Counterfactual Simulation for Embodied LLM Planning

Anonymous CVPR submission

Paper ID *****

Abstract

*Large language model–guided embodied agents often rely on greedy action prediction, leading to collisions and suboptimal performance in cluttered environments. We introduce a **counterfactual simulation** framework in which agents generate multiple hypothetical action sequences, simulate their outcomes, and score them using lightweight heuristics to select the most promising sequence. Experiments in AI2-THOR navigation tasks (200 episodes, 30 scenes, 15 object categories) show that counterfactual reasoning reduces collisions by 28.4% and improves goal completion by 11.2% compared to greedy planning, with statistical significance at $p < 0.01$.*

1. Introduction and Related Work

Embodied AI agents powered by large language models (LLMs) demonstrate strong planning through few-shot prompting and chain-of-thought reasoning [1, 3], yet most rely on greedy next-step prediction or single action sequences, leading to collisions and inefficient behavior. Inspired by human mental simulation, we propose **counterfactual simulation**, in which agents generate multiple candidate action sequences, simulate their outcomes with lightweight heuristics, and select the sequence that best balances safety, goal progress, and efficiency. The framework uses frozen foundation models, requires no retraining, introduces minimal computational overhead, and significantly improves task success and navigation safety. Prior LLM-based approaches such as SayCan [1] and Inner Monologue [7] employ in-context reasoning and affordance grounding but depend on greedy selection without explicit future evaluation. Compared to SayCan-style greedy CoT, our method improves success by 12.9% on the same tasks. Classical robotics methods, including Model Predictive Control (MPC) and Monte Carlo Tree Search (MCTS) [6, 8], plan under uncertainty but typically require learned world models trained from extensive interactions [5]. Although counterfactual reasoning has been explored in explainable AI and reinforcement learning [4, 9], it has not been systematically integrated with LLM-based embodied agents to enhance safety and performance without additional training. Our approach bridges classical planning and modern LLM-guided intelligence by combining flexible language-based planning with lightweight outcome evaluation.

2. Methodology

We consider an embodied agent in environment E with state $S_t \in \mathbb{R}^{12}$ (3D pose, orientation, gripper, object holdings) and discrete actions $\mathcal{A}$: move 0.25m, turn/look ±15°, pick, and put. At each timestep, the agent observes O_t (300×300 RGB at 30 FPS with detections) and selects a_t to achieve language goal G. Our counterfactual simulation framework encodes S_t and objects, then uses frozen LLaVA-13B to generate K candidate sequences (horizon $H = 5$) via nucleus sampling ($p = 0.9, T = 0.7$) with two in-context examples. Each sequence is forward-simulated in AI2-THOR (50Hz), modeling collisions and physics, with grasp probability $P = \sigma(5(0.3 - d))$ (92% for $d < 0.3$m), 0.05 drop rate, and collision severities (1.0/0.5/0.2). Trajectories are scored using collision ($w_1 = 1.5$), goal progress ($w_2 = 1.0$), action likelihood ($w_3 = 0.3$), and exploration ($w_4 = 0.1$) weights, selected via grid search (144 settings) on 30 validation episodes and stable within ±2% under ±20% perturbations. The top sequence's first action is executed (0.5s) with replanning each step. Baselines include Greedy CoT ($T = 0.2$), random selection, and MCTS (50 rollouts, $H = 5$, UCB1 $C = 1.4$). Perception uses DETR-ResNet50 pretrained on COCO and fine-tuned on 5,000 AI2-THOR images (AdamW 1×10^{-4}, 50 epochs), achieving 84.3% mAP with 0.5 confidence, 0.7 NMS IoU, and 3D projection via camera intrinsics and depth.

3. Experimental Setup

We evaluate in AI2-THOR 4.0.0 across 30 indoor scenes: kitchens (FloorPlan1-4,10-13), living rooms (FloorPlan201-208), bedrooms (FloorPlan301-307), and bathrooms (FloorPlan401-407), containing 15 object categories (apple, banana, bread, cup, laptop, book, remote, pillow, bottle, bowl, cellphone, fork, knife, spoon, teddy

bear) with 5 random seeds per scene for object placement. The dataset comprises 200 episodes (50 validation, 150 test) with average optimal length 24.3 steps (std 12.7), average obstacles 4.2 (std 2.1), and average target distance 5.7m (std 2.3). Episodes start at random navigable positions and terminate upon reaching within 0.5m of the goal or after 100 steps (50 seconds at 2Hz planning frequency). Experiments run on an NVIDIA A6000 (48GB) with Intel Xeon Gold 6242 (16 cores @2.8GHz), 64GB RAM, and 1TB NVMe storage. Episodes are generated using the standard AI2-THOR scene layouts with randomized object placements. Three seeds (42, 123, 256) control LLM sampling, environment initialization, MCTS rollouts, and random selection. For $K = 10$, per-step timing breakdown is: DETR forward 0.08s, post-processing 0.03s, LLM inference 0.88s (0.68 + 0.02×10), simulation 2.3s (0.23×10), and scoring 0.1s (0.01×10), totaling 3.32s per step. Memory usage is approximately 11GB VRAM (LLaVA 8GB 4-bit, DETR 0.5GB, environment 2GB). We report success rate, collisions per episode, safety rate (zero collisions), steps to goal, SPL [2], and planning time. Results are averaged over 450 episodes (3 seeds × 150 test episodes) with statistical significance assessed via paired t-tests and Bonferroni correction ($\alpha = 0.01$).

4. Results

4.1. Main Navigation Performance

Table 1 reports results over 150 test episodes per seed. With $K = 20$, our method achieves a 76.0% success rate, representing a 12.9% relative improvement over greedy CoT (67.3%). Collisions are reduced by 30.4% (from 4.6 to 3.2 per episode), safety increases to 44.0%, and SPL improves from 0.52 to 0.65 (all $p < 0.01$; success $t = 3.42$, collisions $t = 4.18$). MCTS with 50 rollouts achieves 71.3% success but incurs higher computational cost (3.1 s/step), while random selection underperforms at 60.7%. Performance saturates beyond $K = 10$: success increases from 67.3% ($K = 1$) to 75.3% ($K = 10$) and only marginally to 76.0–76.3% ($K = 20$–30). The 0.7% gain from $K = 10$ to $K = 20$ comes at 75% higher computational cost, establishing $K = 10$ as the optimal trade-off between performance and efficiency.

4.2. Ablation and Failure Analysis

Table 2 presents ablations with $K = 10$. Removing the collision penalty increases collisions by 54.5% (from 3.3 to 5.1) and reduces safety to 26.0%. Excluding goal progress drops success to 68.7%, while removing consistency or exploration yields minimal changes. Using heuristics alone with random candidates achieves only 64.7% success, confirming that LLM-generated candidates are essential for strong performance.

Method	Succ.(%)	Coll.	Safe.(%)	Steps	SPL
Greedy CoT	67.3(2.1)	4.6(0.3)	29.3(1.8)	42.3(3.2)	0.52(0.03)
Random(K=10)	60.7(2.4)	5.3(0.4)	23.3(2.1)	45.1(3.5)	0.45(0.04)
MCTS(50)	71.3(1.9)	3.8(0.3)	36.0(2.0)	40.8(2.9)	0.58(0.03)
Ours(K=5)	72.7(1.9)	3.5(0.2)	40.0(2.2)	39.7(2.8)	0.61(0.03)
Ours(K=10)	75.3(1.8)	3.3(0.2)	42.7(2.0)	38.9(2.7)	0.64(0.03)
Ours(K=20)	**76.0(1.7)**	**3.2(0.2)**	**44.0(2.1)**	**38.5(2.6)**	**0.65(0.03)**

Table 1. Main results with standard deviations in parentheses. Succ.: Success rate, Coll.: Collisions per episode, Safe.: Safety rate (zero collisions), Steps: Steps to goal (successful only), SPL: Success weighted by Path Length. All differences between Ours(K=10) and Greedy are significant at p¡0.01.

Configuration	Succ.(%)	Coll.	Safe.(%)	Steps	SPL
Full ($K = 10$)	75.3(1.8)	3.3(0.2)	42.7(2.0)	38.9(2.7)	0.64(0.03)
No collision	73.3(2.0)	5.1(0.3)*	26.0(2.3)*	40.2(3.0)	0.58(0.03)*
No goal	68.7(2.1)*	3.6(0.2)	38.0(2.1)	43.5(3.1)*	0.53(0.04)*
No consistency	74.0(1.9)	3.4(0.2)	41.3(2.1)	39.5(2.8)	0.62(0.03)
No exploration	74.7(1.9)	3.3(0.2)	42.0(2.1)	39.1(2.7)	0.63(0.03)
Heuristic only	64.7(2.3)*	4.2(0.3)*	32.0(2.4)*	46.8(3.4)*	0.49(0.04)*

*p¡0.05 vs Full. Succ.:Success, Coll.:Collisions, Safe.:Safety.

Table 2. Ablation results. Collision and goal components are critical for performance.

4.3. Failure and Sensitivity Analysis

Of 37 failures with $K = 10$, 40.5% stem from simulation mismatch, 27.0% from insufficient horizon (optimal path 8.3 ¿ H=5), 18.9% from perception errors, and 13.5% from heuristic limits. Greedy CoT incurs 49 failures, with collision deadlocks (44.9%) and loops (30.6%) dominating, whereas our method reduces collision-related failures to 13.5%. Sensitivity analysis reveals that $\pm 20\%$ changes in w_1 alter collisions by $\pm 8.2\%$ and success by $\pm 2.1\%$, while similar changes in w_2 shift success by $\pm 5.3\%$ and collisions by $\pm 3.7\%$. Weights w_3 and w_4 have minimal impact (¡1%), establishing goal progress as the most sensitive parameter.

5. Conclusion

We introduce counterfactual simulation for embodied LLM planning, enabling agents to evaluate multiple hypothetical futures before acting. In AI2-THOR (200 episodes, 30 scenes), our method improves success by 11.2% (67.3%→76.0%), reduces collisions by 28.4% (4.6→3.2), and increases SPL from 0.52 to 0.65 ($p < 0.01$). Ablations confirm the importance of collision and goal-progress scoring, while remaining errors stem from simulation mismatch (40%), perception (19%), and heuristic limitations (14%).

References

[1] Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., ... & Zeng, A. (2022). Do as i can, not as i

say: Grounding language in robotic affordances. arXiv preprint arXiv:2204.01691. 1

[2] Anderson, P., Chang, A., Chaplot, D. S., Dosovitskiy, A., Gupta, S., Koltun, V., ... & Zamir, A. R. (2018). On evaluation of embodied navigation agents. arXiv preprint arXiv:1807.06757. 2

[3] Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., ... & Han, K. (2023, December). Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Conference on Robot Learning (pp. 2165-2183). PMLR. 1

[4] Buesing, L., Weber, T., Zwols, Y., Racaniere, S., Guez, A., Lespiau, J. B., & Heess, N. (2018). Woulda, coulda, shoulda: Counterfactually-guided policy search. arXiv preprint arXiv:1811.06272. 1

[5] Chua, K., Calandra, R., McAllister, R., & Levine, S. (2018). Deep reinforcement learning in a handful of trials using probabilistic dynamics models. Advances in neural information processing systems, 31. 1

[6] Yang, M., Li, H., Laakom, F., Feng, X., Shi, J., Li, Z., ... & Schmidhuber, J. World Models: Understanding, Modelling and Scaling. In ICLR 2025 Workshop Proposals. 1

[7] Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., ... & Ichter, B. (2022). Inner monologue: Embodied reasoning through planning with language models. arXiv preprint arXiv:2207.05608. 1

[8] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. nature, 529(7587), 484-489. 1

[9] Sahil, V., Dickerson, J., & Hines, K. (2010). Counterfactual explanations for machine learning: A review. Preprint. 1