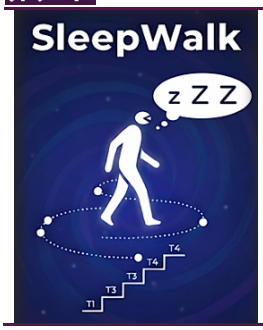


Sleep Walk: A Three-Tier Benchmark for Stress-Testing Instruction-Guided Vision-Language Navigation

Niyati Rawal[†⋆], **Sushant Ravva**[⋆], **Shah Alam Abir**[¶], **Saksham Jain**[*], **Aman Chadha**[‡1], **Vinija Jain**[§2], **Suranjana Trivedy**[⋆], **Amitava Das**[⋆]

[†] Indian AI Research Organization (IAIRO), India,

[⋆] ɹragya Lab, BITS Pilani Goa, India,

[¶] University of Dhaka, Bangladesh,

[*] Delhi Technological University, India,

[‡] Google DeepMind, USA,

[§] Google, USA

Abstract

Vision-Language Models (VLMs) have advanced rapidly in multimodal perception and language understanding, yet it remains unclear whether they can reliably ground language into spatially coherent, plausibly executable actions in 3D digital environments. We introduce ***SleepWalk***, a benchmark for evaluating ***instruction-grounded trajectory prediction*** in ***single-scene 3D worlds*** generated from textual scene descriptions and filtered for navigability. Unlike prior navigation benchmarks centered on long-range exploration across rooms, SleepWalk targets ***localized, interaction-centric embodied reasoning***: given rendered visual observations and a natural-language instruction, a model must predict a trajectory that respects scene geometry, avoids collisions, and terminates at an action-compatible location. The benchmark covers diverse indoor and outdoor environments and organizes tasks into ***three tiers of spatial and temporal difficulty***, enabling fine-grained analysis of grounding under increasing compositional complexity. Using a standardized ***pointwise judge-based evaluation protocol***, we evaluate three frontier VLMs on ***2,472 curated 3D environments*** with ***nine instructions per scene***. Results reveal ***systematic failures in grounded spatial reasoning***, especially under occlusion, interaction constraints, and multi-step instructions: performance drops as the difficulty level of the tasks increase. In general, current VLMs can somewhat produce trajectories that are simultaneously spatially coherent, plausibly executable, and aligned with intended actions. By exposing failures in a controlled yet scalable setting, SleepWalk provides a critical benchmark for advancing grounded multimodal reasoning, embodied planning, vision-language navigation, and action-capable agents in 3D environments. ***Repository***

1 Introduction

The next frontier for Vision-Language Models (VLMs) is not merely to describe the world, but to act in it with spatial precision. Recent progress in multimodal learning has substantially expanded the capabilities of VLMs in image captioning, visual question answering, and instruction following (Vinyals et al., 2015; Antol et al., 2015; Anderson et al., 2018). Yet as these models are increasingly positioned as the reasoning core of embodied agents, robotic assistants, and action-capable foundation models, a more consequential question comes into focus: ***can they reliably translate natural-language instructions into spatially grounded, executable behavior in 3D environments?***

This problem is important because embodied competence demands far more than semantic recognition. An agent must localize itself, infer reachable goals, reason about geometry and oc-

[1] Work done outside Google DeepMind, USA

[2] Work done outside Google, USA

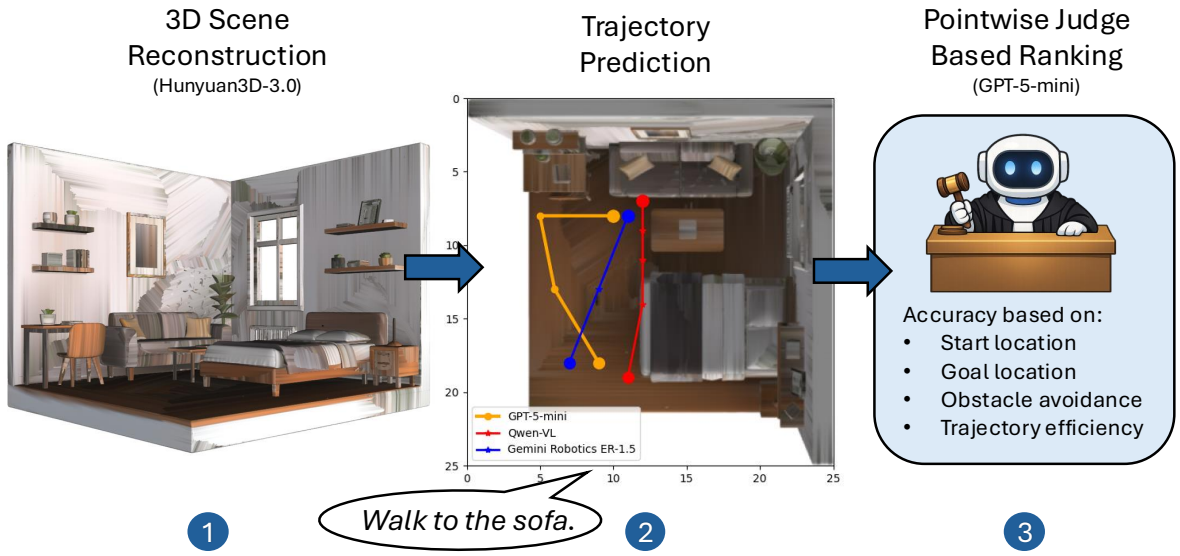


Figure 1: *Overview of SleepWalk.* A language instruction is converted into a single-scene 3D environment using Hunyuan3D-3.0. For each scene, given language instructions, different VLMs predict trajectories, which are visualized using top-down views. A fixed judge model scores trajectories in a pointwise manner, and rankings are aggregated across environments.

clusion, respect environmental constraints, and generate action sequences that remain feasible for downstream interaction. In other words, success depends on whether a model can connect ***language, space, and action*** rather than merely align text with pixels. Recent work has shown that even strong VLMs continue to struggle with embodied spatial understanding, top-view reasoning, fine-grained navigation competence, multi-view robotic reasoning, and precise embodied grounding (Du et al., 2024; Li et al., 2024; Wang et al., 2024; Feng et al., 2025; Xue et al., 2025). These limitations matter directly for future systems that must plan, navigate, manipulate, and interact safely in the physical world.

A particularly important setting is ***3D scene navigation under natural-language instructions***. Classical Vision-and-Language Navigation (VLN) benchmarks have played a foundational role in embodied AI by studying how agents follow instructions in simulated environments (Anderson et al., 2018). However, much of this literature emphasizes long-horizon movement across rooms or buildings, where evaluation is often dominated by endpoint success or goal reachability. While such benchmarks remain essential, they provide only partial visibility into a harder and increasingly practical capability: ***localized, interaction-centric reasoning within a single scene.*** In many real deployments, an embodied agent must do more than reach a destination. It must approach an object from a feasible direction, stop at an interaction-compatible location, avoid clutter and collisions, and preserve the geometric conditions necessary for the requested action.

Recent benchmarks have begun to expose this broader gap. EmbSpatial-Bench highlights weaknesses in embodied spatial understanding from egocentric 3D scenes (Du et al., 2024); TopViewRS studies top-view reasoning relevant to localization and map-based navigation (Li et al., 2024); Navigating the Nuances shows that standard VLN evaluation can obscure systematic deficits in directional, landmark, and numerical instruction following (Wang et al., 2024); and newer embodied benchmarks such as MV-RoboBench and Point-It-Out reveal persistent failures in multi-view robotic reasoning and staged visual grounding (Feng et al., 2025; Xue et al., 2025). ***What remains missing, however, is a benchmark that directly tests whether a model can produce a continuous, spatially coherent, executable trajectory inside a single 3D scene under object interaction and motion constraints.***

To address this gap, we introduce ***SleepWalk***, a benchmark for evaluating ***instruction-grounded trajectory prediction*** in ***single-scene 3D environments*** reconstructed from language descriptions (Fig. 1). Each environment corresponds to one coherent indoor or outdoor scene, deliberately avoiding room-to-room exploration in favor of ***fine-grained spatial understand-***

ing, object-centric interaction, and trajectory feasibility. Given rendered visual observations and a natural-language instruction, a model must infer a continuous path that is consistent with scene layout, avoids collisions, and terminates at a location compatible with the intended action, such as approaching an object, picking it up, or sitting on furniture. Unlike symbolic planning or endpoint-only evaluation, SleepWalk assesses the ***full spatial and temporal coherence*** of the predicted path with respect to both the environment and the instruction.

SleepWalk comprises ***2,472 curated 3D environments*** spanning diverse layouts, object configurations, and levels of clutter. For each scene, we generate ***nine instructions*** across ***three tiers of difficulty***, enabling controlled analysis of embodied reasoning under increasing compositional and interaction complexity. We further visualize selected trajectories with MotionGPT and TLControl to inspect whether predicted paths remain compatible with humanoid execution (Jiang et al., 2024; Wan et al., 2024). To compare heterogeneous model outputs, we introduce a standardized judge-based evaluation protocol that scores trajectories by start-location consistency, goal satisfaction, obstacle avoidance, and trajectory efficiency. Across frontier VLMs, we observe substantial degradation under ***occlusion, interaction constraints, and multi-step instructions***, even when the same models perform reasonably on simple goal-reaching prompts.

The broader significance of SleepWalk is that it operationalizes a critical missing layer between seeing and acting. It is not only a benchmark for navigation, but a diagnostic testbed for future research on ***grounded multimodal reasoning, embodied planning, robot instruction following, action-aware world modeling, and spatially reliable multimodal agents.*** As VLMs move from passive perception toward embodied deployment, the ability to evaluate this intermediate layer—where language must be converted into feasible behavior—will become increasingly central to both capability and safety.

The contributions of this paper are threefold:

- We introduce ***SleepWalk***, a new benchmark that constructs spatially consistent ***single-scene 3D environments*** from language descriptions using Hunyuan3D-3.0, and generates diverse trajectory-following instructions using Qwen3-8B-VL, yielding ***nine instructions per environment*** for multi-level embodied evaluation.

- We benchmark frontier ***Vision-Language Models*** on ***continuous coordinate-based trajectory prediction*** from visual observations and natural-language instructions, emphasizing ***spatial feasibility, temporal coherence, and interaction compatibility*** across easy, medium, and hard tasks.

- We introduce a standardized ***judge-based evaluation protocol*** for comparing heterogeneous model outputs, and use it to reveal persistent limitations in ***grounded spatial reasoning***, especially in ***occluded, compositional, and interaction-heavy*** scenarios.

2 SleepWalk: A Three-Tier Benchmark for Grounded Trajectory Reasoning

SleepWalk is a benchmark for testing whether Vision-Language Models can convert language into spatially grounded, executable behavior in 3D environments. As illustrated in Fig. 2, the benchmark begins from natural-language scene descriptions, reconstructs navigable single-scene 3D worlds, generates tiered navigation instructions, and evaluates predicted trajectories under a standardized protocol. The design of SleepWalk deliberately emphasizes ***localized, interaction-centric reasoning*** rather than long-range exploration: a model must understand scene geometry, identify feasible routes, avoid obstacles, and stop at a location compatible with the requested action. In this way, SleepWalk probes a critical capability that lies between ***seeing*** and ***acting***. This focus is motivated by a growing body of recent evidence showing that current VLMs still struggle with embodied spatial grounding, top-view reasoning, and fine-grained navigation competence, even when they perform strongly on broader multimodal tasks (Du et al., 2024; Li et al., 2024; Wang et al., 2024; Mayer et al., 2025).

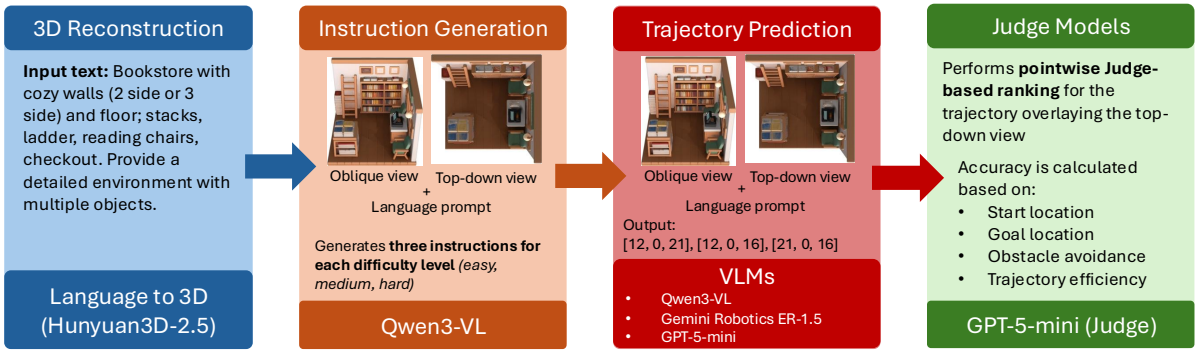


Figure 2: ***SleepWalk pipeline.*** Starting from a natural-language scene description, we reconstruct a single-scene 3D environment with Hunyuan3D-3.0, render top-down and oblique observations, and use Qwen3-8B-VL to generate tiered navigation instructions (*easy*, *medium*, *hard*). Given the scene views and an instruction, a VLM predicts a continuous action trajectory, which is then evaluated by a judge model (GPT-5-mini) using pointwise scoring over *start-location consistency*, *goal satisfaction*, *obstacle avoidance*, and *trajectory efficiency*.

2.1 Text-to-3D Environment Reconstruction

Each SleepWalk instance begins with a natural-language scene description. We sample MS-COCO captions and manually filter or rewrite them to obtain $1{,}200$ descriptions suitable for ***navigable single-scene 3D generation***, spanning both indoor and outdoor settings. We then use Hunyuan3D-3.0 (Cao et al., 2025) to convert each description into a spatially coherent 3D environment by estimating scene layout, object placement, and overall geometry.

The resulting environments are intentionally restricted to ***single coherent scenes*** rather than multi-room worlds. This keeps the benchmark focused on the regime of greatest interest: fine-grained movement, local planning, and action feasibility around objects, furniture, and clutter. By combining indoor and outdoor scenes, SleepWalk captures a broader range of layouts, densities, and navigation constraints, thereby increasing both environmental diversity and reasoning difficulty. This single-scene emphasis complements prior benchmarks that evaluate embodied spatial understanding from egocentric views (Du et al., 2024) or top-view map reasoning (Li et al., 2024), while moving closer to continuous action prediction in executable 3D scenes.

2.2 Instruction Generation

For each reconstructed scene, we generate navigation instructions using Qwen3-8B-VL (Yang et al., 2025). The model is given both ***top-down*** and ***oblique*** views of the environment and is prompted to produce ***nine goal-directed instructions*** organized into three difficulty tiers: *easy*, *medium*, and *hard*.

The generated instructions are designed to remain ***scene-grounded, concise, and executable***. They target behaviors such as approaching an object, moving between landmarks, manipulating an item, or reaching a location compatible with sitting or interaction. Difficulty increases with compositional and spatial demands: easy instructions typically require short-range goal localization, medium instructions involve structured spatial dependencies, and hard instructions introduce multi-step goals, interaction constraints, or longer planning horizons. This three-tier design enables controlled analysis of how model performance degrades as embodied reasoning becomes more demanding, in the same spirit as recent fine-grained evaluations that diagnose failure modes beyond coarse task success (Wang et al., 2024; Mayer et al., 2025).

2.3 Action-Conditioned Trajectory Prediction

Given a reconstructed 3D environment $\mathcal{E}$ and a natural-language instruction $\mathcal{I}$, the task is to predict a continuous trajectory

$$\mathcal{T} = \{p_t\}_{t=1}^{T}, \qquad p_t \in \mathbb{R}^3,$$

where p_t denotes the agent position at time step t. A valid trajectory must satisfy three conditions: it must be ***spatially feasible*** within $\mathcal{E}$, it must ***avoid collisions*** with scene elements, and it must terminate at a location that supports execution of the intended action described by $\mathcal{I}$.

The model receives rendered visual observations $\mathcal{V}$ from the environment together with the instruction, and predicts

$$\mathcal{T} = f_\theta(\mathcal{V}, \mathcal{I}).$$

All evaluated VLMs are tested in a ***frozen, zero-shot setting***: model parameters θ remain fixed, and no task-specific fine-tuning or adaptation is performed on SleepWalk.

This formulation is intentionally stricter than endpoint-based navigation. A model is not rewarded merely for ending near the correct goal; instead, it must produce a trajectory whose ***entire path*** is consistent with scene geometry, object affordances, and temporal ordering. SleepWalk therefore evaluates whether a VLM can ground language into a ***full sequence of spatially and temporally coherent actions***, rather than only a final destination. Relative to recent benchmarks that study embodied spatial understanding (Du et al., 2024), top-view reasoning (Li et al., 2024), or fine-grained VLN diagnostics (Wang et al., 2024), SleepWalk directly targets the missing intermediate layer between instruction interpretation and executable path generation.

2.4 Judge-Based Evaluation Protocol

To assess whether predicted trajectories are grounded, executable, and instruction-consistent, we introduce a standardized ***judge-based evaluation protocol***. Rather than relying only on heuristic metrics such as distance to goal, we use a strong vision-language model as a structured evaluator that jointly considers the instruction, the scene, and the predicted path.

For each candidate trajectory, the judge is given: (i) the ***top-down view*** of the reconstructed 3D environment, (ii) the ***natural-language instruction***, and (iii) the ***trajectory overlay*** **rendered on the map**.

Each trajectory is represented as an ordered sequence of 3D waypoints projected onto the 2D top-down plane. The judge is prompted to evaluate whether the trajectory aligns with the instruction while respecting scene geometry and environmental constraints.

We use a ***pointwise scoring scheme*** in which each trajectory is evaluated independently. The judge assigns a score for four factors:

- ***Start Location Consistency***: Does the trajectory begin at the correct initial region?
- ***Goal Satisfaction***: Does the trajectory end at a location that satisfies the instruction?
- ***Obstacle Avoidance***: Does the path avoid collisions and remain geometrically plausible?
- ***Trajectory Efficiency***: Is the route reasonably direct, without unnecessary detours?

Each factor is explicitly defined in the evaluation prompt to reduce ambiguity and improve reproducibility. For a trajectory τ, the judge assigns a discrete score

$$s_k(\tau) \in \{1, 2, 3, 4, 5\} \cup \{\mathrm{N/A}\}, \qquad k \in \{\mathsf{start}, \mathsf{goal}, \mathsf{obs}, \mathsf{eff}\}.$$

Valid scores are normalized to the interval $[0, 1]$:

$$\tilde{s}_k(\tau) = \frac{s_k(\tau)}{5}.$$

Let $\mathcal{T}_{t,k}^{\mathrm{valid}}$ denote the set of trajectories in difficulty tier t with valid (non-N/A) scores for factor k. The tier-level factor score is then

$$S_k(t) = \frac{1}{|\mathcal{T}_{t,k}^{\mathrm{valid}}|} \sum_{\tau \in \mathcal{T}_{t,k}^{\mathrm{valid}}} \tilde{s}_k(\tau).$$

To summarize performance across difficulty tiers, we compute the overall factor score

$$S_k = \frac{1}{|\mathcal{T}_k^{\mathrm{tiers}}|} \sum_{t \in \mathcal{T}_k^{\mathrm{tiers}}} S_k(t),$$

where $\mathcal{T}_k^{\text{tiers}} \subseteq \{\text{easy}, \text{medium}, \text{hard}\}$ denotes the set of tiers with valid evaluations for factor k. By default, all tiers are weighted equally.

This protocol allows heterogeneous trajectory outputs to be compared under a shared notion of grounded success. Crucially, it evaluates not only where a model ends up, but whether it reaches that endpoint through a path that is spatially plausible and action-compatible. This is especially important in light of recent work showing that coarse task-level metrics can mask systematic spatial and navigational failures (Wang et al., 2024; Mayer et al., 2025).

2.5 Humanoid Visualization and Embodied Execution

To assess whether predicted trajectories are compatible with embodied execution, we visualize selected outputs using a humanoid pipeline based on TLControl (Wan et al., 2024) and MotionGPT (Jiang et al., 2024). TLControl converts waypoint sequences into control signals, while MotionGPT generates realistic full-body motions such as walking and object interaction.

This stage is used for ***qualitative validation*** rather than primary scoring. Its purpose is to reveal whether trajectories that appear reasonable in top-down space remain plausible when executed by an embodied humanoid. In practice, this visualization helps expose failures that may be less obvious in static overlays, such as collisions with nearby objects, awkward stopping positions, or motion patterns incompatible with the intended interaction.

Overall, SleepWalk provides a controlled yet scalable benchmark for grounded trajectory reasoning in Vision-Language Models, combining 3D reconstruction, tiered instructions, trajectory prediction, structured evaluation, and embodied visualization to study the translation of language into ***feasible spatial behavior***.

3 Evaluating Grounded Trajectory Reasoning in 3D Scenes

We now ask a central empirical question: can current Vision-Language Models translate instructions into trajectories that are spatially grounded, physically feasible, and compatible with the intended action in 3D scenes? To answer this, we evaluate three representative frontier VLMs—***Qwen3-VL***, ***Gemini Robotics ER-1.5***, and ***GPT-5-mini***—on SleepWalk.

Because SleepWalk is designed as an evaluation benchmark rather than a training resource, all experiments are conducted in a strictly ***evaluation-only, zero-shot setting***. No model is fine-tuned or adapted to the benchmark. For each scene–instruction pair, each model produces a single trajectory under ***deterministic decoding***, ensuring that comparisons remain reproducible and attributable to capability rather than sampling variance. Each predicted trajectory is rendered using a standardized visual format consisting of one top-down view and one oblique scene view, preserving both geometric structure and task context.

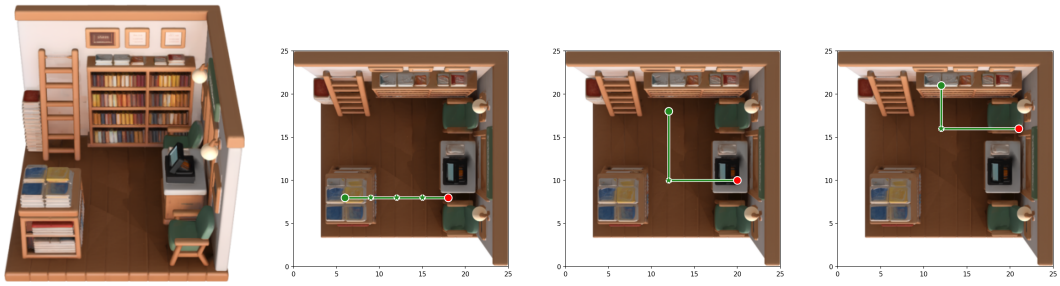
We use ***GPT-5-mini*** as the judge model with a fixed evaluation prompt throughout. The judge scores each predicted trajectory independently using the pointwise evaluation protocol defined in Section 2. All evaluated models are tested under identical conditions—the same reconstructed scenes, the same instructions, the same trajectory representation, and the same judge configuration—so that observed differences reflect ***differences in grounded trajectory reasoning rather than experimental mismatch***.

3.1 Qualitative Results

Figure 3 shows a clear pattern: as task complexity increases, failures shift from coarse recognition to grounded execution. We analyze one example from each difficulty tier to illustrate how different VLMs translate instructions into trajectories.

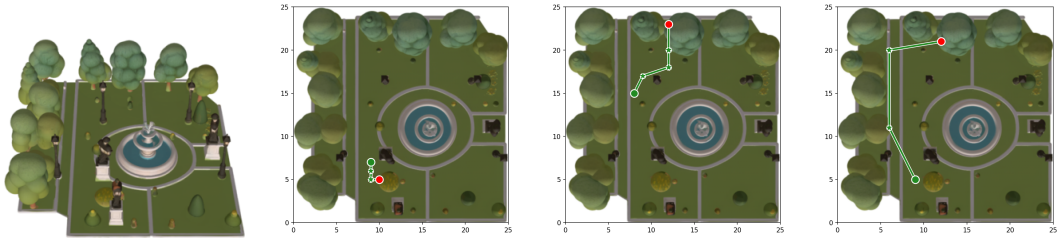
For the ***easy*** example, *"Walk from the bookshelf to the wall-mounted lamp,"* all three models recover the broad task semantics: they localize the bookshelf region and produce plausible routes toward a lamp. GPT-5-mini selects the more plausible target lamp, whereas Gemini Robotics ER-1.5 ends closer to the farther lamp. However, ***none of the models fully satisfies local physical constraints***: parts of their trajectories lie on or too close to the source or target object, which would induce collision during execution. Even simple tasks therefore reveal a gap between ***semantic grounding*** and ***physically feasible motion***.

The ***medium*** example, *"Approach the yellow spherical object and then move to the northern tree,"* requires resolving two grounded references while preserving temporal order. Qwen3-VL



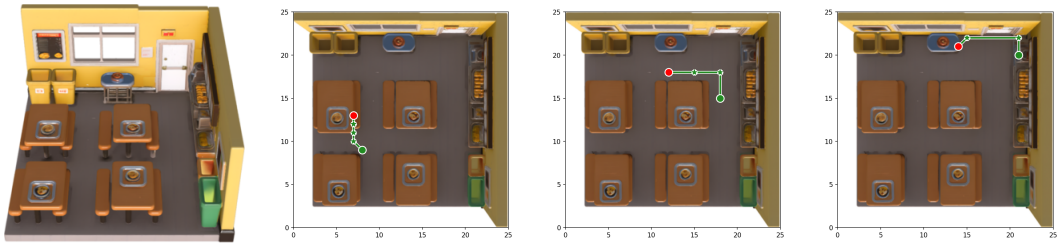
(a) (b) (c) (d)

LEVEL 1 | TASK: Walk from the bookshelf to the wall-mounted lamp | START: Near the bookshelf | END: Near the wall-mounted lamp | INTERACT: none



(a) (b) (c) (d)

LEVEL 2 | TASK: Approach the yellow spherical object and then move to the northern tree | START: Near the yellow spherical object | END: Near the northern tree | INTERACT: yellow spherical object, tree



(a) (b) (c) (d)

LEVEL 3 | TASK: Pick up the tray from the service counter, walk to the round table, and place it there | START: Near the service counter | END: Near the round table | INTERACT: tray, service counter, round table

Figure 3: ***Qualitative trajectory comparison across difficulty levels.*** Column (a) shows the reconstructed scene view, while columns (b), (c), and (d) show trajectory predictions from Qwen3-VL, Gemini Robotics ER-1.5, and GPT-5-mini, respectively. Across levels, the figure highlights differences in start-location grounding, goal accuracy, obstacle avoidance, and robustness under compositional instructions.

fails on both ends, while Gemini Robotics ER-1.5 identifies the final tree more accurately but misses the correct start region. GPT-5-mini is the only model that captures both. Here, the dominant failure is not collision, but ***compositional grounding***: models struggle to bind intermediate references, preserve order, and assign correct spatial roles.

The ***hard*** example, *"Pick up the tray from the service counter, walk to the round table, and place it there,"* introduces explicit interaction and multi-step planning. GPT-5-mini again produces the strongest trajectory, correctly aligning both start and goal while maintaining a plausible route. ***Across tiers, the trend is consistent: as instructions become more interaction-heavy and temporally structured, errors compound rapidly.***

Taken together, these examples expose three recurring failure modes in SleepWalk: ***mislocalized starts, incomplete or incorrect goal grounding, and trajectories that appear semantically plausible yet remain physically unsafe.*** The qualitative evidence therefore supports the main claim of this benchmark: ***current VLMs often understand what an instruction refers to, but still struggle to convert that understanding into spatially coherent, executable behavior in 3D scenes.***

Table 1: ***Model Comparison on SleepWalk.*** Higher scores indicate better performance.

Model	Start loc.	Goal loc.	Avoid obs.	Traj. eff.
Qwen3-VL	0.48	0.20	0.84	0.47
Gemini Robotics ER-1.5	0.58	0.34	0.89	0.58
GPT-5-mini	0.75	0.51	0.91	0.64

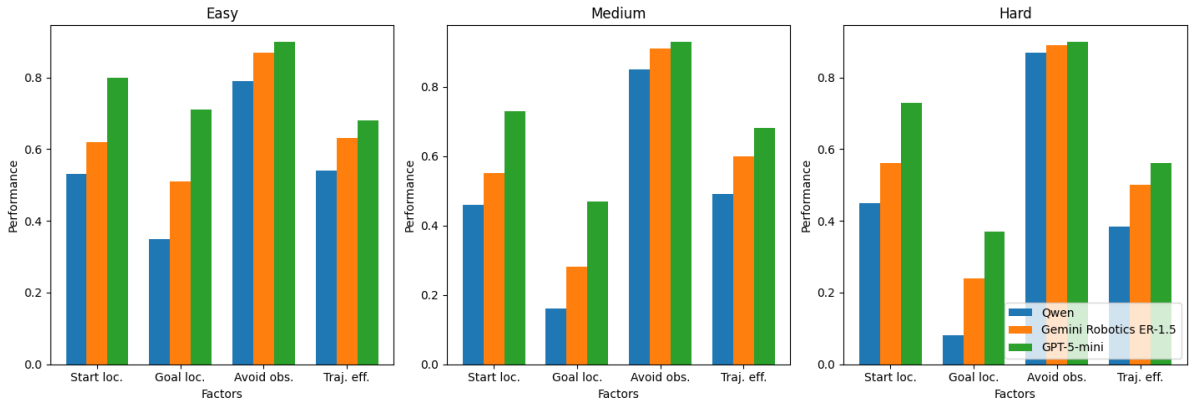


Figure 4: ***Factor-wise performance across difficulty tiers.*** Comparison of Qwen3-VL, Gemini Robotics ER-1.5, and GPT-5-mini on four evaluation factors—start location, goal location, obstacle avoidance, and trajectory efficiency—across easy, medium, and hard tiers.

3.2 Overall Model Ranking

Table 1 reports overall factor scores for Qwen3-VL, Gemini Robotics ER-1.5, and GPT-5-mini. Scores lie in $[0, 1]$, with higher values indicating better performance. ***Among the models with complete quantitative results, GPT-5-mini performs best on all four factors***: start-location consistency, goal satisfaction, obstacle avoidance, and trajectory efficiency.

The pattern is also diagnostic. Both models score relatively high on obstacle avoidance, but much lower on goal grounding. ***This suggests that producing a navigable-looking path is easier than reaching the correct interaction-compatible endpoint.*** Thus, the main bottleneck is not coarse geometry, but ***precise instruction grounding.***

3.3 Ranking by Difficulty Level

Figure 4 breaks down performance by difficulty tier across the four evaluation factors. ***The trend is clear: performance drops from easy to medium to hard tasks.*** This indicates that SleepWalk successfully exposes increasing demands on grounded spatial reasoning.

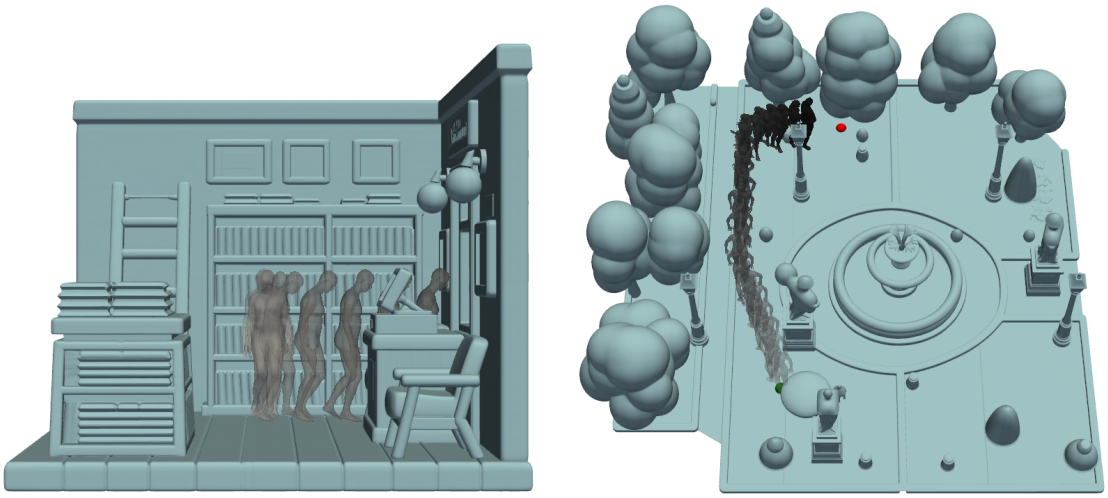
Among the quantitatively reported models, ***GPT-5-mini performs best across factors and tiers***, while Qwen3-VL degrades more sharply, especially on goal grounding and trajectory efficiency. The steepest drop occurs on harder tasks requiring multi-step reasoning, ordered execution, and interaction-aware endpoint selection.

Taken together, the quantitative results reinforce the qualitative findings: current VLMs retain competence on simple goal-reaching behavior, but deteriorate sharply once tasks require ***compositional grounding, interaction awareness, and temporally structured planning***.

3.4 From Predicted Paths to Embodied Execution

To move beyond static trajectory overlays, we perform a final ***embodied executability check*** using a humanoid control and animation pipeline. Specifically, we take the waypoint sequence predicted by GPT-5-mini, project it back into the reconstructed 3D environment, and render it as humanoid motion. TLControl (Wan et al., 2024) converts the trajectory into low-level control signals, while MotionGPT (Jiang et al., 2024) synthesizes realistic full-body movement conditioned on the predicted path.

Figure 5 shows two such examples for the instruction *"Walk from the bookshelf to the wall-mounted lamp."* and *"Approach the yellow spherical object and then move to the northern tree"*.



(a) (b)

Figure 5: ***From predicted path to humanoid execution.*** We animate the trajectory generated by GPT-5-mini for the instruction *"Walk from the bookshelf to the wall-mounted lamp"* and *"Approach the yellow spherical object and then move to the northern tree"* using TLControl (Wan et al., 2024) and MotionGPT (Jiang et al., 2024). This provides a qualitative check of whether a path that appears correct in top-down space remains plausible when executed by a humanoid agent.

The resulting animation allows us to inspect whether trajectories that appears reasonable in top-down space remains ***physically plausible*** when executed by an embodied agent. This matters because geometrically valid paths can still fail at execution time due to collisions, awkward stopping positions, or unnatural motion transitions.

This stage provides a qualitative bridge between geometric correctness and embodied feasibility. It does not replace the benchmark's primary judge-based evaluation, but serves as an additional validation layer for whether predicted paths can support realistic action in 3D environments. In practice, it makes failure modes more visible, especially in cases where a trajectory is semantically plausible yet operationally unsafe or physically incompatible with the intended interaction.

4 Conclusion and Future Avenues

SleepWalk benchmarks whether VLMs can turn language into grounded, executable trajectories in 3D scenes. By focusing on ***localized, interaction-centric reasoning***, it isolates a key gap between perception and action. Across models, we find that current VLMs still struggle with ***spatial grounding, compositional instructions, and executable path generation***, especially in occluded and multi-step settings.

Promising next steps include ***(i)*** richer multi-view and temporal observations, ***(ii)*** direct reasoning over structured 3D scene representations, ***(iii)*** tighter coupling between predicted paths and downstream motion or control models, and ***(iv)*** extending the benchmark into physics simulators for embodied training and sim-to-real transfer.

Overall, SleepWalk provides a scalable testbed for advancing grounded multimodal reasoning, embodied planning, and action-capable vision-language systems.

References

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. 2025. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*.

Mengfei Du, Binhao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. 2024. EmbSpatial-Bench: Benchmarking Spatial Understanding for Embodied Tasks with Large Vision-Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 346–355, Bangkok, Thailand. Association for Computational Linguistics.

Zhiyuan Feng, Zhaolu Kang, Qijie Wang, Zhiying Du, Jiongrui Yan, Shubin Shi, Chengbo Yuan, Huizhi Liang, Yu Deng, Qixiu Li, Rushuai Yang, Arctanx An, Leqi Zheng, Weijie Wang, Shawn Chen, Sicheng Xu, Yaobo Liang, Jiaolong Yang, and Baining Guo. 2025. Seeing Across Views: Benchmarking Spatial Reasoning of Vision-Language Models in Robotic Scenes. *arXiv preprint arXiv:2510.19400*.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. 2024. MotionGPT: Human Motion as a Foreign Language. *Advances in Neural Information Processing Systems*, 36.

Chengzu Li, Caiqi Zhang, Han Zhou, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2024. TopViewRS: Vision-Language Models as Top-View Spatial Reasoners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1786–1807, Miami, Florida, USA. Association for Computational Linguistics.

Julius Mayer, Mohamad Ballout, Serwan Jassim, Farbod Nosrat Nezami, and Elia Bruni. 2025. An Interactive Visual-Spatial Reasoning Benchmark for VLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing", year = "2025*. Association for Computational Linguistics.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Weilin Wan, Zhiyang Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. 2024. TLControl: Trajectory and Language Control for Human Motion Synthesis. In *Computer Vision – ECCV 2024*, volume 15095 of *Lecture Notes in Computer Science*, pages 37–54. Springer.

Zehao Wang, Minye Wu, Yixin Cao, Yubo Ma, Meiqi Chen, and Tinne Tuytelaars. 2024. Navigating the Nuances: A Fine-grained Evaluation of Vision-Language Navigation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4681–4704. Association for Computational Linguistics.

Haotian Xue, Yunhao Ge, Yu Zeng, Zhaoshuo Li, Ming-Yu Liu, Yongxin Chen, and Jiaojiao Fan. 2025. Point-It-Out: Benchmarking Embodied Reasoning for Vision Language Models in Multi-Stage Visual Grounding. *arXiv preprint arXiv:2509.25794*.

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

5 Frequently Asked Questions (FAQs)

► **What is the main scientific contribution of the paper? The current presentation combines scene generation, instruction generation, trajectory prediction, judge-based scoring, and empirical analysis, making it difficult to tell whether the contribution is the benchmark, the pipeline, or the findings.**

➠ *Short answer.* The **primary contribution** is the ***benchmark formulation itself***: a controlled evaluation setting for testing whether Vision–Language Models can convert natural-language instructions into ***continuous, spatially grounded, and plausibly executable trajectories*** in ***single-scene 3D environments***. The generation pipeline instantiates this benchmark at scale, and the empirical study demonstrates the failure modes it reveals. *[Supported by benchmark motivation and stated contributions]*

Clarification. More precisely, the paper contributes:

1. a ***single-scene, interaction-centric benchmark setting*** for grounded trajectory reasoning,
2. a ***scalable construction pipeline*** that reconstructs 3D scenes, generates tiered instructions, and evaluates predicted paths, and
3. an ***empirical diagnosis*** showing that current frontier VLMs degrade substantially on compositional, interaction-heavy, and multi-step instructions.

Why this matters. The paper should not be read as claiming that any one component—for example, the judge or the scene generator—is the sole novelty. Rather, the novelty lies in defining and instantiating a benchmark that measures a capability that existing evaluation protocols often miss: whether a model can map language into ***locally executable spatial behavior*** rather than merely recognizing semantics or reaching a rough endpoint.

Takeaway. The paper is best understood first as a ***benchmark for grounded trajectory reasoning in 3D scenes***, with the pipeline and experiments serving that central benchmark contribution.

► **How is SleepWalk meaningfully different from existing Vision-and-Language Navigation benchmarks? At first glance, this appears to be another instruction-following navigation task in 3D environments.**

➠ *Short answer.* The key distinction is that SleepWalk targets ***localized, interaction-centric reasoning within a single scene***, whereas much prior VLN work emphasizes ***long-horizon movement across rooms or buildings*** and often evaluates success primarily through endpoint reachability. SleepWalk instead requires that the ***entire trajectory*** be spatially coherent, obstacle-aware, and compatible with the intended action. *[Supported by Introduction]*

Why this is not just a smaller VLN benchmark. The central challenge here is not global exploration, but whether the model can:

- identify the correct local target,
- approach it through a feasible route,
- avoid clutter and collisions, and
- stop at a location that is compatible with the requested action.

This is precisely the regime where coarse endpoint-only evaluation can hide important failures.

Why this matters. Many embodied tasks depend on this intermediate capability between passive perception and full action execution. A system may know *what* object is relevant, yet still fail to plan a path that reaches it correctly, safely, and in the right temporal order.

Takeaway. The benchmark is not merely a narrower VLN instance; it is designed to probe a ***different layer of embodied competence***: converting language into ***continuous, executable local motion***.

► **Why restrict the environments to a single coherent scene? Doesn't that make the problem less realistic than full multi-room or building-scale navigation?**

➠ *Short answer.* The **single-scene restriction is deliberate**. The paper is not trying to replace long-horizon navigation benchmarks; it is isolating a complementary capability: ***fine-grained local reasoning around objects, obstacles, and interaction targets***. *[Supported by Section 2.1]*

Why this is a principled design choice. Restricting the task to a single scene reduces confounds from:

- room-to-room exploration,
- large-scale search,
- map-building over long horizons, and
- global navigation heuristics that can obscure local failures.

That makes it easier to study whether a model can actually produce a path that is ***interaction-compatible*** at the local level.

Why this matters. A large fraction of real embodied failures occur not because the system cannot traverse a building, but because it cannot correctly execute the *last few meters*: approaching the wrong side of an object, stopping too close to clutter, colliding with nearby items, or violating temporal structure in a multi-step task.

Takeaway. The single-scene focus should be read as ***diagnostic sharpening***, not as simplification for its own sake. It allows the benchmark to study the gap between ***seeing*** and ***acting*** in a controlled way.

► **The benchmark is highly synthetic: scenes are generated from text, instructions are generated by a model, and evaluation is also model-based. How can we be confident that the paper is measuring grounded reasoning rather than artifacts of the generation pipeline?**

➠ *Short answer.* The paper's claim is **not** that the synthetic pipeline is artifact-free or that it fully substitutes for real-world embodied evaluation. The narrower claim is that synthetic construction provides a ***controlled and scalable way*** to build a diagnostic benchmark for a capability that is currently under-measured: translating language into feasible spatial behavior inside 3D scenes. *[Supported by pipeline description and benchmark framing]*

Why the synthetic pipeline is used. The design supports:

- scalable benchmark construction,
- controlled task generation,
- diverse layouts and clutter conditions, and
- systematic difficulty tiering across easy, medium, and hard tasks.

Why the claim remains meaningful. The scientific question here is not whether the scenes are indistinguishable from all real environments. It is whether they are coherent enough to test whether a model can:

- interpret an instruction,
- reason over rendered scene observations,
- generate a collision-aware path, and
- terminate at an action-compatible location.

That is a benchmark question, and a synthetic but curated pipeline can still be a valid way to study it.

Takeaway. The paper should be read as presenting a ***controlled diagnostic testbed***, not as claiming to exhaust all forms of embodied evaluation.

► **Why use a judge model at all? Wouldn't automatic metrics such as endpoint distance, collision count, or shortest-path deviation be more objective and reproducible than GPT-5-mini scoring trajectories?**

➠ *Short answer.* Automatic geometric metrics are useful, but they are ***insufficient on their own*** for the capability this benchmark targets. A trajectory may be short, collision-free, and

geometrically plausible while still failing the instruction because it reaches the wrong object, ignores a required intermediate subgoal, or ends at a location from which the intended action cannot plausibly be executed. The judge is therefore introduced to capture ***instruction consistency and interaction compatibility*** in addition to geometry. *[Supported by evaluation protocol]*

Why the judge is used here. The paper explicitly scores four factors:

1. ***Start Location Consistency***,
2. ***Goal Satisfaction***,
3. ***Obstacle Avoidance***, and
4. ***Trajectory Efficiency***.

This means the evaluation is designed to reflect not only where a model ends up, but whether it gets there through a path that is spatially plausible and aligned with the intended action.

What the paper does and does not claim. The manuscript does *not* claim that judge-based evaluation is perfect. It makes the narrower methodological choice of using a ***fixed judge and fixed prompt*** to obtain a standardized scoring protocol for heterogeneous path outputs.

Takeaway. The judge is used not because geometric metrics are unimportant, but because ***grounded path correctness in this task is partly semantic and action-dependent***, not purely geometric.

► **A central concern is trustworthiness of the judge. Why should readers trust GPT-5-mini as evaluator, especially when the main empirical conclusions depend on its judgments?**

➠ *Short answer.* The paper's position is limited and operational: GPT-5-mini is used as a ***standardized evaluation instrument*** under fixed conditions, not as an infallible ground-truth oracle. The core argument is that, for this task, a structured judge is a practical way to compare heterogeneous trajectories under a shared notion of grounded success. *[Supported by Section 2.4 and Section 3 setup]*

What is actually claimed in the paper. The paper states that:

- GPT-5-mini is used as the judge model,
- the evaluation prompt is fixed, and
- all models are evaluated under identical conditions.

This supports consistency of comparison.

What the paper does not currently claim. The visible manuscript text does not provide human-agreement studies, multi-judge comparisons, or prompt-sensitivity analyses. Accordingly, the rebuttal should not overstate validation that is not in the paper.

Takeaway. The judge should be understood as a ***practical standardized evaluator for a difficult multimodal task***, not as proof that every score is final or uniquely correct.

► **Could poor performance simply reflect an output-interface problem? Since models are asked to emit explicit coordinate trajectories, perhaps they understand the task but fail because waypoint prediction is a difficult formatting requirement.**

➠ *Short answer.* The benchmark intentionally evaluates ***explicit trajectory prediction*** because the paper's goal is to test whether models can map language and visual context into ***concrete spatial behavior***. In embodied systems, a trajectory is not merely a textual explanation of intent; it is the object that downstream execution would actually require. *[Supported by task definition]*

Why this is more than a formatting issue. The reported failures are not described as random syntax errors. Instead, the paper emphasizes substantive error patterns:

- mislocalized start regions,
- incorrect or incomplete goal grounding,
- failure to preserve temporal order, and

- semantically plausible but physically unsafe trajectories.

These are grounding and planning failures, not merely output-format failures.

Why the quantitative pattern matters. The finding that obstacle avoidance is comparatively stronger than goal grounding suggests a structured failure pattern: models can often generate *some* navigable-looking path, but struggle to generate the *correct* path for the intended action.

Takeaway. While interface burden is a fair consideration, the evidence in the paper points to a deeper issue: ***the main bottleneck is precise instruction-grounded spatial reasoning, not merely coordinate formatting***.

► **How do we know that the easy/medium/hard instruction tiers are meaningful? Could these levels simply be heuristic labels rather than real differences in embodied reasoning difficulty?**

➠ *Short answer.* The tiers are motivated by increasing ***compositional and spatial demand***: easy tasks focus on short-range goal localization, medium tasks add structured spatial dependencies, and hard tasks introduce multi-step goals, interaction constraints, or longer planning horizons. *[Supported by Section 2.2]*

Why the tiering is empirically meaningful. The quantitative and qualitative results both support this organization:

- performance drops from easy to medium to hard,
- harder tasks expose sharper degradation in goal grounding and trajectory efficiency, and
- qualitative examples show that the dominant failure modes become more compositional and interaction-heavy as difficulty increases.

Why this matters. The benchmark is valuable not only because models fail, but because the failure pattern degrades in a structured and interpretable way across increasing task demands.

Takeaway. The difficulty tiers are not merely cosmetic labels; they are a ***diagnostic axis*** that helps reveal how grounded trajectory reasoning deteriorates as the reasoning burden grows.

► **Why are all experiments conducted in a frozen, zero-shot setting? Wouldn't benchmark-specific fine-tuning or adaptation provide a more complete picture of what the task is measuring?**

➠ *Short answer.* The paper uses a ***strict evaluation-only, zero-shot setting*** because its immediate goal is diagnostic: to measure whether current VLMs already possess this capability, not to study how well they can be adapted to it. *[Supported by task setup]*

Why this choice is methodologically useful. A zero-shot setup isolates pre-existing model capability and avoids conflating:

- general grounded reasoning ability,
- adaptation to benchmark-specific distributions, and
- improvements driven by task-specific tuning.

Why this matters. Because SleepWalk is introduced first as a benchmark, it is sensible to establish that it reveals nontrivial failure modes even before any model is tuned to it. Training-time studies can naturally follow in later work.

Takeaway. The zero-shot setting is a ***deliberate diagnostic choice***, not an omission of a required training result.

► **Only three frontier models are evaluated. Is that enough to support the paper's broader claims, or is the empirical evidence too narrow?**

➠ *Short answer.* For the current paper's purpose, yes. The experiments are used to show that SleepWalk is ***nontrivial, unsaturated, and capable of revealing structured failure modes*** even in strong contemporary systems. The paper does not claim to provide the final exhaustive leaderboard across all model families. *[Supported by evaluation section]*

Why this is sufficient for the paper's scope. The empirical role of the model set is to establish that:

1. the benchmark differentiates strong models,
2. the revealed failures are systematic rather than trivial, and
3. performance degrades under higher compositional and interaction demands.

What the results already show. Among the evaluated models, GPT-5-mini performs best across all four reported factors, but even the strongest system degrades as the tasks become harder. This supports the benchmark's central diagnostic claim.

Takeaway. Broader model coverage would certainly strengthen future benchmarking, but the present set is sufficient to establish the ***benchmark's diagnostic value***.

► **The paper reports that obstacle avoidance is relatively strong while goal grounding is much weaker. Why is that distinction important, and what does it tell us about current VLMs?**

➠ *Short answer.* This distinction is important because it separates ***generic path plausibility*** from ***instruction-faithful grounded behavior***. The reported pattern suggests that current VLMs can often produce a route that looks navigable, yet still fail to reach the ***correct interaction-compatible endpoint***. *[Supported by Section 3.2]*

Why this matters scientifically. If models were failing mainly because they could not generate motion at all, then obstacle avoidance and route plausibility would also be weak. But the paper finds a more specific bottleneck: the hard part is not merely moving through space, but binding the instruction to the right object, preserving the right sequence of actions, and terminating at the right place.

What this implies. The main weakness exposed by SleepWalk is therefore not coarse navigation, but ***precise grounded reasoning under action constraints***.

Takeaway. This is one of the paper's most informative findings: current VLMs are better at generating *a plausible path* than generating *the correct grounded path*.

► **What role does the humanoid visualization stage actually play? Is it scientifically central, or is it mainly a demonstration?**

➠ *Short answer.* The humanoid stage is a ***qualitative validation layer***, not the primary evaluation protocol. Its purpose is to test whether trajectories that look reasonable in top-down space remain plausible when executed as humanoid motion. *[Supported by Section 2.5 and Section 3.4]*

Why it is included. This layer helps expose failures that may be less obvious in a static overlay, including:

- near-collisions,
- awkward stopping positions,
- motion patterns incompatible with the intended interaction, and
- trajectories that are geometrically plausible yet operationally unsafe.

What the paper does not claim. The paper explicitly states that this stage is used for qualitative validation rather than for primary scoring.

Takeaway. Readers should interpret the humanoid visualization as ***supporting qualitative evidence*** that strengthens the benchmark's embodied interpretation, not as the benchmark's central quantitative mechanism.

► **The dataset construction details are not fully transparent: the paper starts from 1,000 MS-COCO captions but later reports 2,472 curated 3D environments. Does this undermine confidence in the benchmark construction?**

➠ *Short answer.* It does **not** undermine the benchmark's core idea, but it is a place where the presentation should be made clearer. The visible text indicates that the pipeline begins from ***1,000 MS-COCO captions*** that are ***manually filtered or rewritten*** for navigable single-scene generation, and later reports a final benchmark of ***2,472 curated 3D environments***. *[Supported by Section 2.1 and earlier benchmark summary]*

How to interpret this conservatively. The reasonable reading is that the construction process expands from an initial description pool into a larger curated environment set through generation and curation. However, the exact mapping should be documented more explicitly in the paper.

Why this matters. This is primarily a ***transparency and documentation issue***, not a conceptual flaw in the benchmark formulation itself.

Takeaway. The benchmark's contribution remains intact, but the manuscript would benefit from a clearer description of the relationship between source descriptions, generation stages, filtering, and final retained environments.

► **What is the main empirical takeaway of the paper? Is the claim simply that current VLMs are weak at navigation, or something more specific?**

➠ *Short answer.* The paper makes a more specific claim than "current VLMs are bad at navigation." Its core finding is that strong VLMs may understand the broad semantics of an instruction and even generate roughly navigable paths, yet still struggle to produce ***precise, spatially grounded, interaction-compatible trajectories*** in 3D scenes, especially under ***occlusion, compositional constraints, and multi-step planning demands***. *[Supported by Abstract and Results]*

Why this is the right interpretation. The qualitative analysis identifies recurring failures in:

- start grounding,
- goal grounding,
- temporal ordering, and
- physical executability.

The factor-wise quantitative results reinforce this by showing strong relative performance on obstacle avoidance but weaker performance on goal satisfaction.

Takeaway. The central message is that there remains a measurable gap between ***semantic understanding*** and ***grounded executable behavior***. SleepWalk is valuable because it makes that gap visible in a controlled and scalable way.

A Appendix

This appendix provides additional details on benchmark construction, prompt design, evaluation, and reproducibility. The goal is to make the setup of SleepWalk easy to understand, reproduce, and extend.

A.1 Benchmark overview

SleepWalk evaluates whether a vision-language model can convert a natural-language instruction and rendered scene observations into a spatially coherent, executable trajectory inside a single-scene 3D environment. The benchmark is intentionally designed to emphasize localized, interaction-centric reasoning rather than long-range room-to-room exploration.

Table A1: Core benchmark statistics reported in the current paper.

Statistic	Value
Number of environments	2,472
Difficulty tiers	3
Instructions per scene	9
Instructions per tier	3
Total scene–instruction pairs	22,248
Rendered views per scene	2
Primary evaluation factors	4

A.2 Scene generation and curation

Each benchmark instance begins from a textual scene description specifying a single coherent indoor or outdoor environment. We convert the description into a 3D scene using Hunyuan3D-3.0. Because the paper focuses on instruction-grounded local navigation and interaction, we intentionally exclude multi-room or highly fragmented generations that would shift the task toward global exploration rather than precise within-scene grounding.

After generation, scenes are manually filtered for the following properties:

- **Single-scene coherence:** the environment should correspond to one visually and spatially coherent scene rather than multiple disconnected spaces.
- **Navigability:** the scene should contain sufficient free space for a humanoid-scale agent to move through the environment.
- **Object recognizability:** major landmarks and target objects should be visually identifiable from rendered views.
- **Geometric plausibility:** scenes with severe self-intersections, floating objects, degenerate layouts, or obviously broken geometry are removed.

For each accepted environment, we render two views: a top-down view and an oblique view. The top-down view provides an explicit picture of free space, clutter, and approximate route structure, while the oblique view provides appearance cues, object identity, and interaction context.

A.3 Instruction generation

For each reconstructed scene, we use Qwen3-8B-VL to generate nine navigation instructions, grouped into three difficulty tiers: easy, medium, and hard. The tiering is meant to expose progressively harder forms of embodied reasoning rather than merely longer paths.

We instruct the generation model to produce concise, scene-grounded, and executable instructions. Instructions are rejected if they refer to nonexistent objects, require ambiguous global references, or describe actions that cannot be approximately assessed from the rendered views.

Table A2: Task-tier design in SleepWalk.

Tier	Primary capability	Typical instruction type	Common failure mode exposed
Easy	Single-goal localization	Move to a clearly identifiable object or region	Incorrect endpoint despite broadly plausible path
Medium	Compositional grounding	Resolve two landmarks or a simple ordered objective	Failure to preserve reference binding or temporal order
Hard	Multi-step interaction-aware planning	Pick/place, approach-then-move, or action-compatible stopping	Endpoint/action mismatch, unsafe stopping position, or incoherent route

A.4 Trajectory representation

Given rendered observations V and a natural-language instruction I, a tested model predicts a continuous trajectory

$$T = \{p_t\}_{t=1}^{T}, \qquad p_t \in \mathbb{R}^3. \tag{A.1}$$

For evaluation, the trajectory is represented as an ordered waypoint sequence and projected onto the top-down view for judge scoring. The projection is used only for standardized visual comparison; the intended interpretation remains a path in the underlying 3D environment.

A valid trajectory should satisfy three properties:

- it should begin near the intended start region,
- it should avoid obvious collisions or implausible shortcuts through obstacles, and
- it should terminate at a location compatible with the requested action.

B Prompt Templates

B.1 Instruction-generation prompt format

```
### Role
You are an expert Embodied-Agent Instruction Generator. Your goal is to analyze visual inputs of a 3D environment and generate precise, executable tasks for a robotic agent.
### Inputs
I have provided two images of the same environment:
1. Image 1: Oblique View (Perspective)
2. Image 2: Top-Down View (Map)
### Critical Pre-Condition
Analyze the Top-Down View first.
If the Top-Down view is incomplete, obstructed, or not clearly visible, you must ignore all other instructions and output ONLY this exact phrase:
"The view is not clear to generate instructions"
---
### Phase 1: Environment Analysis
If the views are clear, analyze both images to construct a mental model of the scene:
- Use the Top-Down View for: Global layout, spatial relationships, distances, and valid paths.
- Use the Oblique View for: Object identification, visual attributes, affordances (what can be opened/lifted), and accessibility.
- Identify: All visible, interactable objects (e.g., furniture, appliances, small items).
### Phase 2: Task Generation
Generate exactly {num_per_level} tasks for EACH of the 3 complexity levels defined below (Total tasks: {num_per_level * 3}).
```

Task Levels:
- LEVEL_1 (Navigation): Movement from Object A to Object B. No manipulation.
  Constraint: INTERACT must be "none".
- LEVEL_2 (Simple Interaction):* Interaction with 1-2 specific objects.
  Constraint: INTERACT list must contain 1-2 objects.
- LEVEL_3 (Complex Interaction):* Interaction sequences involving 3+ objects or states.
  Constraint: INTERACT list must contain 3+ objects.

Strict Constraints & Guardrails

1. Object Grounding:
   - Every START and END location must refer to a specific, visible object found in the images.
   - Correct: "Start: Near the red chair"
   - Incorrect: "Start: Near the wall" or "Start: At the start point"
2. Forbidden Terminology (Spatial Hallucination):
   - NEVER use viewpoint-dependent or relative directional terms.
   - BANNED WORDS: left, right, front, back, center, centre, middle, top, bottom, upper, lower.
   - Tasks must be valid regardless of the agent's facing direction.
3. Object Consistency:
   - Do not hallucinate objects. Only reference items clearly visible in the provided images.

Output Format

Step 1: Provide an *ENVIRONMENT SUMMARY* (2-3 sentences describing the room type and listing 10-15 key visible objects).

Step 2: Output the task list following this exact schema:

LEVEL_1 | TASK: <instruction> | START: Near <object> | END: Near <object> | INTERACT: none

LEVEL_2 | TASK: <instruction> | START: Near <object> | END: Near <object> | INTERACT: <obj1>, <obj2>

LEVEL_3 | TASK: <instruction> | START: Near <object> | END: Near <object> | INTERACT: <obj1>, <obj2>, <obj3>

One-Shot Example

Use this structure as a template:

ENVIRONMENT SUMMARY:

This is a modern office space featuring a central workstation and a lounge area. Key objects include: desk, ergonomic chair, monitor, keyboard, filing cabinet, bookshelf, cactus pot, whiteboard, leather sofa, glass coffee table, floor lamp, laser printer, window blinds, and office door.

LEVEL_1 | TASK: Navigate from the entrance to the work desk | START: Near the office door | END: Near the desk | INTERACT: none

LEVEL_1 | TASK: Move from the storage area to the seating zone | START: Near the filing cabinet | END: Near the leather sofa | INTERACT: none

LEVEL_1 | TASK: Walk from the workstation to the meeting area | START: Near the desk | END: Near the whiteboard | INTERACT: none

LEVEL_2 | TASK: Pick up the document from the printer and place it on the desk | START: Near the laser printer | END: Near the desk | INTERACT: document, desk

LEVEL_2 | TASK: Retrieve the book from the bookshelf and place it on the coffee table | START: Near the bookshelf | END: Near the glass coffee table | INTERACT: book, coffee table

LEVEL_2 | TASK: Take the pen from the desk and place it near the whiteboard | START: Near the desk | END: Near the whiteboard | INTERACT: pen, whiteboard

LEVEL_3 | TASK: Open the filing cabinet, retrieve a folder, close the cabinet, and place the folder on the desk | START: Near the filing cabinet | END: Near the desk | INTERACT: cabinet door, folder, desk
LEVEL_3 | TASK: Pick up the laptop from the desk, open it, then move to the sofa | START: Near the desk | END: Near the leather sofa | INTERACT: laptop, laptop lid, leather sofa
LEVEL_3 | TASK: Turn on the floor lamp, pick up the book from the shelf, then place it on the desk | START: Near the floor lamp | END: Near the desk | INTERACT: lamp switch, book, desk
Task:
Analyze the provided images and generate the output now.

B.2 Trajectory-prediction prompt format

You are given:
- An oblique view of the same scene
- A top-down view of a 3D scene

Task: Compute the shortest valid walking path while avoiding obstacles on the top-down view.

==========
GRID

The top-down view is scene with a discrete 25×25×25 grid.

Valid coordinates:
0 <= x,y,z <= 24
Integers only.
Do NOT use pixel or continuous values.

Origin: (0,0,0) = bottom-left-front floor corner
Positive X axis goes right, positive Y axis goes up, positive Z axis goes inside the screen

Floor = XZ plane.
Estimate (x, z) positions using the top-down view.
Estimate the correct floor height (y) using the oblique view.
All path points must lie on the walkable floor plane.

==========
CONSTRAINTS

- All the coordinates should lie on the top-down view
- Determine the ranges of all obstacles (furniture, walls, vehicles, objects, elevated surfaces) on the top-down view
- None of the coordinates (including the coordinates for the starting point) should intersect or touch any obstacle or lie under any object
- Stay inside grid
- Remain on the floor plane
- Carpets, roads, stairs are walkable

==========
OUTPUT (STRICT)

Return a valid json of the following format.

Format:
[
  {{”point”: [x0, y0, z0], ”label”: ””}},
  {{”point”: [x1, y1, z1], ”label”: ””}},
  ...
  {{”point”: [xN, yN, zN], ”label”: ””}}
]

B.3 Judge prompt format

You are a visual navigation judge trained to evaluate instruction-guided trajectories in 3D environments.
You must ground all judgments strictly in visible evidence from the images.
Do not assume intent or hidden scene structure.
If evidence is insufficient, explicitly state that it is unclear.

==========
NAVIGATION TASK
==========
{task}

==========
INPUTS
==========
- Image 1: Oblique view of the scene.
- Image 2: Top-down view of the same scene.
- The predicted trajectory is shown using green stars and green connecting lines.
- The green dot marks the start of the trajectory.
- The red dot marks the end of the trajectory.

==========
OUTPUT FORMAT (JSON ONLY)
==========
{{
  start_location_accuracy: {{
    score: 0-5 or N/A,
    justification: string
  }},
  goal_completion: {{
    score: 0-5 or N/A,
    justification: string
  }},
  obstacle_avoidance: {{
    score: 0-5 or N/A,
    justification: string
  }},
  trajectory_efficiency: {{
    score: 0-5 or N/A,
    justification: string
  }},
  overall_summary: string
}}

B.4 Judge rubric

The following table shows the judge rubric that was used to score each predicted trajectory.

Table B1: Pointwise judge rubric used to score each predicted trajectory.

Factor	Definition
Start Location Consistency	Whether the predicted path begins in the correct initial region or near the intended starting landmark.
Goal Satisfaction	Whether the trajectory ends at a location that satisfies the instruction and is compatible with the requested action.
Obstacle Avoidance	Whether the route avoids obvious collisions, barrier crossings, or geometrically implausible shortcuts.
Trajectory Efficiency	Whether the route is reasonably direct and avoids unnecessary detours relative to the stated objective.

C Experimental Settings and Reproducibility

All experiments are conducted in a zero-shot evaluation-only regime. No model is fine-tuned, adapted, or exposed to benchmark-specific gradient updates. Each model receives the same task inputs: the same rendered views, the same natural-language instruction, and the same output formatting requirement.

C.1 Evaluated models

Table C1: Models evaluated in the current paper.

Model	Inputs	Output	Adaptation
Qwen3-VL	Top-down view, oblique view, instruction	3D waypoint sequence	None
Gemini Robotics ER-1.5	Top-down view, oblique view, instruction	3D waypoint sequence	None
GPT-5-mini	Top-down view, oblique view, instruction	3D waypoint sequence	None

C.2 Decoding and scoring protocol

Let $\bar{s}_k(\tau) \in [0, 1]$ denote the normalized score assigned to trajectory τ on factor $k \in \{\text{start, goal, obs, eff}\}$. We compute tier-level factor scores by averaging valid (non-$\mathrm{N/A}$) normalized scores within each difficulty tier, and we compute overall factor scores by averaging across tiers with equal tier weight.

For reproducibility, we will be releasing the SLEEPWALK dataset containing the 3D environments and the corresponding instructions upon acceptance.

D Additional Qualitative Examples

Figure D1 is examples generated by GPT5-mini model across the three SleepWalk difficulty tiers (Easy, Medium and Hard). Below are the corresponding Instructions we've provided to the model:

- LEVEL 1 (Easy)
  TASK: Walk from the trash can to the whiteboard
  START: Near the trash can
  END: Near the whiteboard
  INTERACT: none
- LEVEL 2 (Medium)
  TASK: Place the green book from the shelf onto the teacher's desk
  START: Near the shelf
  END: Near the teacher's desk
  INTERACT: green book, teacher's desk
- LEVEL 3 (Hard)
  TASK: Move from the whiteboard to the student desk, pick up the book, and place it on the

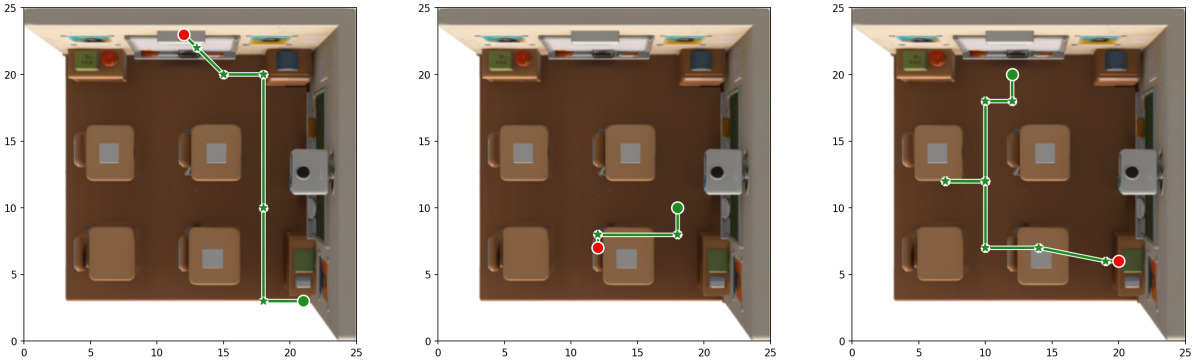


Figure D1: Additional qualitative examples generated by GPT5-mini model across the three SleepWalk difficulty tiers (Easy, Medium, Hard).

teacher's podium
START: Near the whiteboard
END: Near the teacher's podium
INTERACT: student desk, book, teacher's podium

E Limitations, Failure Modes, and Ethical Considerations

E.1 Benchmark limitations

SleepWalk isolates a useful intermediate capability between perception and action, but it does not fully solve embodied evaluation. First, the benchmark uses reconstructed single-scene environments rather than fully interactive physics-based worlds, so object dynamics and low-level contact mechanics are only approximately reflected. Second, the primary evaluation protocol relies on a strong judge model rather than human annotation for every sample, which may introduce scoring bias despite the use of explicit rubrics. Third, top-down trajectory overlays simplify comparison, but they do not capture every detail of embodied execution.

E.2 Observed failure modes

Across models, we repeatedly observe three broad failure patterns:

- **Mislocalized starts:** the model begins from the wrong region even when the overall target object is correctly identified.
- **Incorrect or partial goal grounding:** the path is plausible but ends at the wrong object, the wrong side of the object, or a location that is not interaction-compatible.
- **Geometrically unsafe execution:** the predicted path appears semantically sensible but passes through clutter, clips furniture, or stops in a way that a humanoid agent could not realistically execute.

E.3 Ethical considerations

This benchmark is intended to support safer and more reliable embodied multimodal systems by diagnosing failures in grounded spatial reasoning before real-world deployment. At the same time, better language-conditioned navigation can improve autonomous capabilities in ways that may be dual-use.