GRIDWM-JUDGE: EVALUATING VISION-LANGUAGE MODEL JUDGES IN GRID WORLDS VIA WORLD MODEL DEFICITS

Qinan Zhang[†], **Qihang Jin**[†]
University of Science and Technology of China

ABSTRACT

Vision–language models (VLMs) are increasingly used as automated judges to score agent trajectories, yet their success/fail verdicts are brittle to benign changes in wording, evidence order, or rendering. This instability may reflect a deeper world-model deficit: current VLM judges often lack reliable action-conditioned state tracking. We introduce **GridWM-Judge**, a diagnostic benchmark built on six deterministic MiniGrid environments that generates physically consistent Full/NoCue/Counterfactual trajectory triplets via deterministic planning and minimal interventions. It decomposes evaluation into three tasks: (A) 4-choice atomic next-observation prediction, (B) structured scene perception as canonical JSON, and (C) success/fail judging from rendered storyboards. We quantify reliability via Judgment Consistency Rate and Flip Rate under controlled framing, temporal, and visual-attribute probes, alongside accuracy and correlation analysis linking atomic prediction to judgment stability. Experiments across 13 VLMs reveal a fragility paradox: higher atomic transition accuracy does not necessarily yield more stable judgments, and can even correlate negatively with stability under temporal and visual probes. This reflects a decoupling between accuracy and robustness: weaker models rely on near-constant default verdicts, while stronger models engage in sensitive state tracking that is brittle to non-semantic perturbations. Robustness failures cannot be captured by accuracy alone; GridWM-Judge diagnoses how world-model competence relates to judge reliability. Project repository: `https://github.com/Lucas-Jin-Qh/GridWM_Judge`

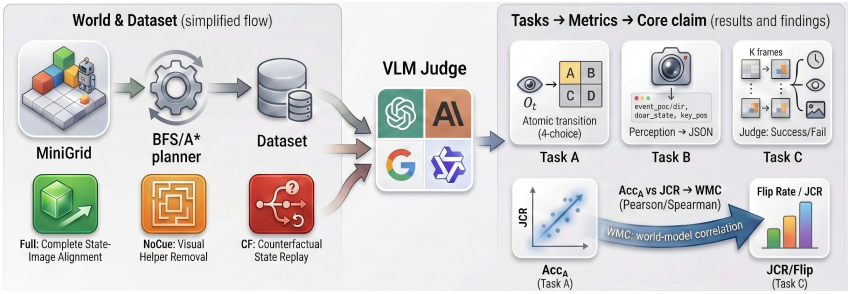


Figure 1: High-level overview of **GridWM-Judge**. **Left:** dataset generation from MiniGrid via deterministic planning producing Full / NoCue / Counterfactual variants (state/image/action aligned). **Center:** VLM judges evaluate diagnostic tasks. **Right:** Connecting atomic transition accuracy (Task A) to judgment consistency (Task C) through correlation analysis and stratified diagnostics.

1 INTRODUCTION

Vision–language models (VLMs) are increasingly deployed as automated evaluators in both natural language and embodied settings. Benchmarks such as MT-Bench and Chatbot-Arena use large

[†]Equal contribution. Contact: {qinanzhang,qihangjin}@mail.ustc.edu.cn

language models to grade conversational responses (Zheng et al., 2023), while Prometheus-Vision explores VLMs as open-vocabulary judges for multimodal tasks (Lee et al., 2024). These systems offer scalable alternatives to rule-based metrics and human annotations. Yet recent studies reveal that VLM judges are *fragile*: minor rephrasing, innocuous visual perturbations or simple reordering of frames can flip a model's success/failure verdict (Lee et al., 2024; Gao et al., 2025). Such instability undermines their reliability and raises concerns about deploying them in safety-critical settings. Existing analyzes often attribute this brittleness to sycophancy or prompt sensitivity (Zheng et al., 2023), implying that careful prompt engineering might suffice. We posit a deeper cause: current VLMs may lack reliable action-conditioned state tracking—they do not learn how low-level actions transform the environment over time. Without an action-conditioned forward model, a judge must rely on superficial cues and is easily swayed by rephrased prompts or irrelevant perceptual changes.

To understand this fragility, we revisit the concept of world models in reinforcement learning. Latent dynamics models such as PlaNet (Hafner et al., 2019b) and Dreamer (Hafner et al., 2019a) learn to imagine future states and support planning, while MuZero unifies dynamics, value and policy learning for complex games (Schrittwieser et al., 2020). These models provide foresight for agents, but VLM judges lack any mechanism to predict the consequences of actions. This leads us to a fundamental question: **Does a VLM's ability to predict atomic state transitions correlate with its robustness as a judge?** If a model can anticipate the next state after a given action, it may remain stable when assessing trajectories under varied framings or visual perturbations. Conversely, if it cannot predict simple transitions, its judgments may be random under diagnostic probes. To investigate this hypothesis, we need a benchmark that isolates world-model ability from perception and prompt bias and systematically probes judge stability under controlled interventions.

However, existing embodied benchmarks focus on agent performance rather than judge reliability. Lightweight environments like MiniGrid (Chevalier-Boisvert et al., 2023) and BabyAI (Chevalier-Boisvert et al., 2019) enable language-conditioned navigation and instruction following, while CLEVRER (Yi et al., 2020) and COVER (Zhou et al., 2025) assess causal and counterfactual reasoning in videos. These tasks evaluate whether agents can follow instructions or answer questions but do not diagnose why a VLM judge fails. Our goal is to provide a controlled, physically grounded testbed to probe the relation between world-model competence and judge stability.

Our Approach. We present *GridWM-Judge*, a diagnostic benchmark that reframes judge instability as a world-model deficit (Figure 1). Built on six MiniGrid environments, GridWM-Judge generates physically consistent success trajectories using deterministic planners, and creates matched hard-negative counterfactuals via minimal interventions on the environment. Each trajectory yields three complementary tasks: (i) a four-choice *Atomic Transition Prediction* task that asks the model to select the correct next frame conditioned on an action; (ii) a *Perception* task that converts a single egocentric frame into a structured scene description; and (iii) a *Judge Robustness* task that presents a storyboard and asks for a success/failure verdict under controlled framing, temporal and visual probes. Decoupling transition prediction, perception and judgment allows us to diagnose which capability drives instability.

Contributions. Our work makes three main contributions:

- We introduce **GridWM-Judge**, a diagnostic benchmark for studying world model deficits in VLM judges. It includes six MiniGrid tasks and physically consistent Full/NoCue/Counterfactual triplets. We report **1,600** *clean-only* exam trials across Tasks A/B/C, and expand Task C into **3,600** robustness inferences when evaluating visual-attribute variants (for a total of **4,000** model queries per model across all tasks).
- We design three tasks—Atomic Transition Prediction, Perception and Judge Robustness—and propose metrics such as the *Judgment Consistency Rate* (JCR), *Flip Rate* and *World-Model Correlation* (WMC) to quantify stability and link it to transition-prediction ability. We further analyze performance by transition type (move, turn, interact, pickup) to pinpoint specific deficits.
- We provide an open-source end-to-end pipeline for data generation, inference and scoring, and report results showing that robustness failures cannot be captured by accuracy alone. In our current cohort, higher Acc_A does not necessarily imply lower flip rates, highlighting a *fragility paradox* and motivating diagnostics that jointly report correctness and stability.

2 Related Work

VLM/LLM as Judges and Evaluators. LLMs/VLMs are increasingly used as automated judges for text and multimodal outputs. MT-Bench and Chatbot-Arena highlight both their utility and systematic biases (Zheng et al., 2023), and G-Eval studies prompt-dependent evaluation behavior (Liu et al., 2023). Recent VLM-judge settings (e.g., Prometheus-Vision; Agent-as-a-Judge) extend this paradigm to multimodal ranking and embodied trajectories (Lee et al., 2024; Zhuge et al., 2024). However, minor prompt rephrasings or benign visual changes can flip verdicts (Lee et al., 2024; Gao et al., 2025), motivating diagnostics that go beyond surface prompt sensitivity.

World Models for Reinforcement Learning and Multi-modal Prediction. World models are a core tool in model-based RL: PlaNet/Dreamer learn latent dynamics for imagination and planning (Hafner et al., 2019b;a), and MuZero unifies dynamics, value and policy learning (Schrittwieser et al., 2020). Subsequent work improves latent planning and efficiency (Hansen et al., 2024; Micheli et al., 2023) and scales action-conditioned prediction from video or embeddings (Bardes et al., 2023; Bruce et al., 2024; Parker-Holder et al., 2024). Recent directions include trajectory/world-model generalization and evaluation challenges (Yin et al., 2025; Zhao et al., 2025; Warrier et al., 2025). Meanwhile, studies probe whether VLMs exhibit implicit dynamics understanding (Gao et al., 2025; Spies et al., 2024; Zhang et al., 2025). Different from using world models to *act* or *generate*, we use world-model competence as a *diagnostic lens* for VLM judges by relating atomic transition prediction to verdict stability.

Embodied Benchmarks and Counterfactual Reasoning. Controlled embodied environments such as MiniGrid and BabyAI support grounded language and sequential decision-making (Chevalier-Boisvert et al., 2023; 2019), with extensions targeting action controllability or broader task coverage (Arai et al., 2024; Zhao et al., 2025). In vision, CLEVRER and COVER study causal/counterfactual video reasoning (Yi et al., 2020; Zhou et al., 2025), and WorldScore proposes metrics for comparing world-model generation (Duan et al., 2025). These benchmarks are largely agent- or QA-centric, and do not directly diagnose *judge* reliability under semantics-preserving perturbations. GridWM-Judge shifts the focus to VLMs as evaluators and introduces controlled framing/temporal/visual probes to measure verdict stability.

3 Our Approach: GridWM-Judge

Our goal is to diagnose whether a vision-language judge possesses reliable action-conditioned state tracking. We test this by measuring atomic transition prediction (Task A) and relating it to judgment stability under semantics-preserving probes (Task C). We instantiate this hypothesis in deterministic MiniGrid environments with physically consistent Full/NoCue/Counterfactual trajectory triplets, and quantify robustness via framing/temporal/visual probes preserving trajectory semantics.

3.1 World and Data Generation

We construct datasets on six MiniGrid environments—DoorKey, Memory, LavaGap, KeyCorridor, MultiRoom and RedBlueDoor—which provide deterministic gridworld dynamics and clear success criteria. For each environment we produce two types of trajectories:

Success Trajectories. Using BFS or A* planners, we generate optimal runs solving each task. Along each trajectory we record MiniGrid's symbolic egocentric observations (a $7 \times 7 \times 3$ tuple encoding object/color/state per tile), discrete action IDs, mission text and symbolic state sequences (agent position, orientation, carried object and visible states). For VLM-facing inputs, we render these symbolic grids into pixel-space egocentric RGB frames and storyboards before applying any visual-attribute probes. Each success trajectory yields one Task A instance (transition prediction) and one Task B instance (perception).

Hard Negative Counterfactuals. For every success trajectory we create a counterfactual variant via minimal interventions: (i) *Action errors*, where one to three key actions are altered and the remainder is rerolled; (ii) *Intervention forks*, where the environment is cloned at a prefix state and

a small modification (e.g., moving a key or locking a door) is applied, followed by rerolling with the original action sequence. Rerolling ensures the new observation sequence and symbolic states remain physically consistent. Each group thus contains a full success, a "NoCue" success (with early cues occluded) and a counterfactual failure. The dataset comprises 200 Task A items, 200 Task B items and 1,200 Task C items. For visual-attribute probes, each Task C item is rendered in three variants (`clean`/`noisy`/`style`), yielding 3,600 judgments for the visual probe family.

3.2 Task Design and Input Presentation

We design three complementary tasks that isolate different capabilities.

Task A: Atomic Transition Prediction. Given a current observation o_t and action a_t, the model sees a 2×2 grid of candidate next observations $\{o_{t+1}^{(1)}, \ldots, o_{t+1}^{(4)}\}$ and must select the correct one. Negatives are sampled from the same environment and action type but come from different state contexts, preventing simple pixel matching heuristics. High accuracy on this task indicates a learned mapping $f(o_t, a_t) \approx o_{t+1}$, i.e., a rudimentary world model.

Task B: Perception. From a single egocentric frame the model must output a canonical JSON containing the agent's (x, y) position, orientation, carried object (if any), the front cell's contents and a list of all visible objects (type, color, state) sorted lexicographically by type and position. This task measures visual parsing and spatial reasoning: errors here explain a portion of Task A or C failures. Only a single ground-truth frame per trajectory is used.

Task C: Judge Robustness. For each trajectory group we sample $K = 8$ key frames (using environment-specific heuristics) and present them as a storyboard (4×2 grid) along with the full action sequence and mission text. The model must output "Success" or "Fail". We consider three diagnostic probe families: (i) *Framing*—positive, neutral or negative preambles that should be semantically irrelevant; (ii) *Temporal*—comparing the original vs. reversed ordering of frames; (iii) *Visual*—mild deterministic perturbations applied in pixel space after rendering, implemented as a clean view, a noisy view with Gaussian pixel noise, and a style-adjusted view (contrast/brightness/sharpness tweaks). The "NoCue" variant removes early cues (e.g., obscures the key) while keeping outcomes unchanged; the counterfactual variant changes the outcome via interventions. This task tests whether the judge bases its verdict on causal understanding or superficial cues.

3.3 Evaluation Metrics

To relate world-model ability and judgment stability, we compute several metrics. These metrics collectively characterize whether a model is (i) accurate and stable, (ii) incorrect yet stable predictions, (iii) random, or (iv) sensitive to specific dynamics.

Task A and B Accuracies. Acc_A and Acc_B—the fraction of correct answers in Tasks A and B. For Task B we require exact JSON matches.

Judgment Consistency Rate (JCR). For each 3-way probe family $F \in \{\text{framing}, \text{visual}\}$, we define

$$\mathrm{JCR}_F = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\big(\hat{y}_i^{(1)} = \hat{y}_i^{(2)} = \hat{y}_i^{(3)}\big). \tag{1}$$

For temporal probes we use a 2-way consistency rate:

$$\mathrm{JCR}_{\mathrm{temp}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\big(\hat{y}_i^{\mathrm{orig}} = \hat{y}_i^{\mathrm{rev}}\big). \tag{2}$$

The corresponding flip rate is $1 - \mathrm{JCR}_F$ for 3-way probes and $1 - \mathrm{JCR}_{\mathrm{temp}}$ for temporal probes. For the visual probe, $(1), (2), (3)$ correspond to predictions on the `clean`, `noisy`, and `style` versions of the same base slice; we denote this specifically as $\mathrm{JCR}_{\mathrm{vis}}$.

Judge Accuracy. $JAcc_{\mathrm{neu}}$—the accuracy of success/failure predictions under the neutral, original condition. This penalizes "stably wrong" judges.

Near-Random Criterion. We classify a model as near-random if $Acc_A \leq 0.28$ or its 95% Wilson confidence interval upper bound is ≤ 0.30. We use this tag for stratified analysis.

World-Model Correlation (WMC). Aggregating scores by model or environment, we compute the Pearson correlation between Acc_A and JCR_F:

$$\mathrm{WMC} = \frac{\sum_i (\mathrm{Acc}_A^i - \overline{\mathrm{Acc}}_A)(\mathrm{JCR}_F^i - \overline{\mathrm{JCR}}_F)}{\sqrt{\sum_i (\mathrm{Acc}_A^i - \overline{\mathrm{Acc}}_A)^2} \sqrt{\sum_i (\mathrm{JCR}_F^i - \overline{\mathrm{JCR}}_F)^2}}. \tag{3}$$

WMC summarizes the association between transition prediction and consistency (we report both Pearson and Spearman).

Transition-Specific Deficits. Actions are categorized as move, turn, interact or pickup. We recompute Acc_A, JCR_F and $JAcc_{\mathrm{neu}}$ for each category to identify which dynamics contribute most to instability.

3.4 Evaluation Pipeline

Our evaluation proceeds in four stages:

1. **Exam Generation**: From raw trajectories we create Task A/B/C items and log dataset statistics in a manifest file. Task C is then queried under multiple probe conditions; for visual probes, this corresponds to three renderings per base instance.
2. **Model Inference**: We query each VLM with fixed token budgets to ensure fairness across models and providers. Responses are stored in JSONL format.
3. **Audit**: We check outputs for format compliance (strict vs. recoverable) to separate formatting errors from reasoning errors.
4. **Scoring**: Using the metrics above, we compute per-task and per-environment scores, failure histograms and WMC statistics; we also log example errors for qualitative analysis.

4 Experiments

We evaluate the proposed benchmark on a suite of controlled MiniGrid environments and use it to probe whether (and how) a model's *atomic transition competence* relates to the *stability of its success/failure judgments* under carefully designed perturbations. We report results in the natural task order: Task A (atomic transition prediction), Task B (structured perception), and Task C (judgment robustness). We close with a focused comparison of Task B in zero-shot versus two-shot prompting, which provides a clean separation between *format compliance* and *semantic understanding*.

4.1 Evaluation Setup

Environments and Exam Construction. We evaluate on six deterministic MiniGrid tasks: DoorKey, Memory, LavaGap, KeyCorridor, MultiRoom, and RedBlueDoor. Using physically con-

Table 1: Exam composition across environments.

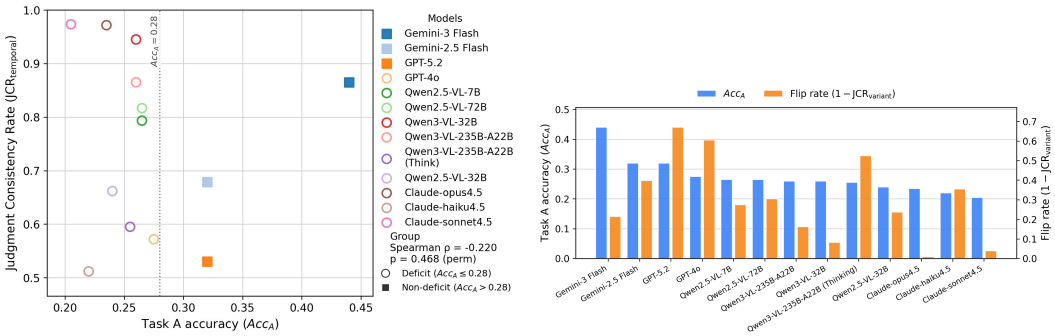
Environment	Task A items	Task B items	Task C (base)	#Judgments	
			(F×T)	F/T base[†]	V[‡]
DoorKey	20	20	120	120	360
Memory	40	40	240	240	720
LavaGap	40	40	240	240	720
KeyCorridor	40	40	240	240	720
MultiRoom	40	40	240	240	720
RedBlueDoor	20	20	120	120	360
Total	200	200	1,200	1,200	3,600

[†]**Base judgments for framing/temporal probes**: each base item is evaluated under 3 framing × 2 temporal variations (6) with clean rendering only.

[‡]**Visual judgments**: the same base item (6 combinations) is further evaluated across three rendering variants (`clean`, `noisy`, `style`); i.e., 6 × 3 per base group.

Table 2: **Per-model aggregate metrics.** Each row reports a VLM's competence on the three tasks. Acc_A and Acc_B are accuracies on Tasks A and B (higher is better). $J\mathrm{Acc}_{\mathrm{neu}}$ is the judge accuracy under the neutral/original condition in Task C. $\mathrm{JCR}_{\mathrm{frame}}, \mathrm{JCR}_{\mathrm{temp}}, \mathrm{JCR}_{\mathrm{vis}}$ are the judgment consistency rates under framing, temporal and visual probes, respectively (higher is more stable).

Model	$\mathbf{Acc}_A$	$\mathbf{Acc}_B$	$\boldsymbol{J}\mathbf{Acc}_{\mathbf{neu}}$	$\mathbf{JCR}_{\mathbf{frame}}$	$\mathbf{JCR}_{\mathbf{temp}}$	$\mathbf{JCR}_{\mathbf{vis}}$
Gemini-3 Flash	0.440	0.155	0.895	0.785	0.865	0.782
GPT-5.2	0.320	0.141	0.515	0.330	0.530	0.308
Gemini-2.5 Flash	0.320	0.186	0.280	0.603	0.678	0.632
GPT-4o	0.275	0.152	0.695	0.395	0.572	0.444
Qwen2.5-VL-72B	0.265	0.154	0.345	0.695	0.817	0.657
Qwen2.5-VL-7B	0.265	0.149	0.395	0.725	0.793	0.754
Qwen3-VL-235B-A22B	0.260	0.158	0.005	0.838	0.865	0.919
Qwen3-VL-32B	0.260	0.141	0.000	0.917	0.945	0.950
Qwen3-VL-235B-A22B(Think)	0.255	0.188	0.695	0.475	0.595	0.512
Qwen2.5-VL-32B	0.240	0.140	0.505	0.762	0.662	0.757
Claude-opus4.5	0.235	0.142	0.980	0.990	0.972	0.976
Claude-haiku4.5	0.220	0.140	0.225	0.645	0.512	0.667
Claude-sonnet4.5	0.205	0.140	0.000	0.960	0.973	0.981



(a) Acc_A vs. JCR correlation (model-level). **(b)** Per-model Acc_A and flip rate $(1 - \mathrm{JCR})$.

Figure 2: **World-model deficits vs. judge robustness.** **(a)** Scatter of atomic transition accuracy Acc_A against a representative robustness metric ($\mathrm{JCR}_{\mathrm{temporal}}$; higher is more stable) across models, illustrating our central finding that the relationship is non-monotonic and can be negative in the evaluated cohort. **(b)** Complementary per-model view showing Acc_A alongside flip rate $(1 - \mathrm{JCR})$, highlighting how "strong" and "weak" models distribute differently in stability even when their transition competence differs.

sistent Full/NoCue/Counterfactual triplets, we instantiate the three diagnostic tasks defined in Section 3. By default, the released exam contains 200 Task A items, 200 Task B items, and 1,200 Task C items (balanced across environments; Table 1). For the visual-attribute probe family, each Task C item is additionally evaluated on three deterministic renders (`clean`/`noisy`/`style`), yielding 3,600 judgments per model.

Models and Inference Budgets. We evaluate a cohort of 13 VLMs covering multiple model families (e.g., Qwen-VL variants, Gemini variants, GPT variants, and representative baselines). To control for response truncation and to keep comparisons consistent across providers, we use fixed generation budgets: 512 tokens for Task A, 2,048 tokens for Task B, and 256 tokens for Task C. All models are queried with temperature 0 (or the closest deterministic setting available per provider) and a single sample per item to minimize sampling variance.

Pipeline and Metrics. Each run follows a uniform pipeline: exam generation, model inference, audit (format vs. semantic failures), and scoring. We report Acc_A, Acc_B, and the neutral correctness anchor $JAcc_{\mathrm{neu}}$, together with robustness metrics $\mathrm{JCR}_{\mathrm{frame}}$, $\mathrm{JCR}_{\mathrm{temp}}$, and $\mathrm{JCR}_{\mathrm{vis}}$ (Table 2). We additionally analyze how Acc_A relates to robustness across models (Figure 2).

Table 3: **Task B audit summary.** Strict parse rate measures exact JSON schema conformance. Recoverable parse rate counts outputs parsable with minor corrections. Acc_B measures semantic correctness after parsing. Model names are colored by top failure type.

Model	Strict	Recov.	Acc_B
Claude-opus4.5	1.000	1.000	0.142
GPT-4o	1.000	1.000	0.152
GPT-5.2	1.000	1.000	0.141
Qwen3-VL-235B-A22B (Think)	1.000	1.000	0.188
Claude-haiku4.5	0.000	1.000	0.140
Claude-sonnet4.5	0.000	1.000	0.140
Qwen3-VL-235B-A22B	0.975	0.975	0.158
Qwen2.5-VL-32B	0.000	0.975	0.140
Qwen3-VL-32B	0.965	0.965	0.141
Qwen2.5-VL-72B	0.890	0.890	0.154
Qwen2.5-VL-7B	0.880	0.880	0.149
Gemini-2.5 Flash	0.530	0.605	0.186
Gemini-3 Flash	0.280	0.375	0.155

Color legend: red = Missing Key; yellow = Invalid JSON; blue = Others.

4.2 Main Results on Task A: Atomic Transition Prediction

Task A evaluates a minimal but consequential capability: given (o_t, a_t), choose the correct o_{t+1} among four candidates. This setting is intentionally local—there is no long-horizon planning—so failures are difficult to attribute to search or credit assignment and instead reflect shortcomings in action-conditioned dynamics modeling and object-state bookkeeping.

Across the evaluated models, Task A performance clusters only modestly above the 25% random baseline, with a noticeable spread between weaker and stronger models (Table 10 in Appendix). For example, the strongest model in our cohort reaches approximately 0.44 accuracy on the 200-item Task A split (Wilson 95% CI), while several other strong general-purpose VLMs remain closer to 0.32; multiple models (including larger variants within the same family) fall near 0.20–0.28.

Qualitatively, the most persistent errors arise on transitions that require updating *discrete latent state* (e.g., carrying status, door toggles, key pickup), rather than purely geometric motion. This is consistent with the benchmark's design intent: the environments are small, but they force correct handling of interaction-induced state changes. We provide representative failure cases in Appendix E.

4.3 Main Results on Task B: Structured Perception from a Single Frame

Task B asks the model to output a canonical JSON description capturing agent pose/orientation, the cell in front of the agent, carrying status, and visible objects. This task serves two roles. First, it measures whether a model can produce a machine-checkable representation of its visual understanding. Second, it provides a control signal when analyzing Task C: if a model cannot reliably parse a frame, downstream judgment instability may be dominated by perception rather than dynamics or reasoning.

In the zero-shot setting, Task B is dominated by *schema and formatting failures* for many models under strict parsing, leading to near-zero measured Acc_B even when partial semantic content is present (Table 3). The audit stage makes this concrete by reporting both strict and recoverable parse rates, allowing us to distinguish "cannot follow the output contract" from "followed the contract but got the scene wrong."

Even when JSON is recoverably parsed, semantic errors are common in fields that require consistent orientation and object bookkeeping (e.g., agent direction, carried object state, and object lists under partial observability). These are precisely the variables that the benchmark later reuses to define controlled perturbations and counterfactuals. In this sense, Task B is not merely a formatting hurdle: it exposes real weaknesses in discrete scene-state extraction that plausibly interact with longer-horizon judgment.

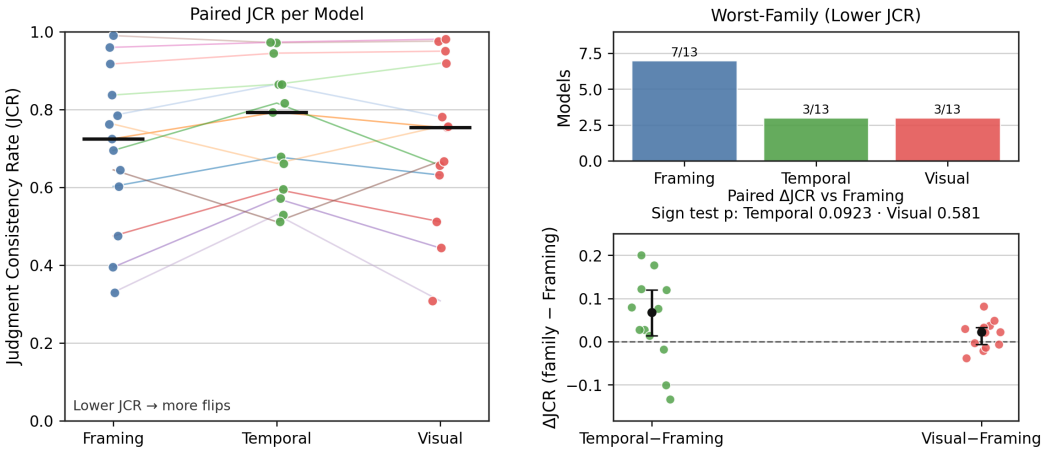


Figure 3: **Judgment consistency by probe family.** Top-right: count of models for which each family yields the lowest JCR. Bottom-right: paired ΔJCR (family - framing) with sign-test p-values. Boxplots summarize model-level JCR distributions under three probe families. The right panels show that framing is the worst-performing family for 7/13 models, while temporal and visual probes show wider variance; the paired ΔJCR analysis (bottom right) indicates that no single family dominates across all models.

4.4 Main Results on Task C: Robustness of Success/Failure Judgments

Task C evaluates whether a model's success/failure judgment is stable under benign transformations of the same underlying trajectory. We instantiate three probe families: (i) *framing* (changing the prompt tone while preserving semantics), (ii) *temporal* perturbations (e.g., re-ordering storyboard evidence), and (iii) *visual attribute* perturbations (clean vs. noisy vs. style-adjusted renders). Robust models should retain consistent verdicts across these probes because the trajectory semantics do not change. Per-variant accuracy decomposition (Full/NoCue/CF) is reported in Table 12 (Appendix D), confirming NoCue trajectories preserve the same outcome label while removing early visual cues.

We observe substantial heterogeneity in judgment stability across models (Table 12): some models exhibit high consistency (high JCR / low Flip), while others flip frequently. Importantly, the dominant sources of instability are typically not mild linguistic framing effects; rather, temporal and visual attribute probes tend to induce larger drops in consistency, suggesting that many VLM judges rely on brittle temporal heuristics or superficial visual cues rather than constructing a coherent causal account of the interaction sequence (Figure 3). Supplementary Table 13 shows that even when clean-slice accuracy is comparable, models' $\mathrm{JCR}_{\mathrm{vis}}$ can vary widely—from nearly invariant to highly sensitive—highlighting the need to report robustness jointly with correctness (Here N counts base slices after expanding rollout variants and temporal orders).

A Counterintuitive Cross-model Trend. A central goal of our approach is to test whether stronger atomic transition modeling predicts more stable judging. Empirically, however, we find that the relationship between Acc_A and judgment stability is not uniformly positive in our current cohort. In particular, the model-level Spearman correlations between Acc_A and stability measures are consistently *negative* across probe families ($\rho = -0.530, p = 0.066$ for $\mathrm{JCR}_{\mathrm{vis}}$; $\rho = -0.483, p = 0.097$ for $\mathrm{JCR}_{\mathrm{semantic}}$; $\rho = -0.220, p = 0.470$ for $\mathrm{JCR}_{\mathrm{temp}}$; Table 14), though none reach conventional significance in this 13-model cohort. A simple group analysis labeling models with $Acc_A \leq 0.28$ as "near-random" further shows that the higher-Acc_A group can even have *higher* flip rates on average (Table 18 in Appendix). At the same time, when we account for dependence structure (e.g., repeated measures across trajectories/models) using mixed-effects modeling, statistical significance weakens.

We interpret this as evidence that robustness is not a monotone function of local transition accuracy: more capable models may be more *sensitive* to perturbed evidence, while weaker models can appear stable by relying on simpler default strategies.

4.5 NOTABLE GAINS: TASK B (ZERO-SHOT VS. TWO-SHOT)

Task B is uniquely sensitive to the output interface: in zero-shot prompting, many models fail before semantics can even be evaluated because they do not reliably emit contract-compliant JSON. To isolate "format" from "content," we introduce a minimal two-shot prompt that includes two schema-faithful examples covering common corner cases (e.g., carrying vs. not carrying). This change produces a large improvement in strict parse success (Table 11 in Appendix), enabling meaningful semantic scoring at scale.

Crucially, two-shot prompting does not "solve" perception; rather, it removes a dominant confound. After the format barrier is reduced, remaining errors concentrate on precisely the fields that require discrete state extraction and consistent orientation. Consequently, the zero-shot vs. two-shot comparison serves as a diagnostic tool: it tells us whether a model is failing due to an interface mismatch or due to genuine limitations in scene understanding. We use this cleaned Task B signal in subsequent analyzes to ensure that conclusions about judgment robustness are not artifacts of malformed outputs.

4.6 DISCUSSION

GridWM-Judge is built around a simple Single Source of Truth (SSOT) principle: *if the underlying trajectory semantics are unchanged, a reliable judge should not change its verdict*. Our experiments suggest that today's general-purpose VLMs can violate this principle in embodied settings, and the failure cannot be explained away by prompt effects alone.

What drives brittle judging? Task A shows that even in compact MiniGrid worlds, **atomic, action-conditioned state tracking remains a bottleneck**, especially for interaction-induced updates (pickup/toggle/carrying). Task B further reveals a practical confound: in zero-shot settings, many apparent perception failures are in fact **interface failures** (schema/format), which can dominate strict scoring. A minimal two-shot schema largely removes this barrier, enabling more faithful measurement of scene-state extraction. Together, these results indicate that brittleness is often rooted in failures to maintain and update discrete latent state, not merely in superficial prompt sensitivity. A full conditional analysis ($Acc_A \mid Acc_B$) is left to future work; however, we note that Task B errors concentrate on orientation and object-state fields, which are also the primary failure modes in Task A (Section 4.2), suggesting shared perceptual bottlenecks.

Why may transition accuracy not monotonically predict robustness? Task C shows that **judgment robustness is not a corollary of local predictive skill**: within our cohort, higher Acc_A does not reliably yield higher consistency under semantics-preserving probes (framing, temporal order, and visual attributes), and the aggregate association can even be negative. A plausible reading is that stronger models may be more *responsive* to perturbed evidence, while weaker models can appear stable by falling back to coarse default heuristics—highlighting why robustness must be reported jointly with correctness.

Overall, our results argue for evaluating VLM-as-judge systems along three axes—**dynamics (A), perception (B), and stability (C)**—and for treating **flip rates under benign probes** as a first-class failure mode alongside $JAcc_{\text{neu}}$. Under the SSOT principle, consistency is not an optional nice-to-have; it is a minimal requirement for a judge to be trustworthy.

5 CONCLUSION

We introduced **GridWM-Judge**, a diagnostic benchmark that reframes VLM judge brittleness as a *world-model deficit*. By generating physically consistent trajectories and matched counterfactual failures, and decomposing evaluation into *Perception* (Task B), *Atomic Transition Prediction* (Task A), and *Judge Robustness* (Task C), our benchmark isolates which capabilities drive unstable judgments. Crucially, we find that robust judging is not guaranteed by strong transition prediction alone: while weak Acc_A coincides with erratic verdicts, higher accuracy does not reliably imply stability under controlled probes. We release an end-to-end pipeline for dataset generation, inference, and scoring, and hope GridWM-Judge facilitates future work on VLM judges that rely on causal, action-conditioned reasoning rather than superficial cues.

REFERENCES

Hidehisa Arai, Keishi Ishihara, Tsubasa Takahashi, and Yu Yamaguchi. ACT-bench: Towards action controllable world models for autonomous driving, 2024. URL https://arxiv.org/abs/2412.05337.

Adrien Bardes, Jean Ponce, and Yann LeCun. MC-JEPA: A joint-embedding predictive architecture for self-supervised learning of motion and content features, 2023. URL https://arxiv.org/abs/2307.12698.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. BabyAI: First steps towards grounded language learning with a human in the loop. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJeXCo0cYX.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & Miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Advances in Neural Information Processing Systems*, 2023.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. WorldScore: A unified evaluation benchmark for world generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2025.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *Advances in Neural Information Processing Systems*, 2022.

Qiyue Gao, Xinyu Pi, Kevin Liu, Junrong Chen, Ruolan Yang, Xinqi Huang, Xinyu Fang, Lu Sun, Gautham Kishore, Bo Ai, et al. Do vision-language models have internal world models? towards an atomic evaluation. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025. URL https://arxiv.org/abs/2506.21876.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2019a. URL https://arxiv.org/abs/1912.01603.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, 2019b.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Seongyun Lee, Seungone Kim, Sue Park, Geewook Kim, and Minjoon Seo. Prometheus-Vision: Vision-language model as a judge for fine-grained evaluation. In *Findings of the association for computational linguistics ACL 2024*, 2024.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-EVAL: NLG evaluation using gpt-4 with better human alignment, 2023. URL https://arxiv.org/abs/2303.16634.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris

Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model, 2024. URL https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Alex F Spies, William Edwards, Michael I Ivanitskiy, Adrians Skapars, Tilman Räuker, Katsumi Inoue, Alessandra Russo, and Murray Shanahan. Transformers use causal world models in maze-solving tasks, 2024. URL https://arxiv.org/abs/2412.11867.

Archana Warrier, Dat Nguyen, Michelangelo Naim, Moksh Jain, Yichao Liang, Karen Schroeder, Cambridge Yang, Joshua B Tenenbaum, Sebastian Vollmer, Kevin Ellis, et al. Benchmarking world-model learning, 2025. URL https://arxiv.org/abs/2510.19788.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: collision events for video representation and reasoning. In *International Conference on Learning Representations*, 2020.

Shaofeng Yin, Jialong Wu, Siqiao Huang, Xingjian Su, Xu He, Jianye Hao, and Mingsheng Long. Trajectory world models for heterogeneous environments, 2025. URL https://arxiv.org/abs/2502.01366.

Gengyuan Zhang, Mingcong Ding, Tong Liu, Yao Zhang, and Volker Tresp. Memory helps, but confabulation misleads: Understanding streaming events in videos with mllms, 2025. URL https://arxiv.org/abs/2502.15457.

Yi Zhao, Aidan Scannell, Yuxin Hou, Tianyu Cui, Le Chen, Dieter Büchler, Arno Solin, Juho Kannala, and Joni Pajarinen. Generalist world model pre-training for efficient reinforcement learning. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025. URL https://openreview.net/pdf?id=WtJnrr4BGO.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in neural information processing systems*, 2023.

Qiji Zhou, Yifan Gong, Guangsheng Bao, Hongjie Qiu, Jinqiang Li, Xiangrong Zhu, Huajian Zhang, and Yue Zhang. Reasoning is all you need for video generalization: A counterfactual benchmark with sub-question evaluation, 2025. URL https://arxiv.org/abs/2503.10691.

Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, et al. Agent-as-a-judge: Evaluate agents with agents, 2024. URL https://arxiv.org/abs/2410.10934.

A Appendix: Appendix Overview

This appendix complements the main paper with comprehensive implementation details and extensive data statistics, facilitating reproducibility and in-depth research. It is organized as: (A) overview, (B) task suite and evaluation design, (C) dataset schema and generation/validation protocols, (D) supplementary results, (E) qualitative examples, and (F) future work.

B Appendix: Task Suite and Evaluation Design

B.1 Triplets as the Atomic Evaluation Unit

GridWM-Judge uses a *triplet* as the atomic evaluation unit: ⟨**Full**, **NoCue**, **CF**⟩. A triplet shares the same `group_id`, environment, mission, and a fixed action sequence (`actions_id`/`actions_text`), and differs only in the rollout variant:

- **Full**: an oracle-success rollout produced by a deterministic planner.
- **NoCue**: a semantics-preserving evidence-removal rollout constructed from the same action sequence, where task-critical objects are tile-masked within a restricted window.
- **CF (counterfactual)**: a matched hard-negative rollout constructed by a minimal environment intervention at a fork point, followed by re-rollout under the unchanged actions.

The design goal is to keep the evaluation physically grounded and auditable: if a model flips its success/failure judgment under semantics-preserving probes (temporal order or visual attributes), this is treated as judge brittleness rather than as a change in underlying trajectory semantics.

B.2 SIX MINIGRID TASKS

Six Minigrid tasks are introduced as shown in Table 4.

Table 4: **Six-task overview.** Environment configuration, the semantic chain required for success, and the primary diagnostic focus of each task. All environments are from `MiniGrid`.

Task	Environment id	Core semantic chain / diagnostic focus
DoorKey	`DoorKey-8x8-v0`	Pickup key → unlock/toggle door → reach goal; tests discrete interaction state (carrying, door locked→open).
KeyCorridor	`KeyCorridorS6R3-v0`	Explore → find key → open locked door → pickup target; tests exploration and long-horizon bookkeeping with interaction.
RedBlueDoor	`RedBlueDoors-8x8-v0`	Toggle red door **before** blue door; tests ordered interaction constraints and temporal dependence.
LavaGap	`LavaGapS7-v0`	Find traversable gap → cross lava barrier → reach goal; stresses hazard-aware navigation and spatial dynamics.
Memory	`MemoryS13-v0`	Observe cue object → traverse corridor → choose matching terminal object; isolates memory under partial observability.
MultiRoom	`MultiRoom-N6-v0`	Open doors across rooms → reach goal; stresses long-horizon interaction and compositional progress tracking.

B.3 TASKS A/B/C AND INPUT PRESENTATION

We evaluate three complementary tasks to isolate different failure modes:

- **Task A (Atomic Transition Prediction)**: Given (o_t, a_t), select the correct o_{t+1} among four candidates (A/B/C/D).
- **Task B (Structured Perception)**: From a single frame, output a canonical JSON scene description (strict schema).
- **Task C (Judge Robustness)**: From a K-frame storyboard, a mission, and an action list, output exactly one token: `Success` or `Fail`.

B.4 EXAM COMPOSITION (ACTUAL)

The released exam contains 200 Task A items, 200 Task B items, and Task C built from **200 storyboard base cases**. Task C is expanded to **1,200 clean-only judgments** by crossing **semantic variants** (`full`/`nocue`/`cf`) with **temporal order** (`orig`/`rev`). Visual-attribute probes further expand Task C to **3,600 judgments** per model by adding `clean`/`noisy`/`style`.

B.5 SEMANTICS-PRESERVING PROBES VS. SEMANTIC VARIANTS

Task C contains two conceptually different axes:

- **Semantic variants**: `variant` $\in \{$`full`, `nocue`, `cf`$\}$. Here, `cf` is a hard failure by construction; invariance across semantic variants is not desirable, and a high "variant-consistency" can indicate degenerate behavior (e.g., always predicting '`Success`').

Table 5: **Exam composition (A): base-case distribution across environments.** N_{base} counts storyboard base cases used for balancing Task A/B and building Task C.

Environment	N_{base} (Task C storyboards)
DoorKey	20
Memory	40
LavaGap	40
KeyCorridor	40
MultiRoom	40
RedBlueDoor	20
Total	**200**

Table 6: **Exam composition (B): derived exam sizes in this release (per model).**

Item set	Count
Task A items	200
Task B items	200
Task C (clean-only; `variant`=3 × `temporal`=2)	$N_{\text{base}} \times 3 \times 2 = 1{,}200$
Task C (with visual probes; `visual`=3)	$N_{\text{base}} \times 3 \times 2 \times 3 = 3{,}600$
Total (base; A+B+C_clean)	**1,600**
Total (with visual probes; A+B+C_visual)	**4,000**

- **Semantics-preserving probes**: (i) **Temporal** reorders the same keyframes (`orig`/`rev`); (ii) **Visual attributes** change appearance without changing semantics (`clean`/`noisy`/`style`).

Accordingly, we report robustness via JCR_{temp} and JCR_{vis} and interpret them jointly with correctness anchors such as $J\text{Acc}_{\text{neu}}$ and per-variant accuracies(Acc_{full},Acc_{cf}).

B.6 Metrics (Definitions Used in this Repo Release)

Let each Task C *base slice* be indexed by $i = (\texttt{env\_task}, \texttt{group\_id}, \texttt{variant}, \texttt{temporal})$ and each *visual slice* by $i' = (\texttt{env\_task}, \texttt{group\_id}, \texttt{variant}, \texttt{temporal}, \texttt{visual})$. For a fixed model, let $\hat{y}(\cdot)$ denote its normalized prediction (`Success`/`Fail`) and $y(\cdot)$ the gold label.

- **Task A accuracy**: $Acc_A = \mathbb{E}[\mathbf{1}(\text{pred} = \text{gold})]$ over the 200 Task A items.
- **Task B semantic score**: mean component score over recoverably-parsed items.
- **Neutral correctness anchor**: $J\text{Acc}_{\text{neu}} = \mathbb{E}[\mathbf{1}(\hat{y}(\texttt{full,orig,clean}) = y(\texttt{full,orig,clean}))]$ over the 200 `full.orig.clean` storyboards.
- **Temporal consistency**: $\text{JCR}_{\text{temp}} = \mathbb{E}_i[\mathbf{1}(\hat{y}(i, \texttt{orig}) = \hat{y}(i, \texttt{rev}))]$ computed on the clean slice.
- **Visual consistency**: $\text{JCR}_{\text{vis}} = \mathbb{E}_i[\mathbf{1}(\hat{y}(i, \texttt{clean}) = \hat{y}(i, \texttt{noisy}) = \hat{y}(i, \texttt{style}))]$.
- **Semantic-variant consistency** (not a robustness goal): $\text{JCR}_{\text{semantic}} = \mathbb{E}_i[\mathbf{1}(\hat{y}(\texttt{full}) = \hat{y}(\texttt{nocue}) = \hat{y}(\texttt{cf}))]$ computed per $(\texttt{env\_task}, \texttt{group\_id}, \texttt{temporal})$ on the clean slice.

C Appendix: Dataset Schema and Generation/Validation Protocols

C.1 Released Artifacts and Roles (See Table 7)

C.2 Triplet JSONL Record Structure and Information Isolation

Each line in `jsonl` is a single trajectory-variant record (Table 8). Triplets are recovered by matching `group_id` across `variant` $\in \{\texttt{full}, \texttt{nocue}, \texttt{cf}\}$.

Table 7: **Artifacts and what they verify.**

Artifact	Role	What it verifies
Triplet JSONL	Data	Full/NoCue/CF alignment; model-input vs. judge-only field separation; per-step traces.
Exam JSONL	Ground truth	Exact question instances and gold answers (Tasks A/B) or gold labels (Task C).
Composite images	Model inputs	Stable input presentation via a single-image interface.
Task validators	Contract enforcement	Hard invariants and task-specific semantic rules (reproducible, executable checks).
Visual-probe generator	Dataset transform	Deterministic construction of `noisy` or `style` storyboards from `clean`.
Scoring and exports	Evaluation	Capability-preserving scoring; per-row CSV for analysis.
Stats and figures	Analysis	Model-level aggregates, plots, and statistical tests.

Table 8: **Key fields in a triplet record.**

Field	Category	Description
`group_id`, `task`, `env_id`, `seed`, `variant`	Both	Triplet identity and variant tag.
`frames`	Model-input	Relative paths to per-step RGB frames used to build Task C storyboards.
`mission`	Model-input	Environment mission string.
`actions_id`, `actions_text`	Model-input	Fixed action sequence (ids and text).
`terminated`, `truncated`	Model-input	Episode termination flags aligned to rollout.
`state_seq`	Judge-only (analysis)	Symbolic state trace aligned to frames (agent pose, front cell, object list, state encoding).
`success`, `reward`	Judge-only	Ground-truth outcome labels (used by Task C gold and validation gates).
`nocue_meta`	Judge-only (audit)	Masking window/targets/budget, alignment and physics checks, cue metadata.
`cf_meta`	Judge-only (audit)	Counterfactual intervention metadata (type, fork step, constraints).

C.3 Exam JSONL Structure (Tasks A/B/C)

- **Task A** (`datasets/exams/task_a_exam.jsonl`): composite image path and gold letter `answer` $\in \{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}\}$.
- **Task B** (`datasets/exams/task_b_exam.jsonl`): frame image and gold `answer_json` (plus a canonical serialized `answer` string).
- **Task C** (`datasets/exams/task_c_exam.jsonl`): storyboard image and gold binary label `label` $\in \{0, 1\}$ with `answer` $\in \{\texttt{Fail}, \texttt{Success}\}$.

C.4 MiniGrid Action Mapping (Table 9)

C.5 Visual-attribute Probes: `clean`/`noisy`/`style`

For each Task C storyboard, we provide three semantics-preserving visual variants: `clean` (original render), `noisy` (deterministic Gaussian pixel noise, $\sigma = 8.0$ in 8-bit RGB), and `style` (deterministic mild global appearance changes via contrast/brightness/sharpness adjustments).

Table 9: **MiniGrid action mapping (used throughout Task A and Task C).**

id	name	semantics
0	left	Turn left (agent rotates 90° counterclockwise).
1	right	Turn right (agent rotates 90° clockwise).
2	forward	Move forward by one cell if unblocked; otherwise no movement.
3	pickup	Pick up an object in the front cell (if pickable).
4	drop	Drop the carried object into the front cell (unused in our six tasks; kept for completeness).
5	toggle	Toggle/activate the object in the front cell (e.g., open/close/unlock doors).
6	done	No-op / unused in our tasks (kept for completeness).

C.6 Validation: Hard Gates and Semantic Checks

To minimize dataset bugs being misattributed to model failures, each task provides a strict validator that enforces hard invariants like:

- **Action identity**: `actions_id` is identical across Full/NoCue/CF for a given `group_id`.
- **Outcome gates**: Full must succeed; CF must fail (hard negative).
- **NoCue evidence removal (not ablation)**: masking is tile-local, within the allowed window, and targets task-critical evidence rather than blanking out the scene.
- **NoCue window/budget compliance**: masking occurs only within the task-defined window and respects a maximum mask ratio / budget.
- **Interaction preservation (NoCue)**: never mask frames that directly support interaction-critical evidence (e.g., the agent is facing/using the target object).
- **Physics consistency**: NoCue and CF preserve rollout consistency under the fixed actions (no silent dynamics drift).
- **State encoding validity**: symbolic encodings conform to MiniGrid conventions.

Task-specific semantic checks are also enforced (e.g., DoorKey requires key pickup before door toggle).

D Appendix: Supplementary Results

D.1 Task A Uncertainty (Wilson 95% CI, Table 10)

Table 10: **Task A accuracy with 95% Wilson confidence intervals.** The lower and upper bounds are computed over 200 Task A examples per model.

Model	$\mathbf{Acc}_A$	$\mathbf{CI}_L$	$\mathbf{CI}_U$
Gemini-3 Flash	0.440	0.373	0.509
GPT-5.2	0.320	0.259	0.388
Gemini-2.5 Flash	0.320	0.259	0.388
GPT-4o	0.275	0.218	0.341
Qwen2.5-VL-72B	0.265	0.209	0.330
Qwen2.5-VL-7B	0.265	0.209	0.330
Qwen3-VL-235B-A22B	0.260	0.204	0.325
Qwen3-VL-32B	0.260	0.204	0.325
Qwen3-VL-235B-A22B (Think)	0.255	0.200	0.320
Qwen2.5-VL-32B	0.240	0.186	0.304
Claude-opus4.5	0.235	0.182	0.298
Claude-haiku4.5	0.220	0.168	0.282
Claude-sonnet4.5	0.205	0.155	0.266

D.2 TASK B: ZERO-SHOT VS TWO-SHOT (FORMAT DE-CONFOUNDING, TABLE 11)

D.3 TASK C CORRECTNESS DECOMPOSITION (LABEL-BIAS FAILURE MODES)

Task C has a built-in semantic-variant imbalance: for each triplet group, `full` and `nocue` are labeled `Success` while `cf` is labeled `Fail` (2:1). Therefore correctness must be decomposed; otherwise degenerate policies can appear "robust" as shown in Table 12.

D.4 PROBE-FAMILY SUMMARY (MODEL-LEVEL)

To summarize robustness at the model level, we report three judge-consistency rates (JCRs) corresponding to our three probe families:

- **Semantic variants** (`full/nocue/cf`): JCR_{semantic} mean $= 0.702$ (min $= 0.330$, max $= 0.990$).
- **Temporal probes** (`orig/rev`, clean): JCR_{temp} mean $= 0.752$ (min $= 0.512$, max $= 0.973$).
- **Visual probes** (`clean/noisy/style`): JCR_{vis} mean $= 0.718$ (min $= 0.308$, max $= 0.981$).

The corresponding JCR-by-family plot is included in the main paper (Figure 3); Task B component scores are provided in Figure 4.

D.5 VISUAL-ATTRIBUTE PROBE SENSITIVITY

Even when `noisy/style` perturbations are visually subtle, many models change their verdicts frequently. We therefore report both *accuracy shifts* and *verdict flip rates*; the latter can be large even when net accuracy is unchanged (because flips can cancel out).

Definitions used in Table 13:

- $Acc_c/Acc_n/Acc_s$: Task C accuracy under `clean/noisy/style` (same base slices).
- JCR_{vis}: fraction of base slices whose predictions are identical across $\{\texttt{clean}, \texttt{noisy}, \texttt{style}\}$.
- $m(c{-}*)$: pairwise mismatch rate between two visual variants.
- $p(c \sim *)$: McNemar two-sided p-value for net accuracy change relative to `clean`.

Table 11: **Task B zero-shot vs two-shot comparison.** Strict and semantic scores are reported for both prompting strategies.

Model	Strict (0-shot)	Strict (2-shot)	Semantic (0-shot)	Semantic (2-shot)
Claude-haiku4.5	0.000	0.000	0.140	0.184
Claude-opus4.5	1.000	1.000	0.142	0.183
Claude-sonnet4.5	0.000	0.000	0.140	0.183
GPT-4o	1.000	1.000	0.152	0.181
GPT-5.2	1.000	0.995	0.141	0.185
Gemini-3 Flash	0.280	0.085	0.155	0.232[†]
Qwen2.5-VL-32B	0.000	0.990	0.140	0.184
Qwen2.5-VL-72B	0.890	1.000	0.154	0.176
Qwen2.5-VL-7B	0.880	1.000	0.149	0.165
Qwen3-VL-235B-A22B	0.975	1.000	0.158	0.184
Qwen3-VL-32B	0.965	0.985	0.141	0.172

[†] Gemini-3 Flash 2-shot outputs contain frequent truncated/invalid JSON (only 17/200 items were recoverably parsed in this rerun), so the semantic score is not directly comparable.

Table 12: **Task C decomposition (per model)** ($n(C) = 1,200$ clean-only trials per model; $n(C) = 3,600$ when expanding visual variants).

Claude-opus4.5

Acc: $Acc_C = 0.661$, $JAcc_{\text{neu}} = 0.980$
Acc(full/nocue/cf): 0.988 / 0.988 / 0.007
Succ(full/nocue/cf): 0.988 / 0.988 / 0.993
Verdict rate: $Succ_C = 0.990$

Gemini-3 Flash

Acc: $Acc_C = 0.644$, $JAcc_{\text{neu}} = 0.895$
Acc(full/nocue/cf): 0.887 / 0.922 / 0.123
Succ(full/nocue/cf): 0.887 / 0.922 / 0.861
Verdict rate: $Succ_C = 0.890$

GPT-4o

Acc: $Acc_C = 0.570$, $JAcc_{\text{neu}} = 0.695$
Acc(full/nocue/cf): 0.635 / 0.654 / 0.421
Succ(full/nocue/cf): 0.635 / 0.654 / 0.579
Verdict rate: $Succ_C = 0.623$

GPT-5.2

Acc: $Acc_C = 0.516$, $JAcc_{\text{neu}} = 0.515$
Acc(full/nocue/cf): 0.552 / 0.533 / 0.463
Succ(full/nocue/cf): 0.552 / 0.533 / 0.532
Verdict rate: $Succ_C = 0.539$

Qwen3-VL-235B-A22B (Think)

Acc: $Acc_C = 0.604$, $JAcc_{\text{neu}} = 0.695$
Acc(full/nocue/cf): 0.709 / 0.728 / 0.375
Succ(full/nocue/cf): 0.709 / 0.728 / 0.687
Verdict rate: $Succ_C = 0.708$

Claude-haiku4.5

Acc: $Acc_C = 0.452$, $JAcc_{\text{neu}} = 0.225$
Acc(full/nocue/cf): 0.464 / 0.474 / 0.417
Succ(full/nocue/cf): 0.464 / 0.474 / 0.583
Verdict rate: $Succ_C = 0.507$

Gemini-2.5 Flash

Acc: $Acc_C = 0.419$, $JAcc_{\text{neu}} = 0.280$
Acc(full/nocue/cf): 0.281 / 0.276 / 0.702
Succ(full/nocue/cf): 0.281 / 0.276 / 0.297
Verdict rate: $Succ_C = 0.285$

Qwen2.5-VL-72B

Acc: $Acc_C = 0.414$, $JAcc_{\text{neu}} = 0.345$
Acc(full/nocue/cf): 0.355 / 0.357 / 0.531
Succ(full/nocue/cf): 0.355 / 0.357 / 0.641
Verdict rate: $Succ_C = 0.451$

Qwen2.5-VL-7B

Acc: $Acc_C = 0.469$, $JAcc_{\text{neu}} = 0.395$
Acc(full/nocue/cf): 0.405 / 0.404 / 0.598
Succ(full/nocue/cf): 0.405 / 0.404 / 0.351
Verdict rate: $Succ_C = 0.387$

Qwen2.5-VL-32B

Acc: $Acc_C = 0.477$, $JAcc_{\text{neu}} = 0.505$
Acc(full/nocue/cf): 0.424 / 0.411 / 0.595
Succ(full/nocue/cf): 0.424 / 0.411 / 0.404
Verdict rate: $Succ_C = 0.413$

Qwen3-VL-235B-A22B

Acc: $Acc_C = 0.338$, $JAcc_{\text{neu}} = 0.005$
Acc(full/nocue/cf): 0.005 / 0.007 / 0.992
Succ(full/nocue/cf): 0.005 / 0.007 / 0.189
Verdict rate: $Succ_C = 0.067$

Qwen3-VL-32B

Acc: $Acc_C = 0.333$, $JAcc_{\text{neu}} = 0.000$
Acc(full/nocue/cf): 0.000 / 0.003 / 0.996
Succ(full/nocue/cf): 0.000 / 0.003 / 0.079
Verdict rate: $Succ_C = 0.027$

Claude-sonnet4.5

Acc: $Acc_C = 0.329$, $JAcc_{\text{neu}} = 0.000$
Acc(full/nocue/cf): 0.000 / 0.006 / 0.980
Succ(full/nocue/cf): 0.000 / 0.006 / 0.020
Verdict rate: $Succ_C = 0.009$

D.6 World-model Ability vs. Robustness/Correctness (correlation summary)

Using the 13 comparable models (excluding `Qwen3-VL-8B`), we summarize how world-model ability (Task A) relates to correctness/robustness (Task C) in Table 14.

The scatter plot used in the main paper (Figure 2) is omitted here to avoid duplication. Instead, we provide the mixed-effects model summary as Table 15 and Table 16.

D.7 Task A Transition-type Breakdown (Table 17, Figure 5)

D.8 Near-random Group Test

We define a practical near-random split as $\text{Acc}_A \leq 0.28$ and test differences in stability-like metrics at the model level in Table 18.

E Appendix: Qualitative Case Studies

We also provide compact, example-backed qualitative cases to connect the released model inputs (composite images and frames) to typical failure modes (Figure 6 $\sim$ Figure 11).

Table 13: **Visual-attribute probe sensitivity (per model)**.

Claude-opus4.5

Acc(c/n/s): 0.662 / 0.657 / 0.663 ($\Delta n = -0.006$, $\Delta s = +0.001$)
Consistency: $JCR_{\text{vis}} = 0.976$, $m(c{-}n) = 0.019$, $m(c{-}s) = 0.014$
Succ(c/n/s): 0.986 / 0.990 / 0.993
McNemar p: $p(c \sim n) = 0.210$, $p(c \sim s) = 1.000$

Gemini-3 Flash

Acc(c/n/s): 0.642 / 0.633 / 0.656 ($\Delta n = -0.009$, $\Delta s = +0.013$)
Consistency: $JCR_{\text{vis}} = 0.782$, $m(c{-}n) = 0.160$, $m(c{-}s) = 0.142$
Succ(c/n/s): 0.868 / 0.898 / 0.903
McNemar p: $p(c \sim n) = 0.457$, $p(c \sim s) = 0.233$

GPT-4o

Acc(c/n/s): 0.583 / 0.583 / 0.544 ($\Delta n = +0.001$, $\Delta s = -0.038$)
Consistency: $JCR_{\text{vis}} = 0.444$, $m(c{-}n) = 0.366$, $m(c{-}s) = 0.363$
Succ(c/n/s): 0.613 / 0.655 / 0.601
McNemar p: $p(c \sim n) = 1.000$, $p(c \sim s) = 0.031$

GPT-5.2

Acc(c/n/s): 0.523 / 0.497 / 0.527 ($\Delta n = -0.026$, $\Delta s = +0.004$)
Consistency: $JCR_{\text{vis}} = 0.308$, $m(c{-}n) = 0.466$, $m(c{-}s) = 0.452$
Succ(c/n/s): 0.529 / 0.494 / 0.595
McNemar p: $p(c \sim n) = 0.203$, $p(c \sim s) = 0.863$

Qwen2.5-VL-72B

Acc(c/n/s): 0.373 / 0.487 / 0.383 ($\Delta n = +0.113$, $\Delta s = +0.009$)
Consistency: $JCR_{\text{vis}} = 0.657$, $m(c{-}n) = 0.282$, $m(c{-}s) = 0.111$
Succ(c/n/s): 0.370 / 0.613 / 0.371
McNemar p: $p(c \sim n) = 9.7\text{e}{-14}$, $p(c \sim s) = 0.386$

Claude-haiku4.5

Acc(c/n/s): 0.422 / 0.496 / 0.438 ($\Delta n = +0.073$, $\Delta s = +0.015$)
Consistency: $JCR_{\text{vis}} = 0.667$, $m(c{-}n) = 0.255$, $m(c{-}s) = 0.142$
Succ(c/n/s): 0.476 / 0.561 / 0.484
McNemar p: $p(c \sim n) = 5.5\text{e}{-07}$, $p(c \sim s) = 0.192$

Gemini-2.5 Flash

Acc(c/n/s): 0.438 / 0.412 / 0.408 ($\Delta n = -0.025$, $\Delta s = -0.029$)
Consistency: $JCR_{\text{vis}} = 0.632$, $m(c{-}n) = 0.260$, $m(c{-}s) = 0.229$
Succ(c/n/s): 0.281 / 0.280 / 0.293
McNemar p: $p(c \sim n) = 0.100$, $p(c \sim s) = 0.0401$

Qwen3-VL-235B-A22B (Think)

Acc(c/n/s): 0.594 / 0.591 / 0.587 ($\Delta n = -0.003$, $\Delta s = -0.007$)
Consistency: $JCR_{\text{vis}} = 0.512$, $m(c{-}n) = 0.333$, $m(c{-}s) = 0.319$
Succ(c/n/s): 0.706 / 0.714 / 0.703
McNemar p: $p(c \sim n) = 0.881$, $p(c \sim s) = 0.683$

Qwen3-VL-32B

Acc(c/n/s): 0.329 / 0.315 / 0.327 ($\Delta n = -0.014$, $\Delta s = -0.003$)
Consistency: $JCR_{\text{vis}} = 0.950$, $m(c{-}n) = 0.034$, $m(c{-}s) = 0.028$
Succ(c/n/s): 0.028 / 0.023 / 0.030
McNemar p: $p(c \sim n) = 0.0115$, $p(c \sim s) = 0.728$

Qwen3-VL-235B-A22B

Acc(c/n/s): 0.318 / 0.312 / 0.322 ($\Delta n = -0.005$, $\Delta s = +0.004$)
Consistency: $JCR_{\text{vis}} = 0.919$, $m(c{-}n) = 0.053$, $m(c{-}s) = 0.049$
Succ(c/n/s): 0.068 / 0.069 / 0.063
McNemar p: $p(c \sim n) = 0.532$, $p(c \sim s) = 0.603$

Qwen2.5-VL-32B

Acc(c/n/s): 0.476 / 0.476 / 0.478 ($\Delta n = +0.000$, $\Delta s = +0.003$)
Consistency: $JCR_{\text{vis}} = 0.757$, $m(c{-}n) = 0.183$, $m(c{-}s) = 0.113$
Succ(c/n/s): 0.411 / 0.416 / 0.413
McNemar p: $p(c \sim n) = 1.000$, $p(c \sim s) = 0.863$

Qwen2.5-VL-7B

Acc(c/n/s): 0.400 / 0.413 / 0.407 ($\Delta n = +0.013$, $\Delta s = +0.007$)
Consistency: $JCR_{\text{vis}} = 0.754$, $m(c{-}n) = 0.183$, $m(c{-}s) = 0.135$
Succ(c/n/s): 0.410 / 0.368 / 0.382
McNemar p: $p(c \sim n) = 0.312$, $p(c \sim s) = 0.582$

Claude-sonnet4.5

Acc(c/n/s): 0.327 / 0.334 / 0.325 ($\Delta n = +0.008$, $\Delta s = -0.002$)
Consistency: $JCR_{\text{vis}} = 0.981$, $m(c{-}n) = 0.014$, $m(c{-}s) = 0.012$
Succ(c/n/s): 0.013 / 0.001 / 0.012
McNemar p: $p(c \sim n) = 0.049$, $p(c \sim s) = 0.791$

Table 14: **Correlation summary (model level).** We report Pearson r, Spearman ρ, and a permutation p-value.

Pair	**Pearson** r	**Spearman** ρ	**Perm.** p
Acc_A vs. JCR_{vis}	-0.271	-0.530	0.066
Acc_A vs. JCR_{temp}	-0.042	-0.220	0.470
Acc_A vs. $JAcc_{\text{neu}}$	+0.420	+0.320	0.282
Acc_A vs. JCR_{semantic}	-0.246	-0.483	0.097

Table 15: **Mixed-effects fixed effects (log-odds).**

Term	**Post. mean**	**Post. SD**	**Interpretable quantity**
Intercept	-1.5406	0.0338	Baseline flip probability at mean Acc_A: 0.176
Acc_A slope (centered)	0.4989	0.3632	Odds ratio per $+1.0$ in centered Acc_A: 1.647

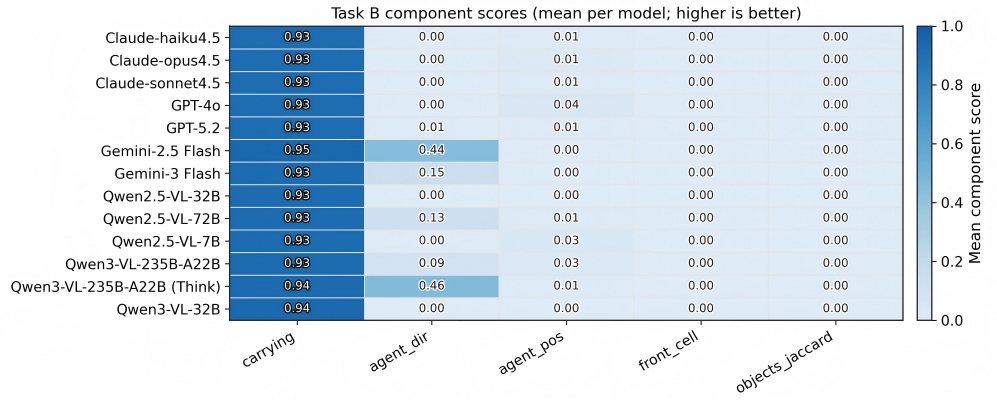


Figure 4: **Task B component scores.** Heatmap of mean Task B component scores per model, computed on recoverably-parsed outputs. This complements Table 3 by showing which semantic components fail (e.g., agent pose/orientation vs. object-set overlap).

Table 16: **Random intercept SDs (logit scale; posterior SD with 95% interval).** By model (13 levels) and by environment (6 levels).

By model			By environment		
Model	**SD**	**SD 95% CI**	**Environment**	**SD**	**SD 95% CI**
Gemini-2.5 Flash	0.855	[0.252, 2.897]	multiroom	1.260	[0.430, 3.691]
Gemini-3 Flash	1.012	[0.321, 3.197]	lavagap	1.626	[0.596, 4.435]
Qwen2.5-VL-7B	0.460	[0.087, 2.426]	doorkey	0.500	[0.103, 2.414]
Qwen2.5-VL-32B	0.594	[0.142, 2.480]	redblue	0.678	[0.177, 2.593]
Qwen2.5-VL-72B	0.498	[0.103, 2.414]	memory	0.481	[0.096, 2.415]
Qwen3-VL-235B-A22B	1.031	[0.329, 3.233]	keycorridor	0.444	[0.081, 2.443]
Qwen3-VL-235B-A22B (Think)	1.338	[0.465, 3.850]			
Qwen3-VL-32B	1.643	[0.604, 4.469]			
GPT-4o	1.598	[0.583, 4.379]			
GPT-5.2	1.786	[0.670, 4.762]			
Claude-haiku4.5	0.725	[0.197, 2.668]			
Claude-opus4.5	3.126	[1.309, 7.465]			
Claude-sonnet4.5	2.175	[0.852, 5.553]			

Table 17: **Task A accuracy by transition type**.

Transition type	n **(micro)**	**Acc (micro)**	**Acc (macro mean)**
Move	1924	0.275	0.275
Turn	442	0.260	0.260
Pickup	91	0.330	0.330
Interact	143	0.266	0.266

Table 18: **Near-random vs. other models.** Near-random models satisfy $\mathrm{Acc}_A \leq 0.28$. Flip rates refer to the primary stability probe $\mathrm{Flip}_{\mathrm{pri}} := 1 - \mathrm{JCR}_{\mathrm{vis}}$ (Section 3.3). Permutation p-value is 0.242 here.

Group	**#Models**	**Mean** $\mathbf{Acc}_A$	**Mean Flip**
Near-random	10	0.248	0.260
Other	3	0.360	0.427

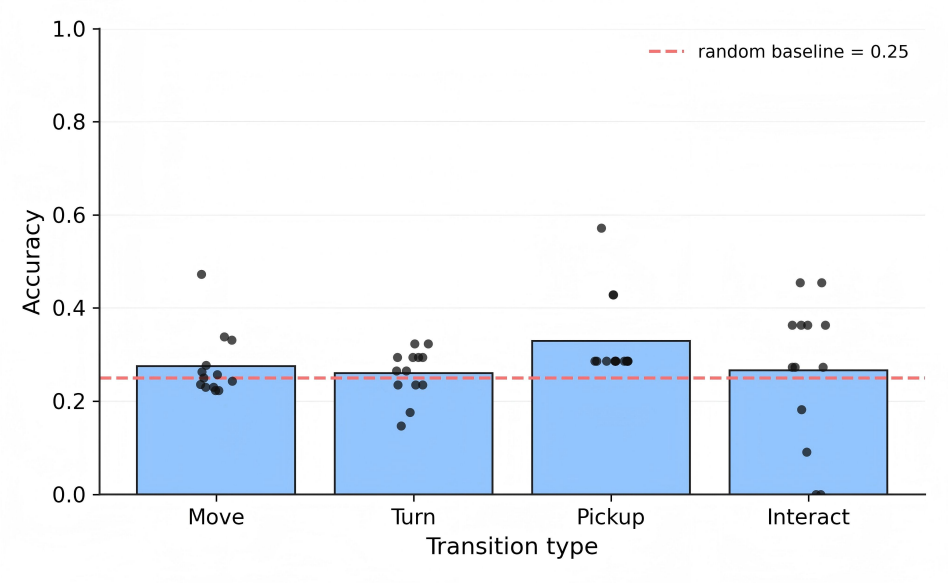
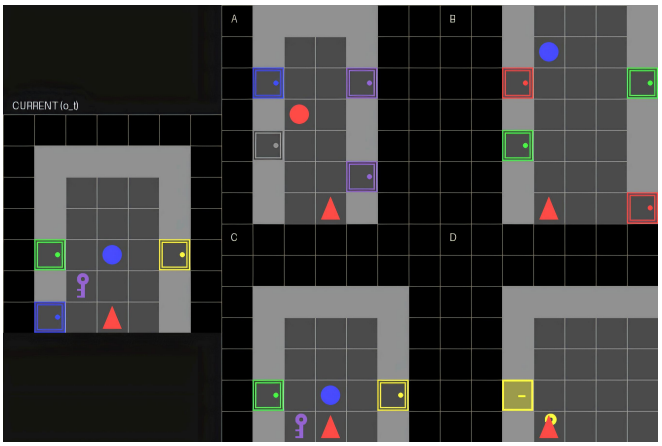


Figure 5: **Task A accuracy by transition type.** Accuracy differs by transition category, enabling mechanistic diagnosis.



Prompt (excerpt).
Action: forward (id=2).
Left: current state o_t.
Right: 2x2 candidates for o_{t+1} (A/B/C/D).
Answer with ONLY: A/B/C/D.
Exam JSON (abridged).
{
"exam_id": "A.keycorridor.keycorridor_s00100
"task": "A",
"env_task": "keycorridor",
"action": "forward (id=2)",
"image": "images/taskA/keycorridor/keycorrid
"answer": "C"
}

Figure 6: **Task A composite image template (example).** A single-image interface combines the current frame o_t (left) with four candidate o_{t+1} frames (right, labeled A/B/C/D).

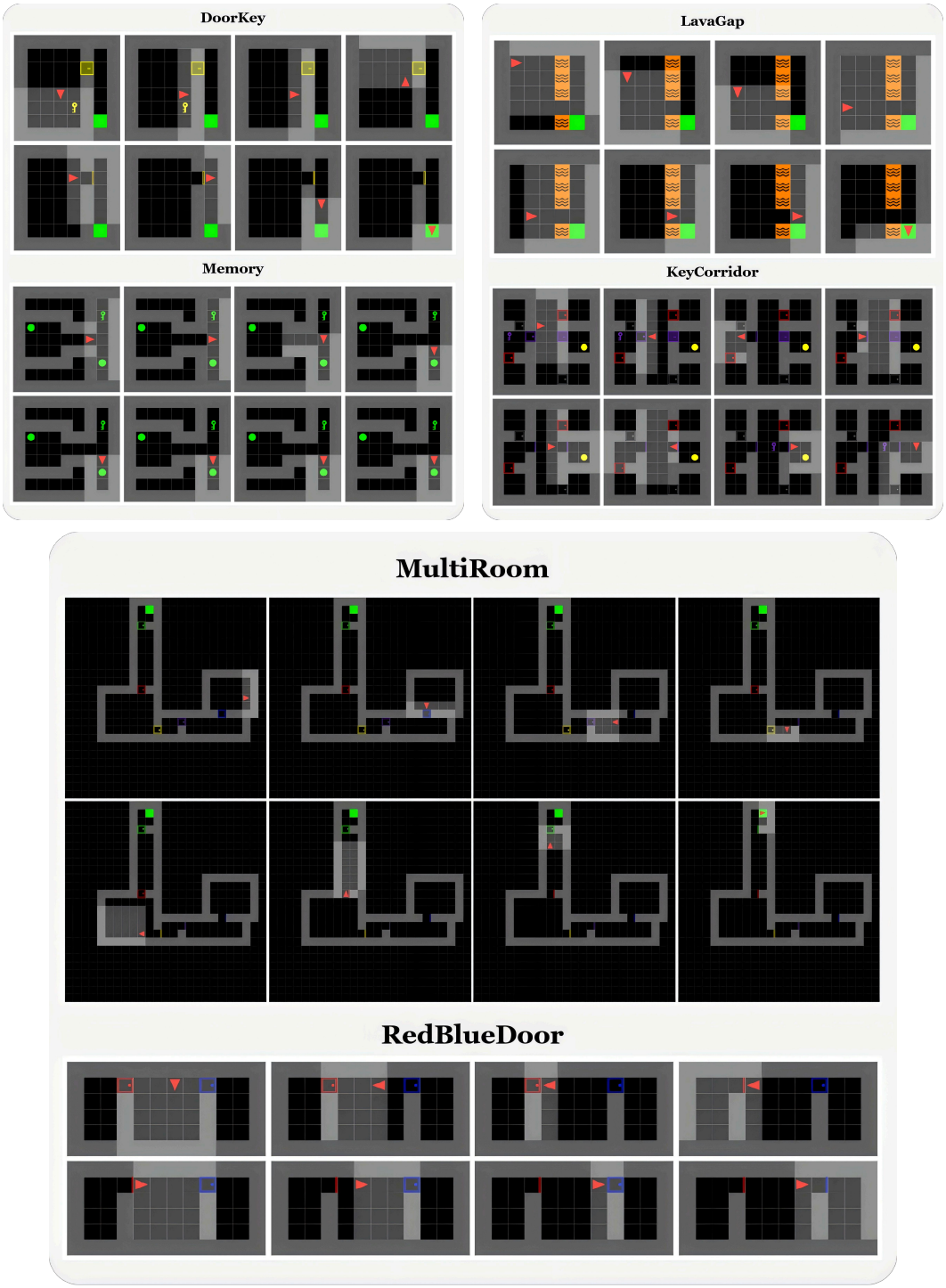
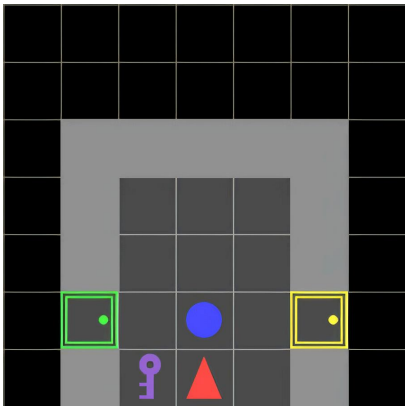


Figure 7: **Representative storyboards for the six tasks.** We show one Full-success storyboard per task, rendered in the same 4×2 storyboard layout (8 keyframes, $K{=}8$) as Task C. The three panels jointly cover *DoorKey, Memory* (left), *LavaGap, KeyCorridor* (right), and *MultiRoom, RedBlueDoor* (bottom).



Prompt (excerpt; 0-shot).
Return ONLY a JSON object with keys:
agent, front_cell, objects.
GRID: 7x7, [0,0] top-left, x->right, y->down.
Directions: 0=north, 1=east, 2=south, 3=west.
front_cell: dir=0->[x,y-1], 1->[x+1,y], 2->[x,y+1], 3->[x-1,y].
IMPORTANT: raw JSON only (no markdown / no explanation).
Gold answer json (abridged).
{
"agent": {"pos": [6,3], "dir": 0, "carrying": null},
"front_cell": {"pos": [6,3], "type": "ball", "state": 0},
"objects": [
{"type":"ball","pos":[5,3],"color":"blue","state":0},
{"type":"door","pos":[5,5],"color":"yellow","state":2},
{"type":"key","pos":[6,2],"color":"purple","state":0}]
}

Prompt (excerpt; 2-shot).
Return ONLY a JSON object with agent, front_cell, and objects keys.

GRID: 7x7 (x=0-7, y=0-7). [0,0] top-left, x→right, y↓down. Agent POV: faces UP in image.
Directions: 0=north↑, 1=east→, 2=south↓, 3=west←.

VISUAL DIRECTION CUES:
- If agent faces WALL/DOOR ahead: likely dir=0 (north) or dir=2 (south)
- If agent faces OPEN space ahead: check surroundings to determine direction
- Count grid cells: 7 across × 7 down
- Agent position anywhere in [0-7, 0-7]

Direction→front_cell: dir=0→[x,y-1]; dir=1→[x+1,y]; dir=2→[x,y+1]; dir=3→[x-1,y].

IMPORTANT: Return ONLY raw JSON, no markdown, no code blocks, no explanations.

Example 1 (agent facing south):
{
  "agent": {"pos": [5, 6], "dir": 2, "carrying": null},
  "front_cell": {"pos": [5, 7], "type": "floor", "state": 0},
  "objects": [
    {"type": "door", "pos": [3, 2], "color": "red", "state": 1},
    {"type": "goal", "pos": [7, 1], "color": "green", "state": 0}]
}
Example 2 (agent facing east):
{
  "agent": {"pos": [6, 4], "dir": 1, "carrying": {"type": "key", "color": "blue"}},
  "front_cell": {"pos": [6, 3], "type": "wall", "state": 0},
  "objects": [
    {"type": "door", "pos": [7, 6], "color": "yellow", "state": 0},
    {"type": "key", "pos": [2, 7], "color": "red", "state": 0}]
}

Figure 8: **Task B schema and example.** Task B requires a canonical JSON scene description from a single frame (top). We show prompt excerpts for both 0-shot (bottom left) and 2-shot (bottom right) settings, which specify the strict JSON-only contract and provide schema-faithful examples for format de-confounding.

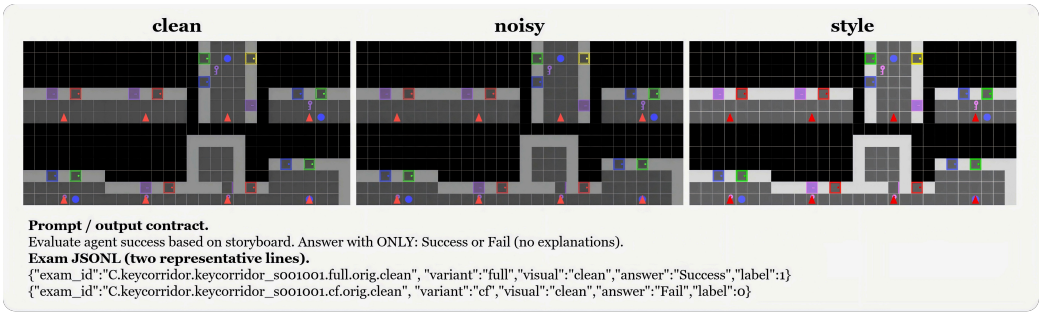
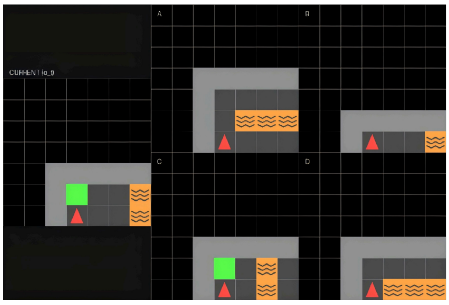
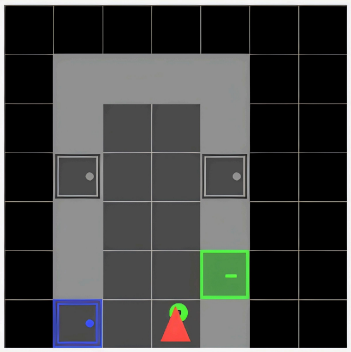


Figure 9: **Task C input and output contract (example).** Models judge success from a $K{=}8$ storyboard (shown under visual probes `clean/noisy/style`) and must output exactly one token in {`Success`, `Fail`}. The example illustrates how the JSONL schema ties each judgment to a `variant` (Full/NoCue/CF), `temporal` order, and `visual` rendering condition.



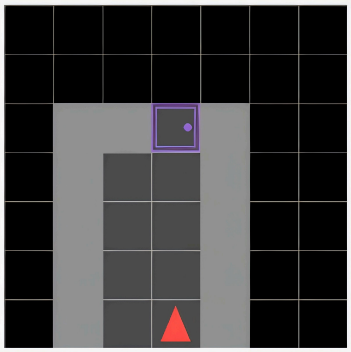
Gold (Task A exam; abridged).
Gold (from exam JSON):
uid: lavagap_s000065.t8
answer: **B**
Model output (error; abridged).
Model (from log JSON):
uid: lavagap_s000065.t8
pred prefix:"**A**:** Agent is at""(row 6, col 2)..."

Figure 10: **Task A failure example (LavaGap).** Even when the visual interface is fixed, some models fail to follow the strict "single-letter" contract (A/B/C/D) and/or confuse the correct next-frame option under forward transitions.



KeyCorridor (Task B).

Gold:
agent.pos=[6,3], dir=0 (north)
objects: doors x4, key x1
Pred:
agent.pos=[10,7], dir=0 (north)
objects: box x1, goal x1



MultiRoom (Task B).

Gold:
agent.pos=[6,3], dir=0 (north)
objects: doors
Pred:
agent.pos=[10,14], dir=0 (north)
objects: boxes + goal

Figure 11: **Task B failure examples (schema/coordinate drift).** Typical errors include misinterpreting the global grid coordinate system and collapsing object categories, yielding structurally valid JSON but semantically incorrect scene states.

F APPENDIX: FUTURE WORK

We view GridWM-Judge as a first step toward principled diagnostics for VLM-as-judge reliability. We highlight several high-impact directions.

Scaling Beyond Deterministic Gridworlds. Our current benchmark uses small, deterministic MiniGrid environments to tightly control confounds. An important next step is to stress-test robustness in larger maps and partially stochastic dynamics, and eventually in continuous 3D simulators (e.g., MiniWorld(Chevalier-Boisvert et al., 2023) or lightweight 3D navigation benchmarks),or even open-world settings such as Minecraft-based embodied learning platforms(Fan et al., 2022). This would introduce long-horizon credit assignment, occlusion, and richer object interactions, enabling stronger tests of temporal reasoning and memory.

Improving Judges via Transition-centric Training. Since Task A measures atomic dynamics prediction, a natural direction is to fine-tune VLMs on transition data (or to attach an explicit latent dynamics module) and test whether improvements in Acc_A causally reduce flips in Task C. This directly evaluates whether world-model competence is a *cause* of judge robustness rather than a correlational artifact.

Human Ceiling and Ambiguity Auditing. To calibrate the attainable stability, we plan to collect human judgments on a subset of Task C under the same probes (framing, temporal order, and counterfactuals). This will (i) estimate an upper bound on JCR, and (ii) identify inherently ambiguous cases that should be excluded or separately analyzed.

Richer Causal Interventions and Explanations. Beyond minimal counterfactual edits, future versions can parameterize interventions with explicit causal variables (e.g., key location $\rightarrow$ door unlockability $\rightarrow$ goal reachability) and require judges to output short rationales or causal attributions, enabling finer-grained failure diagnosis than a binary success/failure label(Yi et al., 2020; Zhou et al., 2025).