are in Appendix E.3.

Perception does not separate success from failure. Within each trial, perception of free space and walls is essentially identical regardless of outcome: clearance varies by only 0.020 between successful and failed segments, and collision rate by only 0.001, Figure 7. This stability matches our earlier observation that perception applies the same pattern across the floor plan, with no special treatment of where the target lies. Variation in success is therefore not explained by variation in perception. The variable that determines whether a path reaches its target lies outside the perceptual capability that priors so reliably deliver.

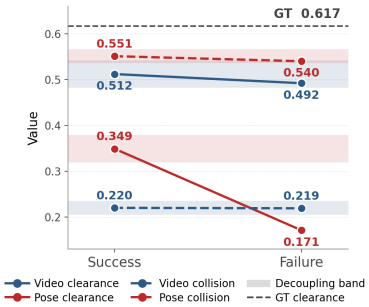
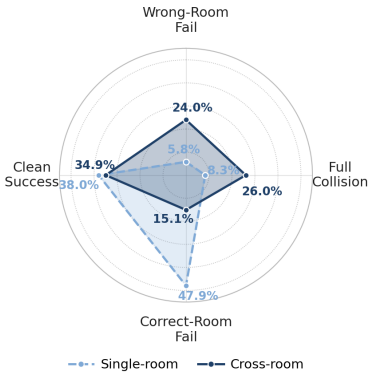
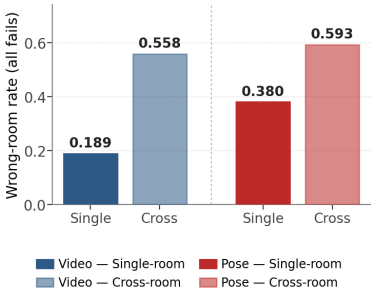


Figure 7. Perception signals conditional on task outcome. Except Pose Clearance, all other evaluation stays within decoupling band. Video clearance ($0.512 \rightarrow 0.492$) and Video collision ($0.220 \rightarrow 0.219$). Pose collision stays inside ($0.551 \rightarrow 0.540$).



(a) Video-stream outcome in single- or cross- room paths.



(b) Wrong-room failure rate among all failure segments

Figure 8. Reasoning-stratified failure composition. (a) Video-stream outcome distribution across single-room (dashed) and cross-room (solid) tasks: cross-room shifts mass from Correct-Room Fail toward Wrong-Room Fail and Full Collision. (b) Wrong-room failure rate among clean-fail segments for both modalities: rising from 0.189 to 0.558 for video and 0.380 to 0.593 for pose between single- and cross-room tasks.

4.3. Qualitative Results

Fig. 11 shows four successful generations: two from the pose stream and two from the video stream. For the pose stream, we render the predicted pose sequence back onto the floor plan so that we can compare it with the video-stream output in the same 2D format. For the video stream, we show the generated floor-plan video directly. The frustum overlay becomes less stable when the camera must represent pitch changes, such as looking up or down. More visualizations are shown in Appendix B.

Human Evaluation. We conduct a human evaluation with 10 volunteers, where each volunteer rates 30 generated paths: 15 pose-stream outputs and 15 video-stream outputs. For each stream, we include 5 manually selected high-quality examples and 10 randomly sampled examples, allowing us to compare best-case generations with typical generations. Volunteers score three aspects on a 1–5 Likert scale: target findability, 2D path quality, and rendered-video quality. The full evaluation interface and results are provided in Appendix A.

The human study shows four main trends. First, target findability increases with target size: large targets receive the highest mean score, followed by medium and small targets. Second, volunteers clearly distinguish hand-picked examples from random examples, with hand-picked pose-stream and video-stream outputs receiving near-ceiling rendered-video ratings. Third, rendered-video scores are more bimodal than 2D path scores, because the rendered view makes target visibility and path failures easier to judge: the target is often either clearly visible or not visible. Finally, human scores sometimes exceed engine scores when a path approaches and views the target but does not satisfy the engine's stricter pass-and-look motion pattern. This suggests that the automatic score captures fine-grained motion requirements, while human raters focus more on whether the target is visually found.

5. Conclusion

We introduced a floor-plan-conditioned camera trajectory task in which a model receives a top-down floor plan and generates both a path video and the corresponding camera pose sequence. We also constructed a dataset and evaluation protocol that separate local free-space perception from route-level reasoning. Our analysis shows a clear gap in current video generators: they can learn local spatial cues, such as open space and obstacle layout, but this perception does not reliably support topological abstraction, goal-conditioned route planning, or executable pose control. Future spatial video models and world models need mechanisms that complement local perception with spatial abstraction and connectivity reasoning.

Acknowledgment

This work is supported in part by the U.S. Office of Naval Research Award under Grant Number N00014-24-1-2668, the National Science Foundation under Grants IIS-2316306 and CNS-2330215, and the National Institutes of Health (NIH) under Grant R01EB293388.

References

Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Bar, A., Zhou, G., Tran, D., Darrell, T., and LeCun, Y. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15791–15801, 2025.

Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., and Ramesh, A. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Bruce, J., Dennis, M. D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Carvalho, J., Le, A. T., Baierl, M., Koert, D., and Peters, J. Motion planning diffusion: Learning and planning of robot motions with diffusion models. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1916–1923. IEEE, 2023.

Chen, H. H., Lan, D., Shu, W.-J., Liu, Q., Wang, Z., Chen, S., Cheng, W., Chen, K., Zhang, H., Zhang, Z., et al. Tivibench: Benchmarking think-in-video reasoning for video generative models. *arXiv preprint arXiv:2511.13704*, 2025.

Chen, K., Huang, Y., An, D., He, J., Su, Y., Liu, J., Liu, N., and Wang, L. Floorplan-vln: A new paradigm for floor plan guided vision-language navigation. *arXiv preprint arXiv:2603.17437*, 2026a.

Chen, L., Gu, Y., and Mao, Q. Univid: Unifying vision tasks with pre-trained video generation models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6754–6763, 2026b.

Courant, R., Dufour, N., Wang, X., Christie, M., and Kalogeiton, V. Et the exceptional trajectories: Text-to-camera-trajectory generation with character awareness. In *European Conference on Computer Vision*, pp. 464–480. Springer, 2024.

Gao, S., Yang, J., Chen, L., Chitta, K., Qiu, Y., Geiger, A., Zhang, J., and Li, H. Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems*, 37: 91560–91596, 2024.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Hansen, N., Su, H., and Wang, X. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

Hong, H., Qiao, Y., Wang, S., Liu, J., and Wu, Q. General scene adaptation for vision-and-language navigation. *arXiv preprint arXiv:2501.17403*, 2025a.

Hong, Y., Mei, Y., Ge, C., Xu, Y., Zhou, Y., Bi, S., Hold-Geoffroy, Y., Roberts, M., Fisher, M., Shechtman, E., et al. Relic: Interactive video world model with long-horizon memory. *arXiv preprint arXiv:2512.04040*, 2025b.

Janner, M., Du, Y., Tenenbaum, J. B., and Levine, S. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Li, J., Huang, W., Wang, Z., Liang, W., Di, H., and Liu, F. Flona: Floor plan guided embodied visual navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 14610–14618, 2025.

Li, X., Lai, Z., Xu, L., Qu, Y., Cao, L., Zhang, S., Dai, B., and Ji, R. Director3d: Real-world camera trajectory and 3d scene generation from text. *Advances in neural information processing systems*, 37:75125–75151, 2024.

Liao, B., Chen, S., Yin, H., Jiang, B., Wang, C., Yan, S., Zhang, X., Li, X., Zhang, Y., Zhang, Q., et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12037–12047, 2025.

Liu, W., Zhao, H., Li, C., Biswas, J., Okal, B., Goyal, P., Chang, Y., and Pouya, S. X-mobility: End-to-end generalizable navigation via world modeling. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7569–7576. IEEE, 2025.

Liu, Y., Lin, X., Li, X., Yang, B., Wang, C., Sunkavalli, K., Hold-Geoffroy, Y., Tan, H., Zhang, K., Xie, X., et al. Omniroam: World wandering via long-horizon panoramic video generation. *arXiv preprint arXiv:2603.30045*, 2026.

Luo, Y., Zhao, X., Lin, B., Zhu, L., Tang, L., Liu, Y., Chen, Y.-C., Qian, S., Wang, X., and You, Y. V-reasonbench: Toward unified reasoning benchmark suite for video generation models. *arXiv preprint arXiv:2511.16668*, 2025.

Newman, K., Zhu, T., and Russakovsky, O. Video models reason early: Exploiting plan commitment for maze solving. *arXiv preprint arXiv:2603.30043*, 2026.

Qi, Z., Zhang, Z., Yu, Y., Wang, J., and Zhao, H. Vln-r1: Vision-language navigation via reinforcement fine-tuning. *arXiv preprint arXiv:2506.17221*, 2025.

Serpiva, V., Sam, J., Simon, C., Amjad, H., Zhura, I., Lykov, A., and Tsetserukou, D. Dreamtonav: Generalizable navigation for robots via generative video planning. *arXiv preprint arXiv:2603.06190*, 2026.

Shaoul, Y., Mishani, I., Vats, S., Li, J., and Likhachev, M. Multi-robot motion planning with diffusion models. *arXiv preprint arXiv:2410.03072*, 2024.

Shen, Y., Wei, F., Du, Z., Liang, Y., Lu, Y., Yang, J., Zheng, N., and Guo, B. Videovla: Video generators can be generalizable robot manipulators. *arXiv preprint arXiv:2512.06963*, 2025.

SpatialVerse Research Team, M. T. I. Interiorgs: A 3d gaussian splatting dataset of semantically labeled indoor scenes. https://huggingface.co/datasets/spatialverse/InteriorGS, 2025.

Sridhar, A., Shah, D., Glossop, C., and Levine, S. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 63–70. IEEE, 2024.

Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Wang, D., Ye, H., Liang, Z., Sun, Z., Lu, Z., Zhang, Y., Zhao, Y., Gao, Y., Seegert, M., Schäfer, F., et al. Target-bench: Can world models achieve mapless path planning with semantic targets? *arXiv preprint arXiv:2511.17792*, 2025a.

Wang, M., Wang, R., Lin, J., Ji, R., Wiedemer, T., Gao, Q., Luo, D., Qian, Y., Huang, L., Hong, Z., et al. A very big video reasoning suite. *arXiv preprint arXiv:2602.20159*, 2026a.

Wang, S., Wang, Y., Li, W., Cai, X., Wang, Y., Chen, M., Wang, K., Su, Z., Li, D., and Fan, Z. Aux-think: Exploring reasoning strategies for data-efficient vision-language navigation. *arXiv preprint arXiv:2505.11886*, 2025b.

Wang, X., Zhu, Z., Huang, G., Wang, B., Chen, X., and Lu, J. Worlddreamer: Towards general world models for video generation via predicting masked tokens. *arXiv preprint arXiv:2401.09985*, 2024a.

Wang, Y., He, J., Fan, L., Li, H., Chen, Y., and Zhang, Z. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14749–14759, 2024b.

Wang, Z., Zhang, J., Ge, J., Lian, L., Fu, L., Dunlap, L., Goldberg, K., Wang, X., Stoica, I., Chan, D. M., et al. Visgym: Diverse, customizable, scalable environments for multimodal agents. *arXiv preprint arXiv:2601.16973*, 2026b.

Xie, A., Rybkin, O., Sadigh, D., and Finn, C. Latent diffusion planning for imitation learning. *arXiv preprint arXiv:2504.16925*, 2025.

Xiong, Z., Ye, X., Yaman, B., Cheng, S., Lu, Y., Luo, J., Jacobs, N., and Ren, L. Unidrive-wm: Unified understanding, planning and generation world model for autonomous driving. *arXiv preprint arXiv:2601.04453*, 2026.

Xu, L., Wong, C., Zhong, Y., Lin, J., Hou, J., and Gao, F. Primitive-based truncated diffusion for efficient trajectory generation of differential drive mobile manipulators. *arXiv preprint arXiv:2604.04166*, 2026.

Yang, S., Du, Y., Ghasemipour, K., Tompson, J., Kaelbling, L., Schuurmans, D., and Abbeel, P. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Yoon, J., Cho, H., Baek, D., Bengio, Y., and Ahn, S. Monte carlo tree diffusion for system 2 planning. *arXiv preprint arXiv:2502.07202*, 2025.

Zhang, H., Gu, X., Li, J., Ma, C., Bai, S., Zhang, C., Zhang, B., Zhou, Z., He, D., and Tang, Y. Thinking with videos: Multimodal tool-augmented reinforcement learning for long video reasoning. *arXiv preprint arXiv:2508.04416*, 2025a.

Zhang, L., Hao, X., Xu, Q., Zhang, Q., Zhang, X., Wang, P., Zhang, J., Wang, Z., Zhang, S., and Xu, R. Mapnav: A novel memory representation via annotated semantic maps for vlm-based vision-and-language navigation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13032–13056, 2025b.

Zhang, M., Wu, T., Tan, J., Liu, Z., Wetzstein, G., and Lin, D. Gendop: Auto-regressive camera trajectory generation as a director of photography. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18229–18239, 2025c.

Zhu, C., Yu, R., Feng, S., Burchfiel, B., Shah, P., and Gupta, A. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025a.

Zhu, Y., Feng, J., Zheng, W., Gao, Y., Tao, X., Wan, P., Zhou, J., and Lu, J. Astra: General interactive world model with autoregressive denoising. *arXiv preprint arXiv:2512.08931*, 2025b.

Ziakas, C., Bar, A., and Russo, A. Grounding generated videos in feasible plans via world models. *arXiv preprint arXiv:2602.01960*, 2026.

Appendix

A. Human Evaluation

We conduct a human evaluation with 10 volunteers. Each volunteer rates 30 generated paths, split evenly between the pose stream and the video stream. For each stream, we include 5 manually selected high-quality examples and 10 randomly sampled examples. This split lets us compare best-case generations with typical generations. For each path, volunteers give three ratings on a 1–5 Likert scale: target findability, 2D path quality, and rendered-video quality. Target findability measures how easily a volunteer can find the requested item along the path. The 2D path rating measures how plausible the path looks on the floor plan. The rendered-video rating measures how well the camera path works in the rendered 3D scene. These three ratings let us test whether our dataset creates meaningful target-search tasks and whether human judgments follow the trend of our automatic score engine. Fig. 10 shows the evaluation interface.

Table 6. Human-evaluation summary. (a): target findability by target size. (b): mean human ratings by stream and subset, compared with the automatic engine score. Human ratings use a 1–5 Likert scale. Engine scores lie in $[0, 1]$.

(a) Q1 target findability by target size.

Target size	#ratings	Mean Q1 ↑
Large	80	**4.61**
Medium	150	3.81
Small	70	2.20

(b) Mean ratings by stream and subset.

Stream	Subset	Engine ↑	Q2 plan ↑	Q3 render ↑
Pose	hand-picked	0.47	4.42 $\pm$0.67	**4.96** $\pm$0.20
	random	0.11	2.82 $\pm$1.27	1.93 $\pm$1.59
Video	hand-picked	0.52	4.56 $\pm$0.64	**4.98** $\pm$0.14
	random	0.21	3.26 $\pm$1.47	2.72 $\pm$1.90

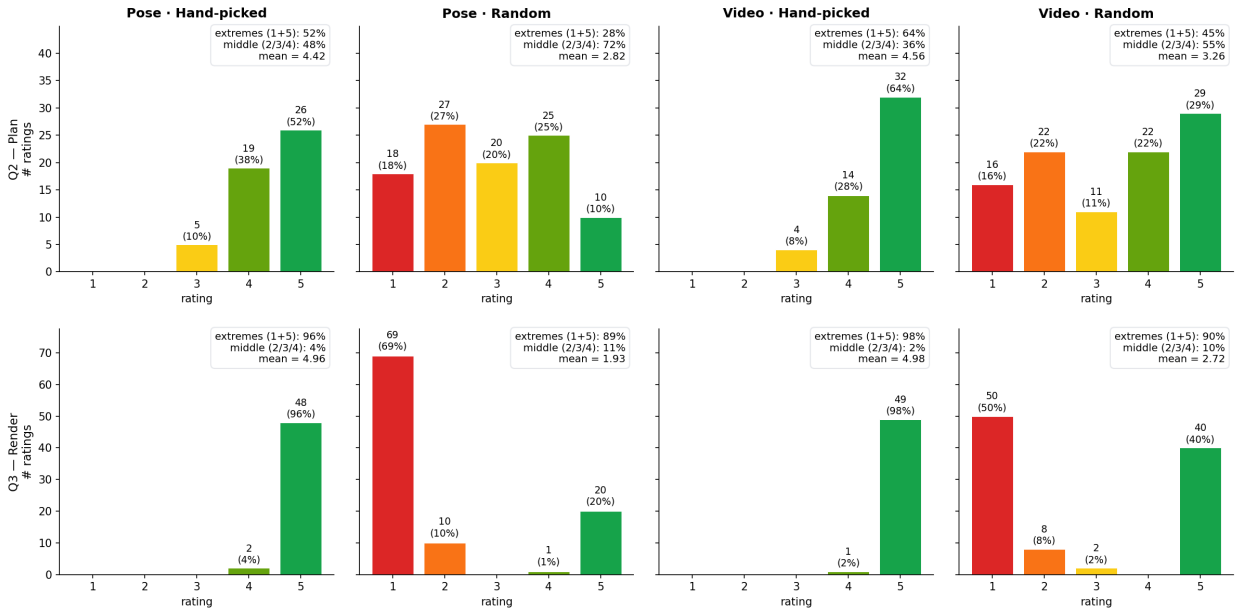


Figure 9. Distribution of human ratings for 2D plan quality and rendered-video quality. Rendered-video ratings show a stronger bimodal pattern because the rendered view makes target visibility and path failure easier to judge.

Tab. 6a shows that target findability increases with target size. Volunteers give large targets the highest mean score, 4.61, medium targets a lower score, 3.81, and small targets the lowest score, 2.20. This trend supports the dataset design: the target-search task becomes easier when the requested object occupies more visible space.

Tab. 6b shows that volunteers distinguish hand-picked generations from random generations. Hand-picked pose-stream and video-stream examples receive near-ceiling rendered-video scores, 4.96 and 4.98 out of 5. Random examples receive much lower rendered-video scores, 1.93 for the pose stream and 2.72 for the video stream. The same table also shows that 2D plan ratings are higher than rendered-video ratings for random examples. This gap suggests that the floor-plan view can make a path look plausible even when the rendered video reveals a failure.

Fig. 9 shows a stronger bimodal pattern for rendered-video ratings than for 2D plan ratings. We expect this pattern because the rendered video gives volunteers a concrete view of the final outcome. In many cases, the camera either reveals the target clearly or fails to reveal it. The 2D plan view gives less direct evidence. It shows the path geometry, but it does not show camera-facing direction, occlusion, object visibility, or whether the target appears in the rendered view. As a result, volunteers use more middle scores for the 2D plan rating.

We also find several records with low engine scores but high human ratings. These cases usually involve a mismatch between the motion required by the score engine and the motion accepted by volunteers. The score engine rewards paths that satisfy a specific event pattern, such as reaching the target region and then looking at the target while walk around the item. Some generated paths instead approach the target while keeping it visible. Volunteers often rate these paths highly because the target appears clear in the rendered video. The engine gives them lower scores because they do not satisfy the stricter motion pattern. This mismatch shows that the score engine captures fine-grained motion requirements that human raters may overlook or treat as less important.

B. More Qualitative Results

Figure 13 further shows failure modes. This includes arranging the wrong target, overlap with the item, or accidentally looked at the item but not actually targeted at the correct item.

C. Evaluation Details

C.1. Dot and Frustum Extraction.

We recover the camera pose from the rendered floor-plan video. Each frame contains a pure-red apex dot for camera position and a colored frustum outline for camera heading. We detect the apex with an RGB threshold, $R > 180$, $G < 50$, and $B < 50$, excluding the top-left text label region. We keep connected components with areas in $[5, 400]$ pixels and select the largest component whose centroid moves less than 40 pixels from the previous frame. When a frame has no valid detection, we linearly interpolate the missing apex position to obtain a continuous pixel track $\mathbf{u}_t = (u_t, v_t)$.

We estimate heading from the frustum pixels around the detected apex. For each frame, we collect pixels in an annulus

$$10 \leq \|\mathbf{p} - \mathbf{u}_t\| \leq 70$$

that satisfy the frustum color test

$$\min_c c \leq 30, \qquad \max_c c \geq 120, \qquad \sum_c c \geq 220.$$

These pixels correspond to the colored frustum sides. We convert each selected pixel $\mathbf{p}_i = (u_i, v_i)$ into an image-plane yaw angle,

$$\theta_i = \operatorname{atan2}\big(-(v_i - v_t),\, u_i - u_t \big),$$

where the negative sign flips the image y axis into the floor-plan coordinate convention. If the frame has at least eight frustum pixels, we compute the heading by circular mean:

$$\hat{\psi}_t = \operatorname{atan2}\left(\sum_{i \in \mathcal{A}_t} \sin\theta_i,\, \sum_{i \in \mathcal{A}_t} \cos\theta_i \right).$$

Frames with fewer than eight frustum pixels have undefined heading.

Finally, we map the apex from video pixels to floor-plan world coordinates using the cached map metadata $(\text{ppm}, x_{\min}, y_{\max})$ and the video-to-map scale factors (s_x, s_y):

$$x_t = x_{\min} + \frac{s_x u_t}{\text{ppm}}, \qquad y_t = y_{\max} - \frac{s_y v_t}{\text{ppm}}.$$

This procedure yields the time-aligned pose sequence $(x_t, y_t, \hat{\psi}_t)$ used by the translation and rotation scoring metrics.

C.2. Score Engine Details

The score engine evaluates each annotated segment as a sequence of per-frame scores. Each segment belongs to one of three slot types: `stop_and_look`, `pass_and_look`, or `room_span`. For item-target slots, the engine measures whether the camera reaches a useful viewing position, whether the target has line of sight, whether the camera points toward the target, and whether the camera follows the expected motion pattern. For `room_span` slots, the engine replaces item-target scores with a room-coverage score.

`distance_score`. `distance_score` measures whether the camera is at an appropriate planar distance from the target. The score reaches its maximum when the camera lies inside the preferred distance band $[r_{\min}, r_{\max}]$. It decreases when the camera is too close or too far from the target. This score measures spatial proximity only; it does not check line of sight or camera orientation.

`visible_score`. `visible_score` is a binary line-of-sight score. The engine ray-casts from the camera position to the target center on the floor plan. If a wall blocks the ray, the score becomes 0, unless an admissible opening, such as a door or window, aligns with the wall hit. This score depends on static scene geometry and ignores camera heading and field of view.

`look_score` and `rotation_score`. `look_score` measures whether the target lies inside the useful camera cone. The engine compares the camera forward direction with the direction from the camera to the target, using yaw and pitch errors. The target bounding box gives a small angular tolerance, so the camera does not need to point exactly at the target center. The engine penalizes vertical misalignment more strongly than horizontal misalignment. We refer to this same quantity as `rotation_score` when separating translation and rotation behavior.

`translation_score`. `translation_score` combines distance and visibility:

$$\texttt{translation_score}_t = \texttt{distance_score}_t \cdot \texttt{visible_score}_t.$$

This score captures whether the camera is in a useful position to observe the target. It becomes zero when the camera is outside the valid distance range or when a wall blocks the target.

`coverage_score`. For `room_span` slots, the engine uses `coverage_score` instead of item-target scores. The engine samples the room polygon on a 0.14 m grid and tracks which sample points fall inside the camera's horizontal field of view over time. The score increases as the camera observes more of the room, with diminishing returns for later coverage gains.

`motion_score`. `motion_score` checks whether the camera follows the motion expected by the slot type. For `stop_and_look`, the engine rewards low planar speed. For `pass_and_look`, it rewards steady forward motion and local path linearity. For `room_span`, it rewards a controlled sweep, requiring low translation speed and consistent yaw or pitch rotation.

`raw_score`, `score_line`, and `segment_final_score`. For item-target slots, the engine first combines position, visibility, and orientation:

$$\texttt{raw_score}_t = \texttt{translation_score}_t \cdot \texttt{look_score}_t.$$

For `room_span` slots, `raw_score` is the `coverage_score`. The engine then smooths `raw_score` over time to produce `score_line`, which reduces single-frame spikes from jitter or grazing line-of-sight rays. The final per-frame segment score is

$$\texttt{segment_final_score}_t = \texttt{score_line}_t \cdot \texttt{motion_score}_t.$$

Thus, an item-target segment receives a high final score only when the camera is close enough to the target, has line of sight, faces the target, and follows the slot-specific motion pattern. For slots with multiple candidate targets, the engine evaluates each candidate separately and keeps the highest-scoring candidate at each frame.

Table 7. Top-10 item-target categories in the single-task dataset.

Target label	n	%
chair	4,554	13.8%
wardrobe	3,257	9.9%
curtain	2,588	7.9%
body pillow	1,910	5.8%
carpet	1,645	5.0%
bedside table	1,213	3.7%
toilet	1,202	3.7%
bath heater	1,138	3.5%
Side table	1,136	3.5%
bed	1,105	3.4%
Top-10 total	19,748	60.0%
Other (147 labels)	13,148	40.0%
Item-target total	32,896	100.0%

Table 8. Top-10 item targets in the multi-task dataset, listed separately for stop_and_look and pass_and_look sub-tasks. Counts are sub-task instances.

stop_and_look		pass_and_look	
Label	Count	Label	Count
chair	16,933	wardrobe	16,199
wardrobe	12,502	chair	13,767
Multi person sofa	6,559	bed	9,936
dining table	5,269	toilet	7,199
carpet	4,518	Ceiling lamp	6,685
bed	4,182	bedside table	6,384
teatable	4,040	shower head	5,361
Ceiling lamp	3,618	Side table	4,771
tv	3,614	Basin cabinet	4,418
Side table	3,614	carpet	3,866

Table 9. Target-room composition of the single-task dataset. The room_span bucket contains clips that name a room rather than an item.

Room type	n	%
bedroom	20,227	41.3%
living_room	17,245	35.2%
bathroom	5,338	10.9%
kitchen	1,740	3.6%
room_span	4,409	9.0%
Total	48,959	100.0%

Table 10. Per-kind sub-task duration for the multi-task dataset.

Kind	n	Mean	Range	Mean (s)
pass_and_look	121,286	201.1	26–616	8.38
stop_and_look	97,147	44.7	9–84	1.86
room_span	46,451	145.0	32–277	6.04

D. Dataset Details

D.1. Vantage Point selection.

For each labeled object we form a thin annulus on the floor centered on the object, with inner radius $0.75\,d$ and outer radius $1.75\,d$, where d is the 3D diagonal of the object's bounding box; the radii are clamped to $[0.6, 4.0]$ m and $[r_{\min} + 0.5, 6.0]$ m so small items keep a usable standoff and large items do not blow up. We sweep this annulus with one ray every 5° (72 rays total) and sample 15 radii per ray, giving roughly 10^3 candidate camera positions per object placed at eye height 1.6 m.

Each candidate is filtered by four tests: it must lie inside the room (with a 5 cm wall margin), not fall inside any other object's box, sit on open floor according to the clearance map (normalized clearance ≥ 0.02), and have unobstructed line of sight to the object's center. The LOS check casts a 3D segment from the candidate to the look point and rejects it if any wall or any other box that is both taller than 0.8 m and larger than 1.3 m in diagonal intersects the segment; the object itself is excluded as a self-occluder. Among survivors on a single ray we keep the top three by clearance, enforcing a minimum radial gap of 0.35 m to avoid near-duplicates, and finally cap each object at 24 vantage points (preferring the farthest).

D.2. Path Arranger

Mixed-task tours. For mixed-task tours the path is grown one item at a time. At each step we score the remaining items by a mix of factors: the item's intrinsic importance, how far it is from the current camera position, how far it is from the previously visited item (so the tour spreads out instead of clustering), how well it lines up with the camera's current heading, how diverse it is in category and region from what has already been seen, and a penalty for items the tour has already visited. The next item is then picked greedily, by softmax sampling over scores, or by an ϵ-greedy mix

Table 11. Distance from each full-collision frame to the nearest door opening. **Coll. rec.** counts records with at least one full collision.

Stream	Path	Coll. rec.	Mean	Med.	p25	p75	< 0.5m	< 1m
Video	Single	42	1.97	1.76	0.16	3.73	34.8%	40.2%
	Cross	431	0.67	0.33	0.10	0.82	58.4%	79.3%
Pose	Single	176	1.47	0.92	0.42	2.31	27.4%	53.5%
	Cross	975	1.17	0.73	0.26	1.63	40.0%	59.1%

of the two. Once an item is chosen, the camera moves to one of its precomputed vantage points by A^* on the clearance map, preferring open routes that keep line of sight to large furniture intact, and then either stops to look or, if there is time, walks past while rotating toward the item; an optional room-pan can be inserted between items when the room is spacious enough. The resulting path is finally segmented into labeled actions, normalized to room-relative coordinates, and downsampled to a fixed frame count.

Single-task segments. For single-task segments the goal is the opposite: produce one fixed-length clip that visits exactly one item under one of three task types (pass-and-look, stop-and-look, or room-span). Items are first filtered against a visibility rule list that removes labels too small or too structural to be useful targets, and only items with at least one valid vantage point are kept. For each candidate we plan both a direct route and a few detoured routes through adjacent rooms, and we evaluate how well the resulting walking time fits the task's frame budget (e.g. 162 frames for pass-and-look, 126 for stop-and-look, 114 plus a fixed pan budget for room-span). Candidates are then ranked by how closely they hit the target duration, with light bonuses for visiting a fresh room and for higher item importance, plus a penalty for repeating recently used rooms and randomness for tie-breaking. The chosen route is realized with the same clearance-aware pathfinder, retimed to land exactly on the target frame count, and downsampled to the final output length.

D.3. Single-task target and room composition.

Table 7 reports the ten most frequent item-target labels in the single-task dataset. These categories account for 60.0% of all item-target records, while the remaining 40.0% are spread across 147 labels. This distribution shows that the dataset has several frequent household categories, such as `chair`, `wardrobe`, and `curtain`, while still retaining a long tail of less frequent targets.

Table 9 reports the target-room composition for single-task records with resolved room labels. Bedrooms and living rooms form the largest share of target rooms, with bathrooms and kitchens appearing less often. The `room_span` category contains clips whose instruction names a room-level goal rather than an item-level target. Together, these statistics complement Table 1 by showing not only which task types appear, but also where those tasks occur in the floor plan.

D.4. Multi-task target and temporal composition.

Table 8 lists the most frequent item targets in the multi-task dataset, separated by `stop_and_look` and `pass_and_look` sub-tasks. The two sub-task types share common household categories, but their rankings differ. For example, `chair` is the most frequent `stop_and_look` target, while `wardrobe` is the most frequent `pass_and_look` target. This separation matters because the two sub-tasks require different behavior: stopping near a target versus passing it while maintaining a useful view.

Table 10 reports the duration of each multi-task sub-task type. `pass_and_look` segments are longest on average, `stop_and_look` segments are shortest, and `room_span` segments fall between them. This duration pattern reflects the structure of tour-like clips: the model must spend many frames moving through space, but only a shorter interval holding a target-facing view near an object.

Table 12. Door-crossing breakdown for cross-room records. A trajectory *crosses* a door when any consecutive xy frame segment intersects a door opening. **Clean**: crossed a door with no full wall collision. **Bumped**: crossed a door but had at least one full collision elsewhere. **No door, no coll.**: never crossed a door and never collided. **No door, coll.**: never crossed a door but collided, indicating wall-based room entry rather than doorway traversal.

Stream	Clean	Bumped	No door, no coll.	No door, coll.
Video	1020 (60.6%)	270 (16.0%)	226 (13.4%)	168 (10.0%)
Pose	339 (20.1%)	438 (26.0%)	332 (19.7%)	575 (34.1%)

Table 13. Distance to target for the single-task video+pose model.

			Inference dist (m)		GT dist (m)	
Modality	Path Type	n	Succ	Fail	Succ	Fail
Video	Single-room	462	1.76	2.48	1.70	1.67
	Cross-room	733	2.02	2.89	1.79	1.78
Pose	Single-room	462	2.10	3.11	2.28	1.68
	Cross-room	733	1.10	3.04	1.50	1.78

E. Additional Discussions

E.1. Alignment

The two streams are only moderately concordant (Pearson $r = 0.41$, Spearman $\rho = 0.43$, both $p \ll 0.001$), as shown in Figure 14. The video and pose streams are only moderately aligned, and their disagreement is asymmetric: the pose stream often under-reports translation that is visible in the video stream. This mismatch is concentrated in pass-and-look segments, where the agent should move through the scene while maintaining visual attention.

E.2. Door way traversal.

Cross-room paths produce many more full wall collisions than single-room paths. Here we further test whether these collisions reflect generic obstacle failure or a more specific doorway-traversal failure. We measure two properties: where collision frames occur relative to doorways, and whether cross-room trajectories actually intersect a doorway opening.

Tab. 11 reports the distance from each full-collision frame to the nearest doorway. For the video stream, cross-room collisions concentrate near doors: 79.3% of cross-room collision frames occur within 1 m of a doorway, compared with 40.2% for single-room paths. This gap suggests that many cross-room failures happen near the correct room-transition region rather than at arbitrary walls. The model often moves toward a doorway area, but it does not consistently pass through the opening without entering wall pixels. The pose stream shows a weaker but similar pattern: 59.1% of cross-room collision frames occur within 1 m of a doorway.

Tab. 12 gives a record-level view of the same failure. For cross-room records, 60.6% of video-stream paths cross a doorway without any full wall collision, while another 16.0% cross a doorway but still collide somewhere in the segment. However, 10.0% of video-stream records and 34.1% of pose-stream records never cross a doorway and still collide. These cases indicate a stronger structure error: the generated camera leaves the spawn room through a wall rather than through an annotated opening.

The model does not simply lack local wall perception. Instead, cross-room navigation exposes a gap between local spatial perception and room-level structure use. The video stream can often identify the door region, but it does not reliably treat the doorway as the valid passage between rooms. The pose stream shows a larger failure rate, especially in records that collide without ever crossing a doorway.

Table 14. Score engine quality measurement by outcome bucket for the single-task models, item-target records only. $\overline{\text{trans}}_{\text{eng}}$ and $\overline{\text{rot}}_{\text{eng}}$ are from the score engine: `distance_score` × `visible_score` and `look_score` × `visible_score` respectively. **Strict Success** = `segment_final_score` > 0.Due to there is no guranteed accurate way to extract the frustums roation from the video, thus for fairness, rotation entries for the video streams are dashed.

Model	Stream	Bucket	n	$\overline{\text{trans}}_{\text{eng}}$	$\overline{\text{rot}}_{\text{eng}}$	$\overline{\text{min dist}}$ (m)
GT (reference)	GT	Strict Success	1195	0.719	0.702	1.86
Video only	Video	Strict Success	783	0.579	–	2.20
		Wrong-Room Fail	317	0.022	–	3.52
		Correct-Room Fail	95	0.037	–	4.02
Video + Pose	Video	Strict Success	760	0.539	–	2.32
		Wrong-Room Fail	252	0.010	–	3.80
		Correct-Room Fail	183	0.005	–	3.69
	Pose	Strict Success	196	0.612	0.655	2.23
		Wrong-Room Fail	550	0.094	0.039	5.26
		Correct-Room Fail	449	0.352	0.180	2.60
Pose only	Pose	Strict Success	204	0.640	0.542	2.50
		Wrong-Room Fail	574	0.120	0.056	3.69
		Correct-Room Fail	417	0.436	0.156	2.74

E.3. Distance to the target.

Tab. 13 shows that successful video-stream generations end closer to the target than failed generations. For the video stream, the inference distance rises from 1.76 m to 2.48 m on single-room paths and from 2.02 m to 2.89 m on cross-room paths. The corresponding ground-truth distances remain nearly unchanged across success and failure buckets. This indicates that the success–failure split is not caused by failed records having systematically farther targets. Instead, the generated path itself fails to move close enough to the target.

The pose stream shows a related but different pattern. Failed pose-stream generations remain farther from the target than successful ones: 3.11 m versus 2.10 m on single-room paths and 3.04 m versus 1.10 m on cross-room paths. However, the component scores in Tab. 14 show that pose failures are not only distance failures. Some pose outputs still place the camera in a plausible local region but fail to orient it toward the target.

E.4. Translation and rotation components.

Tab. 14 explains this difference more directly. For the video streams, correct-room failures have near-zero translation scores: 0.037 for the video-only model and 0.005 for the video stream of the video+pose model, compared with the GT reference value of 0.719. These records reach the correct room but do not localize the target well enough to produce a useful camera position.

The pose streams fail in a different way. For correct-room failures, the pose-only model keeps a translation score of 0.436, and the pose stream of the video+pose model reaches 0.352. These values are much higher than the video-stream correct-room failure scores, which suggests that pose outputs can move closer to the target once they enter the correct room. However, their rotation scores remain low: 0.156 for the pose-only model and 0.180 for the pose stream of the video+pose model, compared with the GT reference value of 0.702. Thus, the pose stream often reaches a more plausible local position but does not reliably face the target.

Together, these diagnostics support the main-text claim that correct-room arrival does not imply successful target grounding. The model must both select the correct room and solve a local camera-placement problem inside that room. Current models often satisfy only part of this requirement: video streams struggle mainly with local target position, while pose streams retain more translation signal but still fail target-facing orientation.

E.5. Target feasibility.

Figure 15 tests which target labels appear more or less often in the *Strict Success* bucket than in the ground-truth (GT) distribution. For each stream, we compute

$$\text{ratio} = \frac{\text{share in Strict Success}}{\text{share in GT}}.$$

Ratios above $1\times$ indicate labels that appear more often in successful outputs than their GT frequency predicts. Ratios below $1\times$ indicate labels that appear less often.

The under-represented labels show that target feasibility matters. Small or low-saliency items, such as `body pillow` and `stool`, appear on the under-represented side for weaker streams. This pattern is expected in a floor-plan-conditioned task. Small objects occupy few pixels on the floor plan, provide weaker spatial cues, and can be harder to ground than large room-defining furniture.

However, target size does not explain the full pattern. Large and visually clear categories, such as `wardrobe`, `side table`, `floor lamp`, `chair`, and `TV`, also appear as under-represented labels in some streams. These objects should be easier to locate than small items, but their success ratios still vary across model outputs.

The cross-stream comparison makes this point clearer. Some labels that are under-represented for the pose streams become over-represented for the V+P / Video stream. For example, `side table`, `chair`, and `wardrobe` move into the success-heavy side for V+P / Video. Thus, the same target category can be easy for one stream and hard for another stream on the same dataset.

Overall, Fig. 15 supports a mixed interpretation. Small targets create a real grounding challenge because they are less visible on the floor plan. At the same time, the inconsistent behavior on large targets shows that target feasibility alone cannot explain the failures. The remaining gap reflects stream-level differences in spatial grounding and planning.

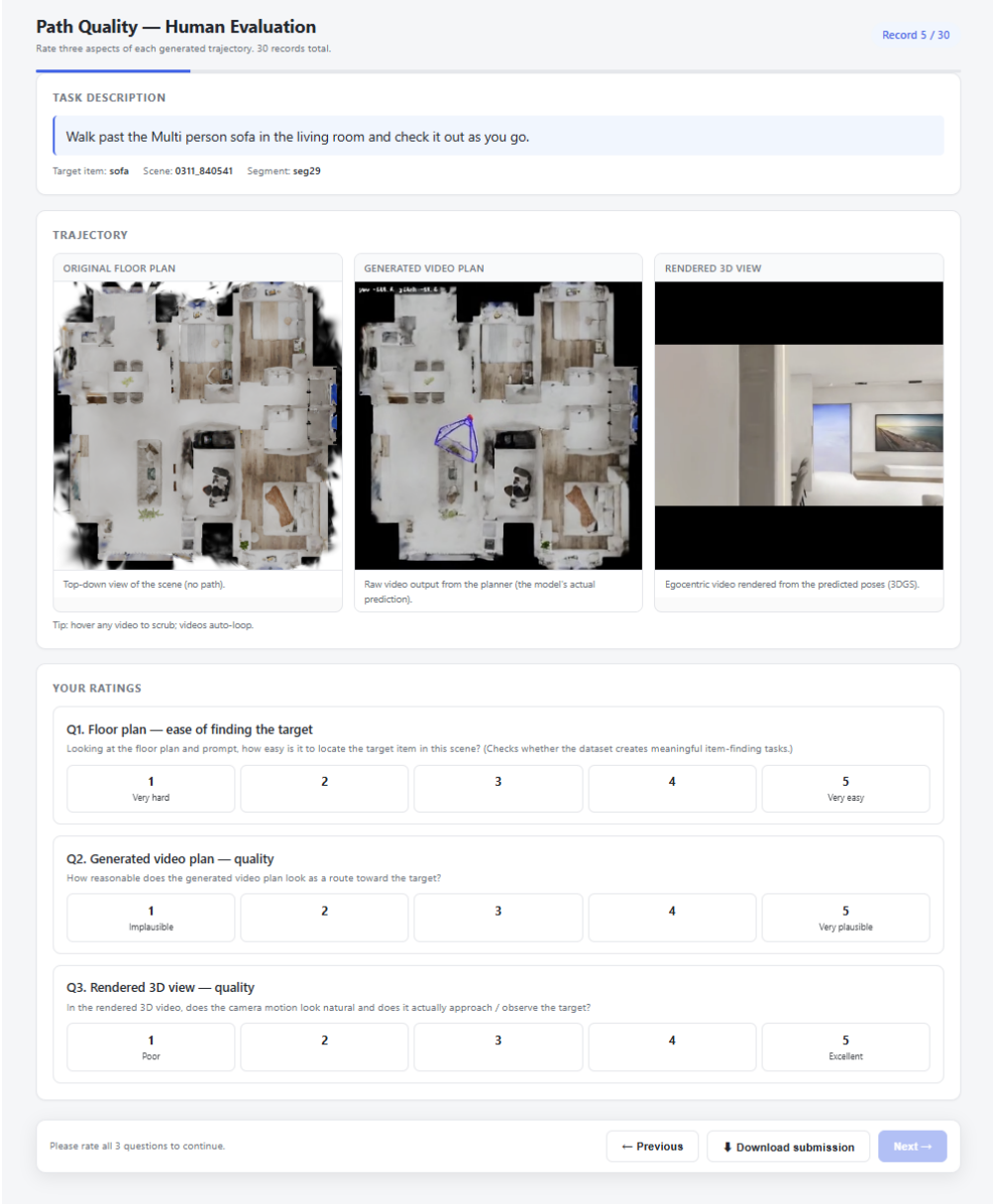


Figure 10. Human-evaluation interface. Volunteers rate target findability from the floor plan and prompt, 2D path quality from the floor-plan video, and rendered-video quality from the 3D scene.

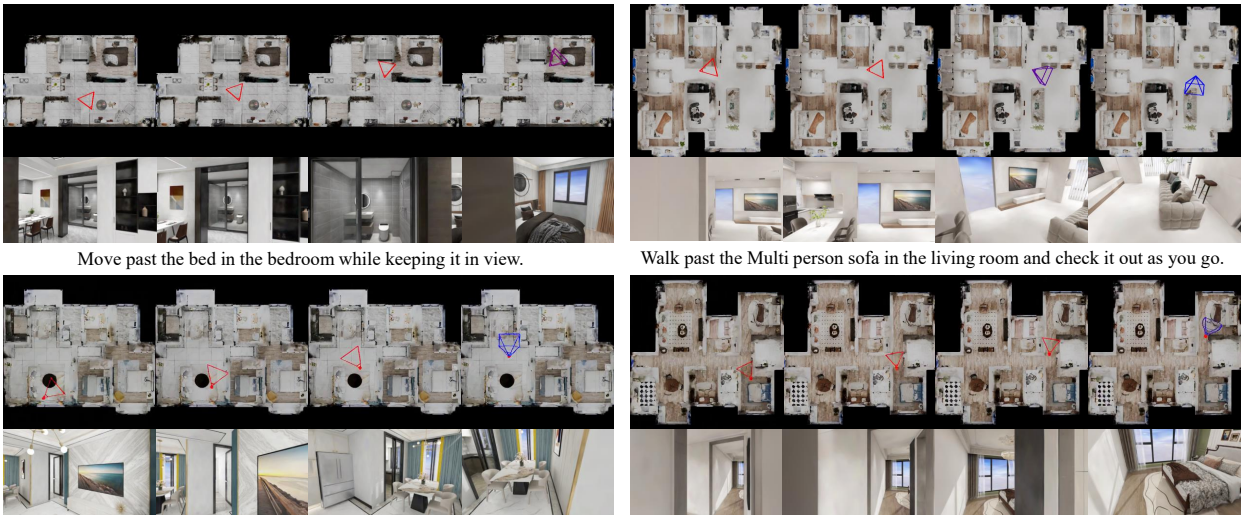


Figure 11. Qualitative results for four generated paths from the video and pose streams. Video-stream examples show the directly generated floor-plan videos. Pose-stream examples show paths reconstructed from the predicted pose sequences.

Figure 12. Four success case from video stream.

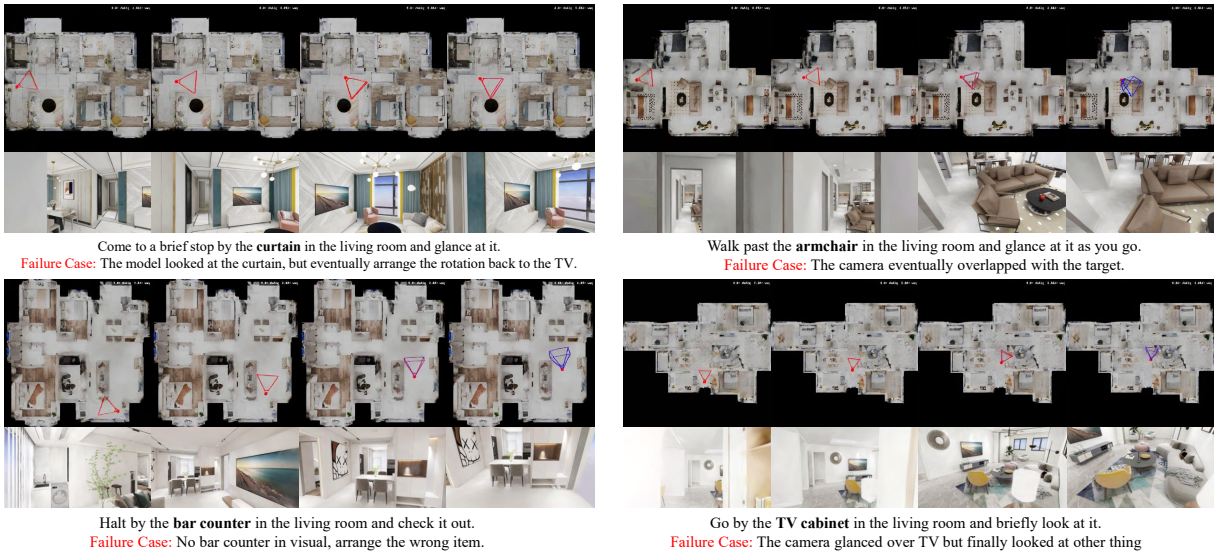


Figure 13. Four different failure cases.

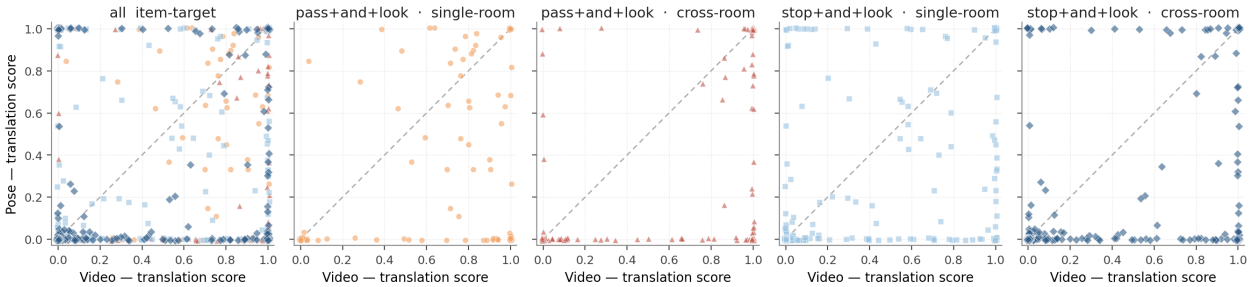


Figure 14. Video–pose translation alignment: the pooled scatter (left) and four (kind, path-type) cells share identical axes; points below the dashed $y = x$ line indicate video translation that the pose stream fails to capture.

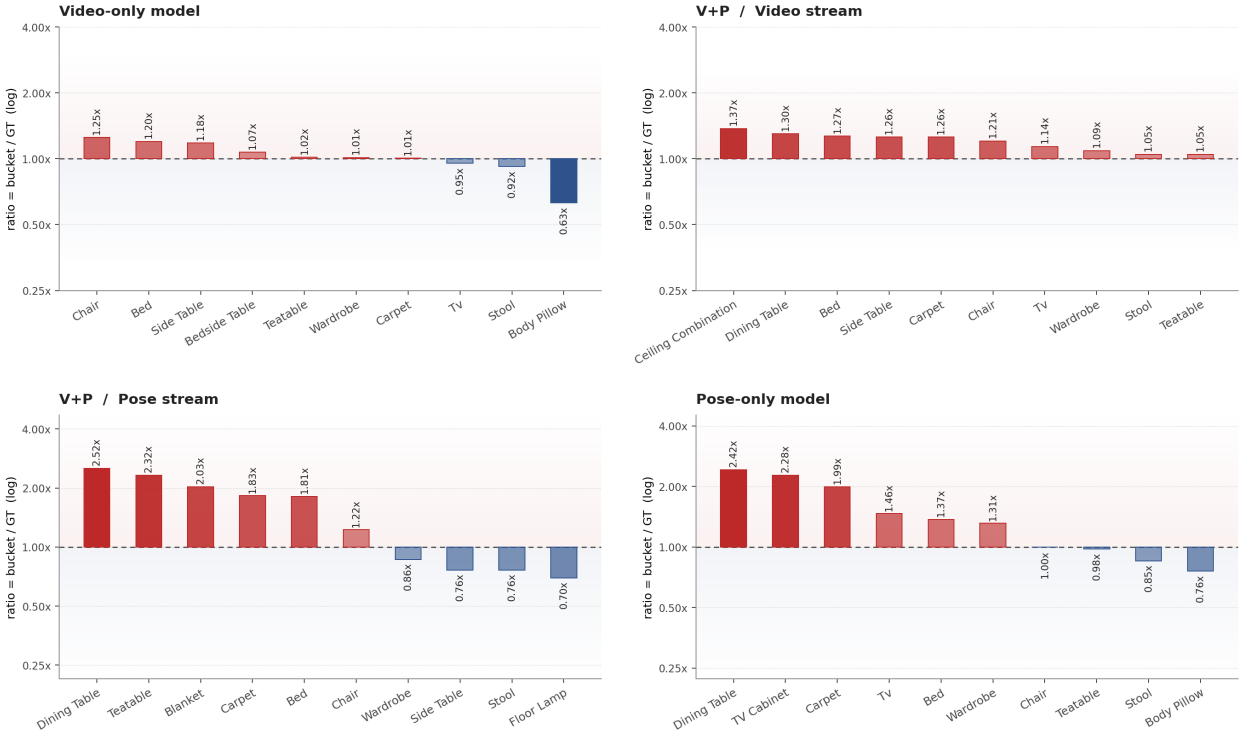


Figure 15. Target-label over- and under-representation in the *Strict Success* bucket. Each panel shows the top-10 target labels for one stream, with ratio = Strict-Success share/GT share on a $\log_2$ axis. Red bars mark over-represented labels, and blue bars mark under-represented labels.