Agent-as-a-Judge: Multi-Turn Rubric-Guided Hallucination Detection for Embodied Agents

Sam Blouir[1] Buse Demir[2]

[1]George Mason University, Fairfax, VA, USA
[2]Harvard University, Cambridge, MA, USA
sblouir@gmu.edu, bdemir@mgh.harvard.edu

Abstract

Symbolic interfaces for embodied agents allow large language models (LLMs) to plan in terms of goals, subgoals, actions, and transition rules. However, even with complete and accurate environment context, LLM planners frequently hallucinate entities or effects that are incompatible with the environment, undermining reliability [7].

We present an inference-time *agent-as-judge* interface based on a rubric-guided planner-judge architecture for the Transition Modeling (*TM*) module of the Embodied Agent Interface (EAI) benchmark [8]. A planner agent proposes multiple candidate transition rules, while a distinct judge agent assigns structured rubric scores that explicitly capture hallucination-related criteria and environment consistency.

Our orchestrator uses these rubric scores to filter hallucinated candidates and select a final output, without any additional model training. Instantiated for VIRTUALHOME Transition Modeling [10] and evaluated on the local EAI VIRTUALHOME validation split [8], a 4-sample planner-judge variant improves choice accuracy from 42.1% to 51.7% and reduces obvious hallucinations relative to a single-sample planner baseline.

1 Introduction

Embodied agents must interpret language instructions, decompose them into structured plans, and execute those plans through environment-specific actions and state transitions. The Embodied Agent Interface (EAI) Challenge [8] isolates this process into four modular tasks: *Goal Interpretation*, *Subgoal Decomposition*, *Action Sequencing*, and *Transition Modeling*. Each module operates over symbolic descriptions (objects, actions, predicates, and rules), decoupling high-level reasoning from perception and low-level control. In this work we focus on the Transition Modeling (TM) module, which requires predicting how symbolic world states change when actions are applied.

LLMs are a natural fit for this symbolic setting: they can map language descriptions to goals, hierarchies of subgoals, executable action sequences, and transition rules [1, 2, 6]. Yet they often hallucinate: inventing objects not present in the scene, proposing actions outside the admissible action set, or asserting impossible state changes [7]. In EAI, such failures are especially costly because they propagate across modules and obscure where the pipeline breaks. TM is particularly brittle: hallucinated preconditions and effects can silently corrupt downstream planning even when the rest of the pipeline is correct.

Terminology (avoiding "agent" overload). In EAI, an *embodied agent* refers to the downstream executor that ultimately acts in BEHAVIOR/VIRTUALHOME [10, 12]. In this paper, we use additional *LLM agents* as internal components: a *planner agent* proposes candidate symbolic outputs for TM, and a *judge agent* audits candidates against an environment-aware rubric and returns scores and hallucination flags. A separate *orchestrator* uses these signals to select, reject, or repair proposals.

Component	Output	Role / capability
Planner agent G	candidate y	propose / revise symbolic TM outputs
Judge agent J	rubric scores r, flags	evaluate only (no proposing)
Orchestrator	selected $\hat{y}$	select / reject / trigger repair
Embodied executor	environment actions	execute $\hat{y}$ (outside our interface)

Table 1: Disentangling roles: "agent" in EAI (executor) vs. LLM agents (planner/judge).

Overview. We study hallucination-aware interfaces for LLM-based Transition Modeling in EAI. Rather than trusting a single autoregressive sample, we separate proposal and audit and compare: (i) a **single-sample planner** baseline (one completion, no judge), and (ii) a **4-sample planner-judge** variant, where a planner agent generates a committee of four candidates and a judge agent scores each candidate against an environment-aware rubric. The orchestrator then filters candidates that fail hallucination checks and selects the highest-scoring remaining candidate. We evaluate both systems on the local EAI VIRTUALHOME TM validation split using the publicly released scoring code [8].

Contributions. Our contributions are:

1. **An agent-as-judge interface for EAI Transition Modeling.** A planner-judge architecture tailored to TM, with environment-aware rubrics that decompose evaluation into interpretable criteria including explicit hallucination checks.
2. **A simple 4-sample planner-judge variant for TM.** We instantiate the interface with a single planner model and a rubric-guided judge, and compare a single-completion baseline against a 4-completion planner-judge variant on the local EAI VIRTUALHOME TM validation set.
3. **Empirical and qualitative analysis.** We report improvements in TM choice accuracy and analyze typical hallucination patterns that the judge catches (and misses), connecting our findings to broader work on LLM hallucinations and LLM-as-a-judge evaluation [7, 9, 5, 3].

2 Related Work

Embodied agents and symbolic interfaces. Simulated household environments such as BEHAVIOR and VIRTUALHOME provide rich testbeds under controlled conditions [10, 12]. BEHAVIOR emphasizes realistic 3D scenes and object interactions, while VIRTUALHOME focuses on scripted household activities and program-like representations. The EAI Challenge [8] abstracts these environments into a symbolic interface, enabling evaluation of language-to-symbol pipelines independent of perception and low-level control.

LLMs for planning and robotics. Recent work explores LLMs as planners and high-level controllers for robotics and embodied tasks, using natural language, code, or symbolic action languages [1, 2, 6]. SayCan grounds language in value functions over robot skills [1], RT-2 uses vision-language-action models that transfer web knowledge to robotic control [2], and Inner Monologue combines world feedback with language-based planning [6]. Our work focuses on EAI's modular symbolic interface and on hallucination-aware selection within that interface, instantiated for TM.

Hallucination detection and uncertainty. Hallucinations are a well-documented failure mode of LLMs [7]. A growing body of work studies hallucination detection using token probabilities, semantic entropy, and related uncertainty signals. Farquhar et al. use semantic entropy over clusters of generations to detect confabulated answers [4]; Quevedo et al. analyze token probability profiles as a reference-free indicator of hallucinations [11]; SelfCheckGPT compares multiple stochastic samples for consistency [**?**]; and Semantic Entropy Probes approximate semantic entropy directly from model activations for cheaper detection [**?**]. Our agent-as-judge interface is complementary:

instead of training a separate detector, we leverage rubric scores from a judge model and treat them as an inexpensive hallucination signal specialized to the TM task.

LLM-as-judge and rubric evaluation. LLM-based evaluators can correlate well with human judgments when prompted with structured rubrics. G-Eval [9] uses form-filling and chain-of-thought prompting to produce rubric scores aligned with human ratings, and recent surveys on LLM-as-a-judge [5] catalog opportunities and risks of using LLMs as evaluators. Deployment-focused case studies [3] show that rubric-based judging can help detect hallucinations when comparing responses to context in retrieval-augmented generation settings. We apply this paradigm to EAI TM, where the context includes explicit environment state, object ontology, and action schemas, and the rubric is tuned to detect environment-specific hallucinations.

3 Method

Our interface wraps the TM module in a common orchestration loop that separates *proposal* from *audit* and uses judge scores to manage hallucination risk. We instantiate and evaluate this interface for the Transition Modeling (TM) module in VIRTUALHOME.

3.1 Planner-Judge (Two-Agent) Architecture

For TM, we instantiate a planner agent G and a judge agent J:

- A **planner agent** G, an instruction-tuned LLM that maps a prompt x and environment context c to candidate symbolic TM outputs y (e.g., precondition/effect sets or next-state predictions).
- A **judge agent** J, an instruction-tuned LLM that audits candidates using a TM-specific rubric and returns a vector of scores plus hallucination flags.

Given x, c, the planner agent produces a committee of N candidates via independent sampling:

$$y^{(1)}, \ldots, y^{(N)} \sim G(\cdot \mid x, c).$$

For each candidate $y^{(i)}$, the judge agent produces rubric scores

$$r^{(i)} = (r_1^{(i)}, \ldots, r_K^{(i)})$$

and a hallucination flag $h^{(i)} \in \{0, 1\}$. Each dimension of $r^{(i)}$ corresponds to a rubric criterion (e.g., semantic match to examples, executability, environment consistency, non-hallucination), and the hallucination flag summarizes whether the candidate appears to hallucinate objects, actions, or effects given the provided environment context.

Orchestration with rubric-based selection. The orchestrator treats the judge as a task-specific hallucination detector and uses two simple heuristics:

1. **Filtering.** Discard candidates whose hallucination flag is set ($h^{(i)} = 1$) or whose total rubric score $\sum_k r_k^{(i)}$ falls below a fixed threshold τ.
2. **Selection.** Among the remaining candidates, choose the one with the highest total rubric score; if all candidates are filtered, fall back to the highest-scoring candidate globally (even if flagged) to ensure a deterministic output.

Algorithm 1 summarizes the inference-time loop.

In this paper we compare $N = 1$ (no judge, single-sample baseline) against $N = 4$ with rubric-guided filtering and selection. Larger committees are straightforward but more expensive.

3.2 Transition Modeling Instantiation

Task. The TM module predicts state changes from actions, typically as preconditions/effects or next-state predictions. In VIRTUALHOME, the model receives an action (e.g., `put_on(clothes_jacket, character)`), the current symbolic state, and must output preconditions and effects that are

Algorithm 1 Rubric-guided orchestration for EAI Transition Modeling

Require: Prompt x, context c, planner G, judge J, score threshold τ

Planner proposes $y^{(1)}, \ldots, y^{(N)} \sim G(\cdot \mid x, c)$
for $i = 1$ to N **do**
  Judge audits $y^{(i)}$: scores $r^{(i)}$, flag $h^{(i)} \leftarrow J(y^{(i)}, c)$
  Compute total score $s^{(i)} \leftarrow \sum_k r_k^{(i)}$
end for
$\mathcal{I} \leftarrow \{i : h^{(i)} = 0 \text{ and } s^{(i)} \geq \tau\}$ (low-risk indices)
if $\mathcal{I} \neq \emptyset$ **then**
  $i^\star \leftarrow \arg\max_{i \in \mathcal{I}} s^{(i)}$
else
  $i^\star \leftarrow \arg\max_{i \in \{1,\ldots,N\}} s^{(i)}$ (fallback)
end if
return $\hat{y} \leftarrow y^{(i^\star)}$

compatible with the environment's dynamics [10]. The EAI TM starter code provides a local validation split and a scoring script that compares predicted transitions against ground-truth transitions [8].

Rubric. The judge evaluates each candidate using a small, fixed rubric with integer scores (e.g., 0–3) for each criterion:

- **Format and well-formedness.** Are the outputs structurally valid TM predictions (e.g., syntactically correct precondition/effect sets)?
- **Environment consistency.** Does the candidate respect the provided object ontology, action schemas, and current scene (e.g., no non-existent objects or actions)?
- **Transition plausibility.** Are the predicted preconditions/effects plausible given example transitions in VIRTUALHOME (e.g., objects must be held before being put on)?
- **Non-hallucination.** Does the candidate avoid hallucinating objects, actions, or predicates that are not supported by the environment description?

The rubric is implemented as a natural-language prompt, with explicit definitions for each score level and concrete examples taken from the VIRTUALHOME TM data. The judge is instructed to (i) output per-criterion scores, (ii) set a hallucination flag whenever hallucinated content is detected, and (iii) provide a brief textual rationale that can be logged but is not used programmatically.

Repair. For this paper, we use a single-pass selection strategy: the planner generates $N = 4$ candidates once, the judge scores each candidate once, and the orchestrator selects a final output (Algorithm 1). We do not run iterative repair loops; this keeps the comparison between single-sample and multi-sample planner-judge settings straightforward and focuses the analysis on the effect of rubric-guided selection.

4 Implementation Details

Models. We implement both planner and judge roles using instruction-tuned LLMs accessed through a standard chat or completions API. The same backbone can be used for both planner and judge, or a slightly larger model can be reserved for judging; in our experiments we primarily use a shared backbone for simplicity. We treat the model as a black box and do not perform any additional fine-tuning.

Integration with EAI. We integrate our interface into the publicly released EAI VIRTUALHOME TM starter code [8]. For the single-sample baseline, we call the planner once per TM prompt ($N = 1$) and directly parse its output into preconditions and effects, which are then scored by the local EAI

Method	# planner samples	Choice accuracy (%)
Single-sample planner (no judge)	1	42.1
4-sample planner-judge (ours)	4	51.7

Table 2: Choice accuracy on the local VIRTUALHOME TM validation split.

TM evaluation script on the validation split. For the planner-judge variant, we (i) sample $N = 4$ candidates from the same planner, (ii) call the judge once per candidate with the rubric prompt, and (iii) apply Algorithm 1 to choose the final TM prediction that is passed to the scorer.

Decoding. Unless otherwise stated, we generate candidates with temperature $T \in [0.3, 0.7]$ and nucleus sampling with $p \in [0.9, 0.95]$ for the 4-sample planner-judge variant, and a lower temperature ($T \approx 0$) for the single-sample baseline to match typical deterministic decoding. Output lengths are capped based on expected structure (e.g., maximum number of precondition/effect lines per action).

Compute. Experiments are run on institutional GPU clusters and on George Mason University's Office of Research Computing (ORC) *Hopper* cluster. The planner-judge variant roughly multiplies inference cost by a factor of $N + N_{\text{judge}}$ relative to the single-sample baseline, where $N = 4$ planner calls and $N_{\text{judge}} = 4$ judge calls per TM prompt.

5 Evaluation

We evaluate the interface on the EAI VIRTUALHOME TM *local* validation split provided by the benchmark [8, 10]. We use the official TM scoring script in the starter repository, but we do not submit to the public leaderboard; all numbers are from local runs on the validation data.

Research questions. Our evaluation addresses two questions:
1. **Module performance:** How does rubric-guided 4-sample orchestration affect TM validation metrics compared to a single-sample planner baseline?
2. **Hallucination behavior:** How often does the judge flag hallucinations, and how does filtering based on these flags change TM outcomes on the validation set?

Metrics. We report the same TM matching metrics used in the EAI starter code [8], computed locally on the validation split. In particular, we report *choice accuracy*: the fraction of validation prompts for which the predicted precondition/effect set exactly matches the ground truth under the TM scorer. For qualitative analysis, we also manually inspect a subset of validation examples and categorize common hallucination patterns (e.g., non-existent objects, invalid actions, impossible effects).

Baselines and variants. We compare:
- **Single-sample planner (no judge).** One planner completion per TM prompt; outputs are parsed and scored directly.
- **4-sample planner-judge.** Four planner candidates per TM prompt, scored by the judge; the orchestrator filters hallucinated or low-scoring candidates and selects a final TM output (Algorithm 1).
- **4-sample planner (no judge).** As an ablation, we also consider a variant that samples four candidates but selects one uniformly at random, to isolate the effect of the judge relative to multi-sample decoding alone.

In all cases, we use the same planner backbone, prompts, and parsing logic.

Summary of findings. On the local VIRTUALHOME validation set, the single-sample planner baseline achieves 42.1% choice accuracy, while the 4-sample planner-judge variant reaches 51.7%

choice accuracy (Table 2). The 4-sample random-selection variant sits between these two and shows little or no consistent gain over the single-sample baseline. Qualitatively, the judge reliably filters candidates that (i) reference objects that are not present in the scene, (ii) use actions that are not valid for the given objects, or (iii) produce contradictory effects (e.g., setting an object to both `open` and `closed`). Residual errors tend to involve omissions (missing necessary preconditions/effects) rather than overt hallucinations, echoing broader observations about LLM hallucinations and uncertainty [7, 4, 11, **?**, **?**].

6 Analysis and Discussion

We analyze errors by hallucination type and scene, focusing on:

- **Residual hallucinations** that pass the judge (often subtle constraint violations or borderline cases where scene information is ambiguous).
- **Over-conservative filtering** where valid outputs are discarded because the judge is uncertain and assigns low rubric scores.
- **Structural modeling errors** that are hallucination-free but incomplete (e.g., missing preconditions) or poorly aligned with environment dynamics.

Explicit environment-conditioned rubric criteria reliably catch gross inconsistencies (non-existent objects, invalid actions), while the binary hallucination flag is useful for cheap filtering. Remaining failure modes often involve omissions (missing preconditions/effects) where hallucination-focused criteria are insufficient. In these cases, TM accuracy is limited more by the planner's modeling capacity than by hallucinations.

Our results suggest that even a simple rubric-guided judge, without any additional training or explicit uncertainty modeling, can substantially reduce obvious hallucinations in symbolic TM outputs and improve TM validation scores, at the cost of a small constant-factor increase in inference time. A natural next step is to combine rubric-guided judging with more principled uncertainty signals [4, 11, **?**, **?**] and to extend the interface beyond TM to other EAI modules.

7 Conclusion

We presented an inference-time, agent-as-judge interface for hallucination-aware Transition Modeling in the EAI benchmark. By separating proposal (planner agent) from audit (judge agent) and using rubric-based filtering and selection over a small committee of candidates, the system reduces obvious hallucinations and improves validation performance on VIRTUALHOME TM without any additional training. While our experiments focus on a simple 1-versus-4 comparison on the local EAI validation split, the planner-judge pattern and environment-aware rubrics apply broadly to structured LLM planning and can be extended to other embodied agent benchmarks and modules in future work.

Acknowledgments

We acknowledge the Office of Research Computing (ORC) at George Mason University for providing access to the Hopper cluster and associated research computing resources.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2023. Original version: arXiv:2204.01691.

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, et al. Rt-2:

Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183, 2023. Also available as arXiv:2307.15818.

[3] Datadog, Inc. Detect hallucinations in your rag llm applications with llm-as-a-judge. https://www.datadoghq.com/blog/llm-observability-hallucination-detection/, 2025. Accessed 2025-12-07.

[4] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

[5] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

[6] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Sergey Levine, Pieter Abbeel, Brian Ichter, and Karol Hausman. Inner monologue: Embodied reasoning through planning with language models. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, pages 1769–1782, 2023. Original version: arXiv:2207.05608.

[7] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Wenliang Dai, Ho Shu Chan, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 2023. Original version: arXiv:2202.03629.

[8] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Erran Li Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534, 2024.

[9] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, 2023. Also available as arXiv:2303.16634.

[10] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8494–8502, 2018.

[11] Ernesto Quevedo, Jorge Yero, Rachel Koerner, Pablo Rivas, and Tomas Cerny. Detecting hallucinations in large language model generation: A token probability approach. *arXiv preprint arXiv:2405.19648*, 2024.

[12] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, C. Karen Liu, Silvio Savarese, Hyowon Gweon, Jiajun Wu, and Li Fei-Fei. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, volume 164 of *Proceedings of Machine Learning Research*, pages 477–490, 2022.

A Biography of Team Members

Sam Blouir. Sam Blouir is a PhD student in Computer Science at George Mason University.

Buse Demir. Buse Demir is an AI Engineering PhD student and visiting researcher at Harvard University.