Cube Bench: A Controlled Diagnostic for Spatial and Sequential Reasoning in Multimodal Large Language Models

Anonymous ACL submission

Abstract

Recent multimodal large language models (MLLMs) are credited with three capabilities: they fuse vision and text, forecast the consequences of an action, and recover from mistakes through reflection. A fourth assumption underpins every agentic deployment built on these models: that strong single-step performance transfers to competent multi-step behavior. Existing evaluations cannot adjudicate these claims cleanly: static perception suites never let small errors compound, and agent suites aggregate perception, planning, and control failures into a single score. No instrument exposes where, along the perceive-forecast-act-reflect-recover loop, a multimodal model fails. We introduce **Cube Bench**, a controlled diagnostic built on the Rubik's Cube, where transitions are deterministic and oracle distance to the goal is available at every state. Seven oracle-scored tests isolate each stage of the decision loop under matched conditions. Across open and closed-source MLLMs, current models do not reliably exhibit any of the three credited capabilities, and the fourth assumption fails most sharply: a model with frontier-adjacent single-step skill across perception, prediction, and control can sit at the floor on recovery from its own errors. This dissociation between local skill and multi-step recovery has direct implications for agentic systems that assume these capabilities compose.

1 Introduction

Recent multimodal large language models (MLLMs) are credited with three capabilities that have become central to how the field talks about progress: that they fuse vision and text rather than treating them as parallel channels (Liu et al., 2023b; Tong et al., 2024), that they can forecast the consequences of an action before taking it (Yao et al., 2023a), and that they can recover from their own mistakes through self-reflection (Shinn et al., 2023; Madaan et al., 2023; Yao et al., 2023b). A fourth assumption is rarely stated but is load-bearing for every agentic deployment built on these models: that strong single-step performance transfers to competent multi-step behavior (Dziri et al., 2023; West et al., 2024). Together, the three claims and the assumption justify architectural choices, training objectives, and downstream products.

Existing evaluations do not adjudicate these claims cleanly. Synthetic VQA suites such as CLEVR and GQA (Johnson et al., 2017; Hudson and Manning, 2019) and natural-image spatial-relation probes such as VSR (Liu et al., 2023a), isolate aspects of spatial understanding on fixed scenes, but never couple perception to a dynamics model, never expose progress toward a goal, and never let a small error compound; broader multimodal leaderboards continue this pattern, emphasizing captioning, OCR, and short-form VQA (Li et al., 2024). Web- and embodied-agent suites (Zhou et al., 2024; Koh et al., 2024) restore horizon but collapse perception, planning, and control failures into a single success rate, and their non-deterministic environments make replication and causal attribution difficult. The field lacks an instrument that exposes where a multimodal model fails along the chain of perceiving, forecasting, acting, reflecting, and recovering.

We introduce **Cube Bench**, a controlled diagnostic built on the Rubik's Cube. The cube offers four properties absent from web-agent and embodied-agent suites: an exact transition model, oracle distance-to-solved via optimal solvers (Korf, 1997; Rokicki et al., 2014), paired image and text representations of every state, and a small discrete action set that admits strict answer-position controls. We define seven oracle-scored tests, each isolating one stage of a model's decision process: face reconstruction, cross-modal verification, optimal move prediction under three modalities, causal move-effect, multi-step closed-loop execution, reflection-

guided re-answering, and post-error recovery.

Evaluating eight MLLMs from five vendors (7B to frontier) surfaces four behaviors that contradict prevailing claims about multimodal reasoning, ordered here by their relevance to the NLP literature.

Seven of eight models do not benefit from images when text is available. Only Gemini 2.5 Pro shows the canonical image+text > image > text ordering that fusion should produce (Alayrac et al., 2022). Four models, including Claude Sonnet 4.5 at 100% parse compliance on both modalities, score *strictly higher* on text-only than on image+text (Table 2). The pattern is not a perception artifact: it holds in models that exceed 95% on balanced cross-modal verification.

After a wrong move no model recovers within one attempt. P(1)=0 across eight MLLMs (P(k): fraction of error trials solved within k attempts). Within six attempts, only Gemini achieves non-trivial recovery (Solve Rate SR=40%); the other seven cluster at SR$\leq 8\%$. This is direct evidence against the strong reading of the self-reflection literature (Shinn et al., 2023; Madaan et al., 2023; Huang et al., 2023) in the multimodal-action regime.

Strong single-step skill does not predict multi-step recovery. Sonnet 4.5 ranks second of eight on three independent single-step measures: text-only Move Prediction, $d = 1$ Teacher Adherence (TA, fraction of model moves matching an optimal-solver move), and $d = 1$ causal forecasting. Yet it sits at the open-weight floor on recovery (SR$= 4\%$). The dissociation is isolable within a single frontier closed-source model, ruling out scale or licensing as confounds, and extends to multimodal action the compositional limitations Dziri et al. (2023) document in text-only LLMs.

Causal forecasting collapses with depth. Pre-action κ separates models into three tiers at d=1 and collapses to two by d=3, where only Gemini retains signal. At d=1, κ predicts closed-loop TA across the full model set with r=0.995; this regularity does not survive to d=3.

Our contributions are: (i) **Cube Bench**, a controlled diagnostic that decomposes the perceive-forecast-act-reflect-recover loop into seven oracle-scored tests with strict fairness controls and a released reproducible generator; (ii) four behavioral findings that contradict standard claims about multimodal fusion and self-correction, and connect multimodal-action failures to known compositional limitations of LLMs (Dziri et al., 2023); (iii) two mechanistic regularities at d=1, a near-perfect κ-to-control correlation (r=0.995) and asymmetric reflection effects (+14 to −20), that hold within the single-step regime but break with depth.

2 Related Work

We provide a more substantive related work in Appendix A. Here we provide a very brief overview.

Recent cube-focused VLM/LLM efforts mix perception, planning, and tooling (e.g., CubeRobot, VisionCube, Auto-RubikAI), while earlier robotics work (Dactyl) highlights actuation confounds (Wang et al., 2025b,a; Fan and Yuan, 2025; OpenAI et al., 2019). Our setting keeps the environment compact and oracle-scored to avoid attribution ambiguity. Compared to perception-centric benchmarks (Russakovsky et al., 2015; Lin et al., 2014; Johnson et al., 2017; Hudson and Manning, 2019; Sidorov et al., 2020; Masry et al., 2022; Li et al., 2024; Goyal et al., 2017) and open-web agent suites (Zhou et al., 2024; Koh et al., 2024), our benchmark provides (i) deterministic generation with paired image and text state, (ii) strict I/O with a small, discrete action set, (iii) fairness controls that remove answer position bias, and (iv) exact oracle progress signals grounded in optimal-move distance (Korf, 1997; Rokicki et al., 2014). This design yields compact, reproducible measurements of closed-loop competence and exposes where modern MLLMs fail to maintain multi-step coherence, even when single-step perception appears strong.

3 Cube Bench: A Diagnostic Protocol

A multimodal model deciding under feedback performs five operations in sequence: it **perceives** the current state, **forecasts** how candidate actions would change it, **acts**, **reflects** on the result, and **recovers** if the result was off-trajectory. Standard multimodal evaluations score the composition of these operations through a single accuracy number, which is informative about a model on average but uninformative about where the model fails. Cube Bench is structured to expose the failing operation rather than aggregate it.

The five operations correspond to the three claims and the implicit assumption named in § 1. Perceive, together with the modality conditions of Act, tests fusion (Claim 1). Forecast tests consequence prediction (Claim 2). Reflect tests self-correction (Claim 3). Act under multi-step feedback combined with Recover tests the implicit skill-

transfer assumption. Each of the seven tests described in § 3.2 isolates one operation while holding the others fixed.

The instrument is bounded by design. Cube Bench does not measure perception under partial observability, tool use, or long-horizon planning beyond five moves. We treat the bound as a feature: the setting is small enough to control, exact enough to score without judgment, and uniform enough across models that any observed inter-model difference is attributable to the model rather than to the stimulus. § 5 discusses what the bound implies for external validity.

3.1 Why the Rubik's Cube

The Rubik's Cube combines four properties that we are not aware of in any other setting suitable for behavioral analysis of MLLMs.

Exact transition model. Move semantics are deterministic and lossless; the simulator's next state matches what any rule-following solver would compute.

Oracle distance to solved. For any state, IDA* with pattern databases (Korf, 1997) returns the exact minimum number of moves to the solved configuration under the face-turn metric. We can therefore score progress at every step, not only terminal success.

Paired modalities, both authoritative. Every state is rendered as an unfolded cube-net image and serialized as a canonical text string; either representation fully specifies the state. This property is what makes fusion testable: a model that genuinely combines vision and text should benefit from receiving both, while a model that ignores one modality should not. The implication for the multimodal claim is taken up in §4.2.

Small discrete action set. The full action set contains 18 face-turn moves; the four-option MCQ subset we use at each decision point contains four. Small action sets admit the strict answer-position controls in §3.3 that larger action spaces cannot support.

Two existing uses of the Rubik's Cube should be distinguished from ours. CubeRobot (Wang et al., 2025b) and Dactyl (OpenAI et al., 2019) target physical manipulation policies. Reasoning-gym (Stojanovski et al., 2025) includes a cube task as an RL-training environment. Cube Bench is neither: it is a closed-form behavioral diagnostic for already-trained models, scored against oracle ground truth.

3.2 Seven Tests

(1) Face Reconstruction. From a cube-net image, the model outputs the 3×3 grid of sticker colors on a specified face. Element-wise and exact-match accuracies measure visual recognition with no decision component; the test establishes whether vision alone is the bottleneck for downstream failures.

(2) Cross-Modal Verification. Given a cube image and a text description of the front face, the model answers Yes (consistent) or No (inconsistent). Items are balanced 50/50 across both labels and include lexical polarity reversals to suppress lexical shortcuts. Balanced accuracy measures whether the model actually compares the two modalities; a model that ignores either one cannot exceed chance.

(3) Optimal Move Prediction. From a state one optimal move from solved, the model picks the unique solving move from four labeled options. The test is run under three input modalities (image-only, text-only, and image+text) with prompts, option sets, and decoding identical across the three conditions. Comparing accuracies across modalities is the direct empirical test of Claim 1.

(4) Causal Move-Effect. Given a state and a candidate move, the model predicts whether applying the move would `INCREASE`, `DECREASE`, or leave `NO_CHANGE` the oracle distance, without applying it. Cohen's κ is the headline metric, chance-corrected against each model's empirical marginal so that constant-class or marginal-matching predictors do not score above chance.

(5) Step-by-Step Execution / Closed-Loop. From a state at depth $d \in \{1, 2, 3, 4, 5\}$, the model picks one action per step from four options for up to d steps; the chosen move is applied and the model sees the resulting state before the next step. Teacher Adherence (TA%) is the fraction of the model's moves that match an optimal-solver move from the state the model actually reached. Perfect-Solve% is the fraction of episodes solved without ever deviating from an optimal trajectory.

(6) Reflection-Guided Re-Answering. Starting from items in Test 3, the model is asked to reflect on its initial answer and emit a revised final answer. We report two regimes: a *redacted* default in which the correct option is hidden during reflection, and an *unredacted* oracle ablation that establishes an upper bound. Error-Fix Rate (EFR) counts initially-wrong items corrected after reflection; Overthink Rate (OTR) counts initially-correct items flipped

Model	Element-wise Acc.			Matrix Acc.			Verification (%) @ $d = 5$		
	$d = 1$	$d = 2$	$d = 3$	$d = 1$	$d = 2$	$d = 3$	Bal.	Parse	Yes-Rate (%)
Gemma 3	61.9	47.2	36.60	31.00	8.00	3.00	58.00	100.00	70.00
GLM-4.5V	100.00	99.70	98.70	100.00	98.00	96.00	95.00	100.00	55.00
Llama 4-Scout-17B	70.60	41.00	33.20	36.00	7.00	3.00	47.00	100.00	93.00
Qwen2.5-VL 7B	76.20	70.60	62.00	40.00	22.00	13.00	71.30	98.00	57.10
Qwen2.5-VL 32B	84.00	78.80	71.30	68.00	43.00	19.00	81.00	100.00	65.00
Qwen3-Thinking 30B	98.10	95.40	86.70	94.00	78.00	60.00	84.00	100.00	66.00
Gemini 2.5 Pro	100.00	98.80	96.80	100.00	90.00	84.00	100.00	100.00	52.00
Claude Sonnet-4.5	100.00	100.00	99.77	100.00	100.00	98.00	100.00	100.00	52.00

Table 1: Left: *Cube Face Reconstruction* reported as element-wise (per-sticker) accuracy and exact 3×3 matrix accuracy at depths $d \in \{1, 2, 3\}$. Right: Cross-modal Yes/No Verification at fixed depth $d = 5$ with **Bal**=$(\mathrm{TPR} + \mathrm{TNR})/2$, **Parse**=% outputs matching `Answer: Yes|No`, and **Yes-Rate**=% of "Yes" responses (50 = balanced, deviation in either direction indicates bias)

to wrong.

(7) Learning Curve / Recovery. Beginning at the first state where the model deviated from the optimal trajectory in Test 5, we grant the model a depth-dependent attempt budget to re-enter a solving trajectory; the model's chosen move is always applied regardless of whether it makes progress. Solve Rate (SR) is the fraction of episodes solved within budget. $P(1)$ and $P(\leq 3)$ are the fractions solved within one and three attempts respectively, measuring immediate versus near-immediate repair.

3.3 Fairness Controls

Several observations in § 4 hinge on controls that prevent shortcut strategies. We summarize the load-bearing items here; full protocols are in Appendix B & C.

Deterministic episodes across models. Each episode at depth d uses a fixed seed equal to its index. All evaluated models therefore see the same scrambles, the same option sets, the same prompt, and the same decoding policy (temperature 0, large `max_new_tokens`). Any inter-model variation reflects model behavior rather than stimulus variation.

Independent answer-position shuffling. For multiple-choice items, option positions are permuted by a full-entropy random source independent of both the scramble seed and the item index. The correct-letter distribution is therefore approximately uniform over A-D in expectation, and no model can win by preferring a fixed slot, see Appendix C.8.2

Strict parsing protocol without retries. The model must emit exactly one answer via `<ANSWER> X </ANSWER>` or `ANSWER: X`. Any other output is a parse-failure; parse-failures count as errors and the environment does not advance. There is no resampling and no parser-side leniency. We report parse rates alongside accuracies so that instruction-following deficits cannot masquerade as reasoning competence.

Explicit chance baselines. Per-step four-option MCQ accuracy has a chance baseline of 25% (Tests 3, 5, and the initial and final-accuracy components of Test 6). Test 2 reports balanced accuracy with a chance baseline of 50%. Cohen's κ in Test 4 is chance-corrected by construction, so its chance baseline is $\kappa = 0$. Solve rates and $P(1)$ in Test 7 depend on the post-error state distribution; we report Wilson 95% confidence intervals. All proportion CIs use the Wilson score interval (`statsmodels` proportion_confint, method='wilson', $\alpha = 0.05$); sample sizes per test are given in each table caption.

Identical metric for scrambling and scoring. The face-turn metric (FTM) is used both to generate scrambles and to compute oracle distances. A model is never penalized under a metric different from the one that defined the task.

4 Experiments

We evaluate eight MLLMs spanning five vendors and scales from 7B to frontier: Gemini 2.5 Pro (Google) (Google, 2025), Claude Sonnet 4.5 (Anthropic) (Anthropic, 2025), Gemma 3-27B (Google) (Team et al., 2025a), GLM-4.5v (Z.ai) (Team et al., 2025b), Llama 4-Scout-17B:16E (Meta) (Meta AI, 2025), Qwen2.5-VL-7B/32B-Instruct (Bai et al., 2025), and Qwen3-VL-30B-A3B (Team, 2025). Closed-source models (Gemini 2.5 Pro, Claude Sonnet 4.5) are accessed through their official APIs; open-weight models are run locally on NVIDIA A100-80GB GPUs with vLLM (Kwon et al., 2023), bf16, default chat templates, and request-level batching for throughput. All models share the same decoding policy: temperature

Model	$I{+}T$		I		T		
	Acc	Pa.	Acc	Pa.	Acc	Pa.	Δ_{img}
Gemma 3-27B	33	100	27	100	31	100	+2
GLM-4.5V	31	98	21	98	28	96	+3
Llama 4-Scout 17B	18	100	16	100	25	100	−7
Qwen 2.5-VL-7B	19	100	18	100	26	100	−7
Qwen 2.5-VL-32B	28	100	27	100	26	100	+2
Qwen3-VL-30B-Think.	23	86	19	98	32	77	−9
Gemini 2.5 Pro	96	100	92	100	80	100	+16
Claude Sonnet 4.5	64	100	52	100	66	100	−2

Table 2: Optimal Move Prediction at d=1. Top-1 accuracy (%) and format compliance (Pa. = parse rate, %) across prompt modalities. Parse counts outputs matching `ANSWER: [A-D]` or `<ANSWER>X</ANSWER>`; non-parsing outputs are scored incorrect. Δ_{img} = I+T − T isolates the marginal contribution of image when text is available. Items and option-shuffle seeds matched across rows and conditions; $n = 100$ per cell.

Model	d=1			d=2			d=3		
	Acc.	F1	κ	Acc.	F1	κ	Acc.	F1	κ
Gemma3	32.00	0.30	-0.025	31.50	0.309	-0.027	27.00	0.262	-0.099
GLM-4.5V	36.00	0.31	0.0014	30.00	0.25	-0.0856	29.00	0.237	-0.1011
Llama 4-Scout 17B	28.50	0.27	-0.084	27.75	0.27	-0.079	27.25	0.27	-0.084
Qwen2.5 VL 7B	37.00	0.21	0.0024	35.75	0.20	0.00059	36.50	0.20	0.00902
Qwen2.5 VL 32B	32.20	0.25	-0.074	39.30	0.316	0.0403	41.25	0.335	0.0859
Qwen3-Thinking 30B	33.00	0.29	-0.029	32.00	0.269	-0.034	35.72	0.306	0.022
Gemini 2.5 Pro	79.00	0.78	0.680	65.00	0.632	0.475	62.00	0.606	0.439
Claude Sonnet-4.5	59.50	0.58	0.376	40.00	0.35	0.076	36.50	0.31	0.023

Table 3: Move-Effect results at scramble depths $d \in \{1, 2, 3\}$. Acc. is micro-accuracy; F1 is macro-F1 over {DECREASE, NO_CHANGE, INCREASE}; κ is Cohen's chance-corrected agreement (chance baseline $\kappa = 0$).

0 (greedy), a large `max_new_tokens` cap, identical prompts per task and modality, the same published per-depth seed lists, and a FTM per run.

Parsing is strict: one line outputs of the form `A-D`, `Yes|No`, or `|DECREASE|NO_CHANGE|INCREASE|`, with any deviation counted as an error. Per-condition $N = 100$ is fixed across models for Tests 1-5 and 7, so that comparisons are paired on identical items. In Test 6 (Reflection-Guided Re-Answering), compute constraints at the time of running forced N to vary by model (25-100; see Table 4 and Appendix Table 8); we report 95% Wilson intervals so that paired-vs-unpaired contrasts can be read directly.

Coverage is complete across Tests 1–5 and 7 for the original seven-model set. Two coverage gaps are flagged explicitly. Claude Sonnet 4.5 is evaluated on Tests 1, 2, 3, 4, 5, and 7; we did not run Sonnet on Test 6 (Reflection-Guided Re-Answering) due to compute constraints. Gemini 2.5 Pro is similarly absent from Test 6: at 96% Initial Accuracy on Optimal Move Prediction (Image+Text), the headroom for measuring reflection gains is near-degenerate (EFR is computed over ≤ 4 initially-wrong items), so a reflection score on Gemini would be dominated by sampling noise rather than reflection behavior. No claim in §4.4 or §5 rests on a Test 6 result for either model.

4.1 Perception: Reconstruction and Verification

Before testing behavioral claims, we bound the perception floor against which downstream results must be read. We evaluate **Face Reconstruction** (output the 3×3 Front face from a rendered cube net) and **Cross-Modal Verification** (decide whether a textual state matches a rendered image; balanced Yes/No items).

Results. Every model exceeds chance on element-wise reconstruction averaged over $d \in \{1, 2, 3\}$; the top four (Sonnet, Gemini, GLM, Qwen3) exceed 80% (Table 1, left). Exact 3×3 matrix accuracy is uniformly lower because a single misplaced sticker collapses the score from 1 to 0, and the gap widens with depth. At Verification (d=5), the same top four clears 80% balanced accuracy (Gemini 100, Sonnet 100, GLM 95, Qwen3 84; Table 1, right), with parse rates $\approx$ 100%, so the residual gap is discrimination, not formatting.

Takeaway. Perception is not the binding constraint at $d \leq 3$. Where Move Prediction, Closed-Loop Adherence, or Recovery fails for a model with near-ceiling verification (GLM, Sonnet; §4.2-§4.6), the failure is behavioral, not perceptual.

4.2 Optimal Move Prediction

Multimodal LLMs are assumed to fuse image and text into a representation more useful than either alone: performance should order $I{+}T > I > T$ (Alayrac et al., 2022). We test this with a 4-option MCQ at d=1 under each input condition (Table 2.

Only Gemini fuses. $\Delta_{img} = I{+}T - T$ is positive and substantive only for Gemini (+16). Four models score strictly higher on text-only than on image+text: Qwen3 (−9), Llama 4 (−7), Qwen 2.5-VL-7B (−7), and Sonnet 4.5 (−2). The remaining three (Gemma, GLM, Qwen 2.5-VL-32B) clear T with $I{+}T$ by only 2–3 points. Qwen3's −9 is partly explained by its 77% text-only parse rate; the other three text-biased models parse at 100% on every condition, so their gap is substantive. Sonnet 4.5 is the load-bearing case: a frontier closed-source model parsing at 100% across modalities and still showing $T > I{+}T$.

Takeaway. The prompt marks text as authoritative when present (§3.2). This does not preclude

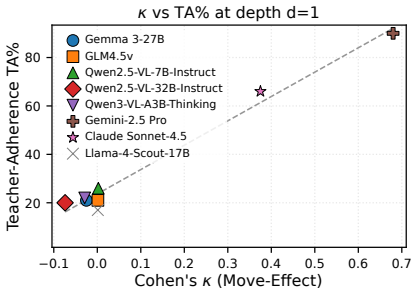


Figure 1: Move-Effect κ vs. Closed-Loop TA% at depth d=1. Pearson $r = 0.995$ (95% CI [0.969, 0.999]); least-squares fit shown.

Model	N	Δ ↑	EFR (%) ↑	OTR (%) ↓
Gemma 3-27B	100	0.0	30.00	70.00
GLM-4.5v	25	-20.0	6.25	66.67
Llama 4-Scout 17B	50	+14.0	20.00	40.00
Qwen 2.5-VL 7B	100	+11.0	30.00	65.00
Qwen 2.5-VL 32B	100	+10.0	34.67	64.00
Qwen3-VL-Thinking 30B	50	+6.0	27.78	50.00

Table 4: Reflection-guided re-answering on Optimal Move Prediction (Redacted). Δ: net accuracy change; EFR: share of initially wrong items corrected; OTR: share of correct items flipped. Image+Text, temperature 0, strict A-D parsing. N varies (25-100); Wilson 95% CIs in Appendix E.2. Gemini and Sonnet omitted (§4).

$I{+}T > T$ under any standard fusion account: image is free to supply complementary signal. That seven of eight models do not gain meaningfully from image when text is available is evidence about fusion behavior, not a prompt artifact.

4.3 Causal Move-Effect

Pre-action causal forecasting separates three tiers at one move and collapses to two by three. At d=1, κ sorts models into three tiers (Table 3): Gemini 2.5 Pro (κ=0.68), Claude Sonnet 4.5 (κ=0.376), and six open-weight models at chance under Cohen's κ (-0.084 to $+0.002$). By d=3 the tiers collapse to two: Gemini retains meaningful forecasting (κ=0.44), while Sonnet has fallen to the open-weight floor (κ=0.023), a 94% loss of d=1 signal against Gemini's 35%. A frontier closed-source MLLM with genuine d=1 signal losing it by d=3 is the load-bearing observation: short-horizon causal evaluation does not survive scrambling, even for models that retain it at one move.

κ predicts closed-loop control at one move, and only at one move. At d=1, pre-action κ and closed-loop Teacher-Adherence (§4.5) are tightly coupled across the full eight-model set: Pearson r=0.995 (95% CI $[0.969, 0.999]$; Fig. 1). Sonnet 4.5 sits as an interior point on the regression line (predicted TA $\approx$ 62, observed 66), confirming the relationship as continuous rather than a two-cluster artifact. The law is regime-bounded: at $d{\geq}2$ the open-weight cluster collapses to the origin while Gemini degrades but stays distinct (per-depth plots in Appendix E.1), so κ-as-predictor characterizes the single-step regime alone. Multi-step recovery is governed by capabilities that this law does not index; we document the resulting dissociation in §4.6.

DECREASE blindness is the behavioral default of current MLLMs. Seven of eight evaluated models, spanning open-weights at 7B to frontier closed-source at 100% (99.83% on this task) parse compliance, produce prediction mixes biased toward INCREASE (55–75% predicted INCREASE against a gold rate of $\approx$ 36%). Qwen-2.5-VL-32B's small κ gains at higher d ($-0.074 \rightarrow 0.040 \rightarrow 0.086$) are a prior-alignment effect: as the depth-wise class prior places more mass on INCREASE, micro-accuracy edges up while κ stays near zero because chance-correction adjusts for the skewed prediction mix. The presence of the same bias in Sonnet 4.5 rules out the open-weight-scale reading of this pattern. Gemini alone calibrates; the remaining seven, including a frontier closed-source MLLM, treat most moves as "making things worse".

Takeaway. (i) κ, not Micro, is the metric that exposes population-level prediction-mix biases that depth-dependent priors otherwise hide. (ii) The r=0.995 κ-TA correspondence is a one-step regularity; reviewers and downstream work should not extrapolate it to multi-step settings, where §4.6 shows it breaks. (iii) DECREASE blindness is a property of current MLLMs broadly, not of open-weight scale or licensing, and is a candidate mechanism behind the universal $P(1)$=0 result in §4.6.

4.4 Reflection-Guided Reasoning

Reflection without access to the gold answer is not a uniformly positive intervention. GLM-4.5v loses 20 percentage points after a single reflection pass (Table 4), collapsing from 36% to 16%. Two thirds of the items GLM initially got right are flipped to wrong (OTR = 66.67%), while only 6.25% of wrong items are corrected (EFR). Gemma-3-27B shows no net benefit ($\Delta = 0$): EFR (30%) and OTR (70%) cancel exactly. At the other end, Llama 4-Scout-17B and Qwen-2.5-VL-7B both gain +14 and +11 points respectively (EFR 20-30%, OTR 40-65%), Qwen-2.5-VL-32B gains +10 (EFR 34.7%, OTR 64%), and Qwen3-VL-Thinking gains a mod-

est +6 (EFR 27.8%, OTR 50%). Four of six models gain, one is flat, and one regresses sharply; the net effect is entirely determined by the EFR-OTR balance. Reflection benefits reported for text-only LLMs (Shinn et al., 2023; Madaan et al., 2023) do not transfer cleanly to the multimodal-action regime.

Unredacted (oracle hint) ablation. Table 4 reports the "Guided (Redacted)" regime, in which reflection operates without access to the gold answer. We additionally evaluated a leaky "Guided (Unredacted)" regime in which the reflection prompt includes the correct option as an oracle hint, establishing an upper bound on reflection-driven gains. Under Unredacted, every model improves sharply: GLM-4.5v and Llama 4-Scout-17B:16E climb from 36% and 10% initial accuracy to 100% (gains of +64 and +90 points); the Qwen models jump to 59-72% final accuracy with gains of +42 to +50 points; and Gemma 3-27B rises from 29% to 81%. The contrast between the two regimes is direct evidence of a generation–discrimination asymmetry: every model can rationalize toward the correct option when shown it, but under fair conditions a single reflection pass helps some models and hurts others.

Takeaway. The 81-100% Unredacted final accuracy versus −20 to +14 Redacted instantiates the generation-discrimination asymmetry of West et al. (2024) in the multimodal-action regime: every model can identify the correct option when shown it, but only some produce it under fair conditions. Reflection benefits reported for text-only LLMs (Shinn et al., 2023; Madaan et al., 2023) do not transfer cleanly here.

4.5 Closed-Loop Control

Depth-collapse is universal across vendors and scales. Every evaluated MLLM loses TA% sharply between d=1 and d=5 (Fig. 2). Gemini starts in the green band at d=1 (TA = 90) and falls to 10 by d=5. Sonnet 4.5 enters the green band at d=1 (TA = 66) and crosses into the red band between d=3 and d=4 (66 → 31 → 17 → 10 → 10). The six open-weight models start at the orange-red boundary at d=1 (cluster between TA = 16 and TA = 26 at $d = 1$) and converge below 10 by $d \geq 3$. [2]

[1] Llama4 and Qwen3 are omitted from the grid for clarity; their TA(d) curves fall inside the open-weight cluster shown and exhibit the same depth collapse pattern.

[2] d=4 → d=5 inversion visible for two models in Fig. 2 is within Wilson 95% CI noise at n=50 and is not interpreted.

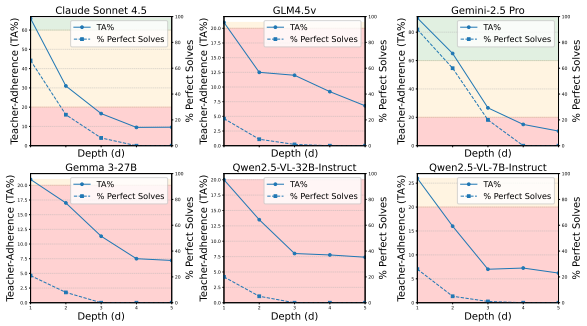


Figure 2: Step-by-step closed-loop results. *TA%* is Teacher-Adherence, i.e., $\frac{\text{avg correct steps}}{d} \times 100$. *Perfect%* is the share of episodes with perfect teacher adherence. Accuracies use unconditional denominators. Bands (w.r.t TA%): 0–20% 20–60% 60–100% .[1]

The frontier-vs-open gap shrinks with depth. At d=1 Gemini shows roughly 4.5× adherence to the open-weight cluster (90 vs ∼22); Sonnet 4.5 shows roughly 3× (66 vs ∼22). The two-vendor closed-source advantage is real at one move but is consumed by depth: by d=3, the Gemini-to-floor gap has narrowed from ∼70 points to ∼20, and by d=5, every model sits within a 10-point band. Whatever ordering held at d=1 is dominated by depth collapse at $d \geq 3$.

Takeaway. Three observations characterize closed-loop control. (i) Depth-collapse is universal across vendors and scales: every model loses at least 65% of its d=1 TA by d=5. (ii) The closed-source advantage exists only at d=1 and is consumed by depth, so frontier-versus-open comparisons made on a single move generalize poorly to multi-step settings. (iii) The failure is planning, not perception, since models with near-ceiling reconstruction and verification (GLM, Qwen3; §4.1) collapse identically to weaker-perception peers. This attribution is by design: we show only the current state S_t rather than the trajectory $(S_0, a_0, \ldots, S_t)$ so closed-loop adherence isolates state-tracking and planning from temporal-context retrieval (§5 discusses transfer implications). An abstention-aware ablation at d=5 with an explicit IDK action (Kalai et al., 2025) modestly improves penalty-aware accuracy without changing the depth-collapse pattern; full numbers are in Appendix C.6.

4.6 Learning-Curve / Recovery

Each Learning Curve episode begins at the first off-trajectory state a model produced in Test 5 (Step-by-Step) at scramble depth $d = 3$, and grants the model a six-attempt budget to re-enter a solving

trajectory (Table 7). The model's chosen move is always applied, regardless of whether it makes progress, and the model sees each resulting state before its next attempt. We run $n = 50$ episodes per model.

No model recovers within one attempt. Across eight evaluated MLLMs (five vendors, 7B to frontier), $P(1) = 0$: no model returns to a solving trajectory on the move immediately following an off-trajectory action (Table 5). Within a six-attempt budget, only Gemini achieves non-trivial recovery (SR $= 40\%$, $P(\leq 3) = 0.40$); every other model clusters at SR $\leq 8\%$, with Qwen-7B at 8%, Llama at 6%, Sonnet 4.5 at 4%, Gemma at 4%, GLM at 2%, Qwen3 at 2%, and Qwen-32B at 0%. This is direct evidence against the strong reading of self-correction claims for the multimodal-action regime (Shinn et al., 2023; Madaan et al., 2023; Huang et al., 2023): only one of eight models repairs even half of error episodes, and none repair within one attempt.

Recovery failures look like exhaustion, not divergence. Low-SR models drift to the budget limit rather than diverge from the solving trajectory. Sonnet 4.5's 48 failed episodes are all labeled `non_progress` at the pre-fail step (the move that triggered the failure is neither progress nor catastrophe), and its Avg@All $= 5.9$ sits inside the open-weight cluster's 5.88 to 6.00, well above Gemini's 4.80. Models with SR $\leq 8\%$ keep trying, rarely find a fix, and exhaust the budget. Extending the budget is unlikely to help: the failure mode is not premature termination.

Single-step skill does not predict multi-step recovery. Sonnet 4.5 is the cleanest case. Behind only Gemini, it places second of eight on three independent single-step measures: Move Prediction text-only $= 66\%$ (Table 2), closed-loop adherence at d=1 TA $= 66\%$ (Fig. 2), and one-move causal forecasting $\kappa = 0.376$ (Table 3). Perception is also at ceiling: 100% element-wise reconstruction at $d = 1$, 100% matrix accuracy at $d = 1$, and 100% balanced verification at $d = 5$ (Table 1). No upstream representational bottleneck is available to explain what follows. On recovery, Sonnet sits at SR $= 4\%$, indistinguishable from Gemma (4%) or Llama (6%). A frontier closed-source MLLM with three frontier-adjacent single-step metrics does not, on the strength of those metrics, recover from off-trajectory states. The metrics that predict d=1 control do not predict d=3 recovery, consistent with prior NLP results that strong local reasoning fails to compose into reliable multi-step behavior (Dziri et al., 2023; West et al., 2024; Saparov and He, 2023; Valmeekam et al., 2023). Effective recovery likely requires explicit mechanisms (short rollouts, self-checks, external state estimation) that current MLLMs do not deploy as part of their default inference behavior.

Model	SR	95% CI	P (1)	P (≤ 3)	Med@ Solved	Avg@ All
Gemini 2.5 Pro	40.00	[27.61, 53.82]	0.00	0.40	3.00	4.80
Qwen2.5 VL 7B	8.00	[3.15, 18.97]	0.00	0.04	4.50	5.88
Llama 4-Scout 17B	6.00	[2.06, 16.22]	0.00	0.00	6.00	5.98
Gemma 3-27B	4.00	[1.10, 13.46]	0.00	0.04	3.00	5.88
Claude Sonnet-4.5	4.00	[1.10, 13.46]	0.00	0.02	3.50	5.90
GLM-4.5v	2.00	[0.35, 10.50]	0.00	0.02	2.00	5.92
Qwen3-Thinking 30B	2.00	[0.35, 10.50]	0.00	0.01	5.00	5.94
Qwen2.5 VL 32B	0.00	[0.00, 7.13]	0.00	0.00	N/A	6.00

Table 5: Learning Curve / Recovery (n=50 episodes per model, six-attempt budget from first off-trajectory state in Test 5). SR = solve rate (%); 95% CI = Wilson interval (%); P(1), P($\leq$3) = fractions solved within 1 and 3 attempts (proportions); Med@Solved = median attempts among solved episodes (N/A if none); Avg@All = mean attempts with failures scored at max_attempts=6.

5 Conclusion

Three capabilities have become central to how current multimodal large language models are described: that they fuse vision and text, forecast the consequences of an action, and recover from their own mistakes. A fourth assumption is implicit but load-bearing for agentic deployments: that strong single-step skill transfers to multi-step behavior. Cube Bench tests each claim across eight MLLMs under matched conditions and finds that none holds without qualification.

Only Gemini 2.5 Pro shows the canonical fusion ordering; seven of eight do not benefit from added images when text is available. Causal forecasting collapses with depth, retaining a signal at one move that vanishes by three for all but Gemini. No model recovers from its own error within one attempt. Claude Sonnet 4.5 makes the dissociation explicit: second of eight on three independent single-step measures, at the open-source floor on recovery.

Progress will require explicit mechanisms for pre-action evaluation, selective control, and recovery that current MLLMs do not deploy by default. Cube Bench is a small, oracle-scored instrument for studying those capabilities in successor models.

Limitations

Cube Bench uses oracle-scored tasks with exact transition models, small discrete action sets, and states fully specified by text. This cleanly attributes inter-model differences (§4) to the models themselves, avoiding stimulus noise, partial observability, or judgmental scoring. While numerical gaps may not transfer to embodied robotics, web agents, or partially observable natural-language tasks, the observed failure modes under these pristine conditions ($P(1) = 0$ across eight MLLMs; frontier-adjacent local skill at $d = 1$ vs. floor-level recovery at $d = 3$) establish credible behavioral patterns that open-ended evaluations would struggle to isolate.

Text as authoritative. When both modalities are present, text is authoritative. This does not weaken our multimodal claims for two reasons. First, seven of eight MLLMs (including Claude Sonnet 4.5, which has 100% parse compliance across modalities) fail to benefit from added images; four score strictly higher on text-only. This reveals actual fusion behavior, not a prompt artifact. Second, genuine fusion remains possible: Gemini 2.5 Pro achieves the canonical ordering (image+text > image > text). Most models failing to benefit from both modalities is a finding, not a methodological flaw.

Model coverage. We evaluate eight MLLMs (open-weight 7B to frontier closed-source) from five vendors (Anthropic, Alibaba, Google, Meta, Z.ai). We explicitly omit GPT-4o and LLaVA; evaluating them would strengthen coverage. Within our evaluated set, Test 6 (Reflection) is not run on Gemini 2.5 Pro or Claude Sonnet 4.5; the former because its 96% initial accuracy leaves no measurable headroom, the latter due to compute constraints. Nonetheless, our findings are robust: text bias spans all vendors and both licensing types, $P(1) = 0$ holds universally, and local-to-recovery dissociation is demonstrable within a single closed-source frontier MLLM.

Bounded horizon. Depths $d \leq 5$ and six-attempt retry budgets reliably elicit collapse in all evaluated models. Deeper scrambles only add compute without altering qualitative findings. This bound aids diagnostic attribution but limits direct transferability to long-horizon agent tasks.

Single domain. Cube Bench is a single closed-form domain. We release a reproducible test generator to support future domain-transfer studies.

Artifact licensing. The Cube Bench generator, evaluation harness, and benchmark metadata will be released under the MIT License (code) and CC BY 4.0 (data and prompts), permitting research and commercial reuse with attribution. All evaluated MLLMs are used under their respective vendor terms: Gemini 2.5 Pro and Claude Sonnet 4.5 are accessed through their official APIs in accordance with Google's and Anthropic's usage policies; open-weight models (Gemma 3, GLM-4.5v, Llama 4-Scout, Qwen2.5-VL, Qwen3-VL) are used under their respective model licenses, which all permit research evaluation and benchmarking.

Use of AI Assistants

We used AI assistants (large language models) for editorial drafting, grammatical fixes, and LaTeX formatting during paper preparation. All experimental design, benchmark construction, model evaluations, statistical analyses, and final claims are the authors' own work. AI-generated text was reviewed and edited by the authors before inclusion.

References

Forest Agostinelli, Stephen McAleer, Alexander Shmakov, and Pierre Baldi. 2019. Solving the rubik's cube with deep reinforcement learning and search. *Nature Machine Intelligence*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. *Preprint*, arXiv:2204.14198.

Anthropic. 2025. Claude (model: Sonnet 4.5). Generative AI tool. Accessed: May 23, 2026.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. *Preprint*, arXiv:2305.18654.

Chongshan Fan and Shenghai Yuan. 2025. Structured task solving via modular embodied intelligence: A case study on rubik's cube. *arXiv:2507.05607*.

Google. 2025. Gemini 2.5 Pro — Model Card. https://modelcards.withgoogle.com/assets/documents/gemini-2.5-pro.pdf. Model card (last updated 2025-06-27). Accessed 2025-10-13.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, pages 6904–6913.

hkociemba. 2025. RubikOptimal: Optimal Rubik's cube solver. https://pypi.org/project/RubikOptimal/. Python package implementation of Korf's IDA* algorithm.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*.

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. Why language models hallucinate. *Preprint*, arXiv:2509.04664.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *Preprint*, arXiv:2401.13649.

Richard E Korf. 1997. Finding optimal solutions to rubik's cube using pattern databases. In *AAAI/IAAI*, pages 700–705.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Jian Li, Weiheng Lu, Hao Fei, Meng Luo, Ming Dai, Min Xia, Yizhang Jin, Zhenye Gan, Ding Qi, Chaoyou Fu, Ying Tai, Wankou Yang, Yabiao Wang, and Chengjie Wang. 2024. A survey on benchmarks of multimodal large language models. *Preprint*, arXiv:2408.08632.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Fangyu Liu, Guy Emerson, and Nigel Collier. 2023a. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

Meta AI. 2025. Llama 4 Scout 17B (16E) — Model Card. https://huggingface.co/meta-llama/Llama-4-Scout-17B-16E. Model card. Accessed 2025-10-13.

OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. 2019. Solving rubik's cube with a robot hand. *Preprint*, arXiv:1910.07113.

Matthew Renze and Erhan Guven. 2024. Self-reflection in llm agents: Effects on problem-solving performance. *arXiv preprint arXiv:2405.06682*.

Tomas Rokicki, Herbert Kociemba, Morley Davidson, and John Dethridge. 2014. The diameter of the rubik's cube group is twenty. *siam REVIEW*, 56(4):645–670.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *Preprint*, arXiv:2210.01240.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. 2020. Textcaps: a dataset for image captioning with reading comprehension. In *European conference on computer vision*, pages 742–758. Springer.

Zvonimir Stojanovski and 1 others. 2025. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards. In *Advances in Neural Information Processing Systems*, volume 38, pages 1–44.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025a. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, and 69 others. 2025b. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *Preprint*, arXiv:2507.01006.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9568–9578.

Karthik Valmeekam, Sarath Sreedharan, Matthew Marquez, Alberto Olmo, and Subbarao Kambhampati. 2023. On the planning abilities of large language models (a critical investigation with a proposed benchmark). *Preprint*, arXiv:2302.06706.

Feiyang Wang, Nan Luo, and Wangyu Wu. 2025a. Visioncube: 3d-aware vision-language model for multistep spatial reasoning. In *CVPR 2025 Workshops (eLVM)*.

Feiyang Wang, Xiaomin Yu, and Wangyu Wu. 2025b. Cuberobot: Grounding language in rubik's cube manipulation via vision-language model. *arXiv:2503.19281*.

Peter West, Ximing Lu, Nouha Dziri, Faeze Brahman, Linjie Li, Jena Hwang, Liwei Jiang, Jillian Fisher, Abhilasha Ravichander, Khyathi Chandu, and 1 others. 2024. The generative ai paradox:"what it can create, it may not understand". In *International Conference on Learning Representations*, volume 2024, pages 47159–47187.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. *Preprint*, arXiv:2307.13854.

A Related Work

Static perception benchmarks for MLLMs. A large body of work evaluates multimodal LLMs on single-shot perception tasks such as classification, detection, captioning, VQA, OCR, and charts, e.g., ImageNet, COCO, CLEVR, GQA, TextCaps, and ChartQA (Russakovsky et al., 2015; Lin et al., 2014; Johnson et al., 2017; Hudson and Manning, 2019; Sidorov et al., 2020; Masry et al., 2022). Probes targeting natural-image spatial relations, notably VSR (Liu et al., 2023a), show that VLMs systematically fail on orientation-dependent relations even on static scenes, motivating controlled diagnostics that isolate the spatial-reasoning component from the broader decision loop. Recent surveys synthesize these trends and note rapid leaderboard gains driven by scaling and instruction tuning (Li et al., 2024). Classic perception datasets (e.g., VQAv2) emphasize recognition under short contexts and have pushed models toward improved visual grounding and bias control (Goyal et al., 2017); ScienceQA further probes multimodal reasoning with thought chains (Lu et al., 2022). However, these settings rarely require *closed-loop* decision making, where actions alter the state and errors can compound across steps. Our benchmark targets this gap by measuring whether models can sustain improvements over multiple actions, given precise state feedback.

Interactive / agent benchmarks. Web-scale agent environments assess language-to-action competence in long-horizon tasks (navigation, form-filling, retrieval, tool use) (Zhou et al., 2024; Koh et al., 2024). While such benchmarks capture realistic challenges (dynamic pages, delayed rewards, partial observability), they introduce confounds, non-deterministic layouts, fragile tool chains, and evaluation noise, that complicate causal analysis of failures. We instead adopt a compact, fully controllable domain with deterministic transitions and exact progress signals, allowing us to isolate where breakdowns arise (perception, evaluation, selection, or reflection) without web-induced variance.

Reasoning and control in structured domains. Structured puzzles and games offer precise state representations, small action alphabets, and well-defined optimality notions. For the Rubik's Cube, shortest-path structure and the diameter ("God's number") enable oracle feedback about optimal move distance (Korf, 1997; Rokicki et al., 2014). Learning-to-search systems such as DeepCubeA demonstrate scalable RL+search on the cube and related combinatorial puzzles (Agostinelli et al., 2019). We repurpose these properties to build an *evaluation*, not a solver, so perception and control can be disentangled and measured reproducibly.

Rubik's-cube with VLM/LLM and embodied control. Recent cube-focused VLM/LLM efforts mix perception, planning, and tooling (e.g., CubeRobot, VisionCube, Auto-RubikAI), while earlier robotics work (Dactyl) highlights actuation confounds (Wang et al., 2025b,a; Fan and Yuan, 2025; OpenAI et al., 2019). Our setting keeps the environment compact and oracle-scored to avoid attribution ambiguity.

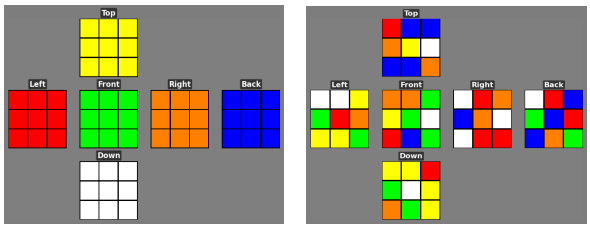
Self-reflection and tool-augmented prompting. Interleaving reasoning and acting (or adding self-critique) can improve sample efficiency but may also induce instability or "overthinking." Representative approaches include Reflexion (Shinn et al., 2023), Self-Refine (Madaan et al., 2023), ReAct (Yao et al., 2023b), and analyses of reflection effects (Renze and Guven, 2024). Our experiments use a constrained, label-aware reflection protocol to quantify when reflection repairs errors and when it degrades coherence, under identical rules across models.

Positioning. Compared to perception-centric benchmarks (Russakovsky et al., 2015; Lin et al., 2014; Johnson et al., 2017; Hudson and Manning, 2019; Sidorov et al., 2020; Masry et al., 2022; Li et al., 2024; Goyal et al., 2017) and open-web agent suites (Zhou et al., 2024; Koh et al., 2024), our benchmark provides (i) deterministic generation with paired image and text state, (ii) strict I/O with a small, discrete action set, (iii) fairness controls that remove answer position bias, and (iv) exact oracle progress signals grounded in optimal-move distance (Korf, 1997; Rokicki et al., 2014). This design yields compact, reproducible measurements of closed-loop competence and exposes where modern MLLMs fail to maintain multi-step coherence, even when their single-step perception appears strong.

B `VirtualCube` Implementation Details

B.1 Visual Rendering

To ensure complete state observability and eliminate ambiguity caused by occlusion in 3D perspective views, we utilize a standardized **unfolded "cube net" visualization** for all image-based tasks. As illustrated in Figure 3, this rendering style flat-



(a) A solved state. (b) A scrambled state.

Figure 3: **Example Rubik's Cube States.** The left image shows a cube in its fully solved configuration, where each face consists of a single color. The right image displays a cube in a scrambled state, typical of a starting position in a puzzle-solving task.

tens the cube's state into a 2D plane, exposing all six faces simultaneously: **Top**, **Left**, **Front**, **Right**, **Back**, and **Down**. Explicit text labels are rendered above each face to aid model grounding and ensure that the orientation is unambiguous.

The background is rendered in a uniform, **neutral gray**. This design choice maximizes contrast against the vibrant sticker colors (W, Y, R, O, G, B) and eliminates potential visual confounders, ensuring that the model's attention is focused exclusively on the state configuration rather than background artifacts.

B.2 Oracle Distance Implementation Details

To ensure reproducible and consistent evaluation of model performance, we utilize a deterministic oracle to compute the exact distance-to-solved $d(s)$ for every state.

B.2.1 Solver Algorithm and Optimality

The oracle distance $d(s)$ is defined as the length of the strictly shortest path from the current state s to the solved state in the Face Turn Metric (FTM).

To compute this, we utilize the `RubikOptimal` Python library (hkociemba, 2025). Unlike heuristic solvers (e.g., standard Two-Phase algorithms) which may return suboptimal solution lengths, `RubikOptimal` implements Korf's IDA* (Iterative Deepening A*) algorithm (Korf, 1997) with pruning tables. This guarantees that for any state s, the returned distance is the theoretical minimum (God's Number):

$$d(s) = \min\{|P| : P \text{ transforms } s \to s_{\text{solved}}\}$$

where $|P|$ denotes the path length in FTM.

B.2.2 Metric Definition (Face Turn Metric)

The benchmark strictly adheres to the Face Turn Metric (HTM/FTM). The solver is configured to treat the following 18 moves as atomic actions with a cost of 1:

- **Clockwise:** U, D, F, B, L, R
- **Counter-Clockwise:** U', D', F', B', L', R'
- **Double Turns:** $U2, D2, F2, B2, L2, R2$

Slice moves (M, E, S) and wide moves (u, d, f, etc.) are not permitted as atomic actions and are decomposed into their constituent face turns for distance calculation.

B.2.3 Progress Computation

In the *Closed-Loop Step-by-Step* task, the oracle evaluates the transition from state s_t to s_{t+1} caused by a model's chosen action a_t. We determine if a step constitutes progress by comparing the exact oracle distances:

$$\text{IsOptimal}(a_t) \iff d(s_{t+1}) < d(s_t)$$

Since $d(s)$ is guaranteed optimal, any move satisfying this condition is mathematically proven to be on a shortest path to the solution.

B.3 Reproducibility and Seed Lists

To ensure exact reproducibility across all models and runs, we strictly enforce a deterministic generation protocol. For every episode i at depth d, the simulator is initialized with `seed = i`. This determines the scramble state $s_{d,i}$, the teacher inverse plan, and the distractor options.

We will provide the full metadata for these episodes in the code package. Future benchmarks can strictly replicate our test set by loading these pre-generated configurations or by re-running the generator with the same seed integers.

C Detailed Task Specifications and Fairness Control

C.1 Cube Face Reconstruction

C.1.1 Definition

Let I be the input image and $M^\star \in \mathcal{V}^{3\times 3}$ the ground-truth face matrix over vocabulary $\mathcal{V}$. The model outputs $\widehat{M} = f(I)$ after normalization $N(\cdot)$. Element-wise accuracy is

$$\text{Acc}_{\text{elem}} = \frac{1}{9}\sum_{i=1}^{3}\sum_{j=1}^{3} \mathbf{1}\{N(\widehat{M}_{ij}) = M^\star_{ij}\},$$

and matrix accuracy is $\text{Acc}_{\text{mat}} = \mathbf{1}\{N(\widehat{M}) = M^\star\}$. Parse-violations are counted when $N(\widehat{M}) \notin \mathcal{V}^{3\times 3}$ (wrong shape/tokens).

C.1.2 Generation & Fairness Controls

Images are generated with `VirtualCube` using a fixed camera pose, a standard color scheme, and uniform random cube states sampled by scramble depth d, ensuring that color-token priors do not deviate from a uniform distribution by more than 5% (maximum deviation < 0.05).. We disable motion blur and heavy lighting effects and clamp minor rotation jitter to reduce nuisance variance.

C.2 Cross-Modal Verification

C.2.1 Definition

Let I be the cube-net image, h the front-face hypothesis, and $y \in \{\text{YES}, \text{NO}\}$ the truth label computed from the rendering state. The model outputs $\hat{y} \in \{\text{YES}, \text{NO}\}$ after normalization. Balanced accuracy:

$$\text{BAcc} = \tfrac{1}{2}\big(\Pr[\hat{y} = \text{YES} \mid y = \text{YES}] + \Pr[\hat{y} = \text{NO} \mid y = \text{NO}]\big).$$

Parse-violations occur when the response lacks a valid `Yes/No` tag.

C.2.2 Generation & Fairness Controls

Hypotheses are auto-derived from ground-truth states produced by `VirtualCube`. We enforce a $50/50$ Yes/No prior per depth (and per template, when applicable), include polarity reversals (e.g., "is green"/"is not green"), and randomize surface forms to avoid lexical shortcuts. Ambiguous references are excluded.

C.3 Optimal Move Prediction (Static MCQ)

C.3.1 Definition

Let $T(s, a)$ be the transition and $d(\cdot)$ the fixed distance (FTM/HTM). For a one-move scramble $s = T(s^\star, a_s)$,

$$y = \arg\min_{a \in \{a_A, a_B, a_C, a_D\}} d(T(s, a))$$

$$\text{and at } d{=}1,\ y = a_s^{-1}.$$

A prediction $\hat{y} \in \{A, B, C, D\}$ is correct iff $\hat{y} = y$.

C.3.2 Generation & Fairness Controls

We fix `n_moves=1` so the gold action is the inverse of the scramble (*unique-optimal* under the chosen metric). Camera pose/colors match Recognition/Verification; the text serialization reuses the same normalization. Options are full-entropy shuffled; over many items this approximates uniform A-D placement.

C.4 Reflection-Guided Reasoning

C.4.1 Definition

Let $I = \{1, 2, \ldots, N\}$ be the set of N test items. For each item $i \in I$:

- s_i is the sample data (cube state, options A, B, C, D).
- $g_i \in \{A, B, C, D\}$ is the ground-truth correct (gold) answer.
- $\hat{y}_{i,1}$ is the initial "Draft" choice from the model.
- $\hat{y}_{i,2}$ is the final "Re-answer" choice from the model, after processing reflection r_i.
- $1\{\cdot\}$ is the indicator function.

We define the initial accuracy (score) $a_{i,1}$ and final accuracy $a_{i,2}$ for item i:

$$a_{i,1} = 1\{\hat{y}_{i,1} = g_i\} a_{i,2} = 1\{\hat{y}_{i,2} = g_i\}$$

To define EFR and OTR, we partition the total set I based on initial performance:

- **Initially Wrong Set (I_W):** $I_W = \{i \in I \mid a_{i,1} = 0\}$. Let $N_W = |I_W|$.
- **Initially Correct Set (I_C):** $I_C = \{i \in I \mid a_{i,1} = 1\}$. Let $N_C = |I_C|$.

(Note: $N_W + N_C = N$).

Initial Acc. The accuracy before reflection (from the "Draft" phase)[cite: 196].

$$\text{Initial Acc} = \frac{1}{N}\sum_{i=1}^{N} a_{i,1}$$

Final Acc (All). The accuracy after reflection and re-answering, measured over all N items[cite: 196].

$$\text{Final Acc (All)} = \frac{1}{N}\sum_{i=1}^{N} a_{i,2}$$

Gain (Δ). The net gain in accuracy after reflection[cite: 197].

$$\Delta = \text{Final Acc (All)} - \text{Initial Acc}$$

Error-Fix Rate (EFR). The fraction of previously incorrect items (I_W) that were corrected in the "Re-answer" phase[cite: 197].

$$\text{EFR} = \begin{cases} \frac{1}{N_W}\sum_{i\in I_W} a_{i,2} & \text{if } N_W > 0 \\ \text{NaN} & \text{if } N_W = 0 \end{cases}$$

Overthink Rate (OTR). The fraction of previously correct items (I_C) that were flipped to wrong in the "Re-answer" phase[cite: 198].

$$\text{OTR} = \begin{cases} \frac{1}{N_C}\sum_{i\in I_C}(1 - a_{i,2}) & \text{if } N_C > 0 \\ \text{NaN} & \text{if } N_C = 0 \end{cases}$$

C.4.2 Generation & Fairness Controls

Items are rebuilt deterministically via per-index seeds; A-D options are fully shuffled; the FTM/HTM distance metric is fixed per run. The re-answer prompt defines the deterministic tie-break (A > B > C > D) and the strict output format. We log tokens and latency for both reflection and re-answer.

C.5 Closed-Loop Step-by-Step

C.5.1 Definition

Let s_0 be the depth-d start state, T the transition, and $d(\cdot)$ the fixed metric. The teacher sequence $(a_1^{\text{teach}}, \ldots, a_d^{\text{teach}})$ is the inverse of the scramble. At step t the model outputs $a_t \in \{a_A, a_B, a_C, a_D\}$ (or `IDK`), the environment updates $s_t = T(s_{t-1}, a_t^\star)$ per policy, and $\Delta_t = d(s_t) - d(s_{t-1})$. The episode halts at the first t with $\Delta_t \geq 0$ or a parse violation; otherwise it runs up to d steps.

C.5.2 Generation & Fairness Controls

States come from `VirtualCube` with depth d. At each step we (i) compute the teacher move from the reversed scramble, (ii) draw 3 unique distractors from the legal move set, and (iii) *full-entropy* shuffle options A-D to dilute position bias. The distance metric (HTM/FTM) is fixed per run. If the parsed option is teacher or oracle-optimal, we apply it; if parsing fails, we apply a deterministic fallback (A) for progress checking and halt on non-progress.

C.6 Abstention (Selective Control)

C.6.1 Definition

We extend the closed-loop next-move task to allow an explicit abstention token `IDK`. Let s_0 be the depth-d start state, $d(\cdot)$ a fixed distance metric (FTM/HTM), and $(a_1^{\text{teach}}, \ldots, a_d^{\text{teach}})$ the teacher sequence (inverse scramble). At step t, the model outputs $y_t \in \{\text{A}, \text{B}, \text{C}, \text{D}, \texttt{IDK}\}$; the option letter maps to a concrete move $a_t \in \mathcal{M}$. The environment updates per the abstention *policy* (below), yielding s_t, and we define the per-step progress

$$\Delta_t \;=\; d(s_t) - d(s_{t-1}).$$

The episode halts at the first t with (i) non-progress, $\Delta_t \geq 0$, or (ii) a parse violation (no valid token). Otherwise it runs to d.

Primary scoring. *Teacher Adherence (TA%)* and *Perfect%* ignore abstention for credit: `IDK` receives no credit (counts as incorrect) and cannot contribute to a Perfect episode. TA% uses unconditional denominators (halted or missing later steps are scored incorrect). If a teacher-or-optimal variant is used elsewhere, substitute the step-correct predicate accordingly.

APA (Abstention-Penalized Accuracy). APA assigns partial credit $\lambda \in [0, 1]$ to abstentions (`IDK`) and full credit 1 to correct answers, with wrong answers receiving 0:

$$\text{APA}_\lambda = \frac{n_{\text{correct}} + \lambda\, n_{\text{IDK}}}{n_{\text{total}}}.$$

Here n_{total} counts all decisions (including parse failures), n_{IDK} counts abstentions, and n_{correct} counts correct answered moves. Properties: (i) APA_0 reduces to standard micro-accuracy that treats `IDK` as incorrect; (ii) $\text{APA}_1 = 1 - \frac{n_{\text{wrong}}}{n_{\text{total}}}$ (abstentions incur no penalty); (iii) it decomposes via

coverage ($c = \frac{n_{\text{answered}}}{n_{\text{total}}}$) and selective accuracy (sel-acc $= \frac{n_{\text{correct}}}{n_{\text{answered}}}$) as

$$\text{APA}_\lambda \;=\; c \cdot \text{sel-acc} \;+\; (1-c)\cdot\lambda.$$

Unless stated otherwise, we set λ=0.25.

Selective metrics. Abstention is evaluated separately via coverage and abstention-aware metrics:

$$\text{coverage} = \frac{n_{\text{answered}}}{n_{\text{total}}},$$
$$\text{sel-acc} = \frac{n_{\text{correct}}}{n_{\text{answered}}},$$
$$\text{APA}_\lambda = \frac{n_{\text{correct}} + \lambda n_{\text{IDK}}}{n_{\text{total}}}.$$

Here n_{answered} counts {A,B,C,D} responses; n_{IDK} counts abstentions; n_{total} counts all decisions (including parse failures). Unless stated, we use λ=0.25.

Parsing & error model. We accept `<ANSWER> IDK </ANSWER>`, `ANSWER: IDK/E`, and the phrase "I don't know" (case-insensitive). Any other malformed output is a parse failure (`MISSING`) and is treated as an incorrect answer for TA%, contributes to n_{total}, and halts the episode.

Policies on abstain. We support two deterministic policies:

- `teacher_on_abstain` (default): apply a_t^{teach} when y_t=`IDK` to preserve episode length; no credit is awarded.
- `skip_item`: halt the episode immediately when y_t=`IDK`.

Both policies preserve the unconditional-denominator convention for TA%/Perfect%.

C.6.2 Generation & Fairness Controls

Prompting. When abstention is enabled, the system prompt adds `IDK` as a fifth option and enforces a strict output tag (`<ANSWER> X </ANSWER>`). Tie-breaking and decision rules are otherwise unchanged.

Decoding. Greedy decoding (`temperature=0.0`) with a hard cap on new tokens (sufficiently large to avoid truncation $\approx 2^{16}$). Outputs are case-folded and whitespace-stripped before parsing.

Model	Coverage %	Sel. Acc %	IDK %	APA (s=0.25)
Qwen2.5 -VL 32B	99.29	23.74	0.00	0.24
Gemma 3 -27B	100.00	17.32	0.00	0.17
Gemini 2.5 Pro	87.50	28.57	12.50	0.28

Table 6: Abstention-aware selective metrics for STEP-BYSTEPTEST with IDK enabled and teacher on abstain at $d = 5$.

Option set construction. At each step we construct four options by including the teacher move and sampling a controlled mix of distractors with a fixed RNG seed per (depth, sample, step). Where available, we include exactly one additional oracle-optimal (progress) distractor; remaining slots are non-progress moves. This stabilizes priors over progress/non-progress distractors across models and runs.

Randomness control. We use a deterministic seed derived from $(d, \text{sample_id}, t)$ for option shuffling and distractor selection, ensuring identical option sets for all models.

Confidence gates (optional). A global confidence threshold can be specified; if a model's self-estimated confidence drops below the threshold, `IDK` is permitted. Thresholds are fixed per run and reported.

Logged configuration. We log (*i*) policy (`teacher_on_abstain` or `skip_item`), (*ii*) λ for APA, (*iii*) any confidence threshold, (*iv*) coverage by step, and (*v*) selective counts $(n_{\text{correct}}, n_{\text{wrong}}, n_{\text{IDK}}, n_{\text{total}})$.

C.6.3 Results

Brief findings. Gemini 2.5-Pro shows the highest precision when it chooses to act (*Sel. Acc* 28.6%) but at the cost of lower coverage (87.5%) driven by explicit abstention (`IDK` 12.5%); with λ=0.25 this yields the best penalty-aware score (APA 0.28). Qwen2.5-VL-32B and Gemma 3-27B almost never abstain (IDK 0%), maintain near-total coverage (99.3-100%), but achieve lower selective accuracy (23.7% and 17.3%), resulting in lower APA (0.24 and 0.17). Overall, this suggests a trade-off: calibrated abstention can modestly improve penalty-aware accuracy by avoiding low-confidence moves, whereas always answering preserves coverage but depresses precision. *Note:* Gemini's row is from a smaller, deeper run (5-move scramble, few episodes), so variance is higher and comparability is limited.

C.7 Causal Move-Effect Probe

C.7.1 Definition

Let $d(\cdot)$ be fixed (FTM via Optimal Oracle i.e., IDA* + pattern match heuristics), T the transition. For any move a:

$$\Delta(a) = d(T(s,a)) - d(s),$$

$$y(a) = \begin{cases} \text{DEC} & \Delta(a) < 0 \text{ or } T(s,a) \text{ solves}, \\ \text{NC} & \Delta(a) = 0, \\ \text{INC} & \Delta(a) > 0. \end{cases}$$

Given options $\{a_A, a_B, a_C, a_D\}$, the model outputs $\hat{y}_K \in \{\text{DEC}, \text{NC}, \text{INC}\}$ for each $K \in \{A, B, C, D\}$. Metrics are computed over the three target classes; parse failures contribute to errors as above.

C.7.2 Generation & Fairness Controls

Option construction (A–D) with fairness. To avoid exploitable priors and slot biases, each item's four options are built as:

1. **One-each + feasible double.** Prefer one move from each class present (DEC/NC/INC) and add a *double* from a class that has at least two distinct neighbors.

2. **Depth-aware target mix.** For depth d, estimate the feasibility of doubling class c as

$$\hat{p}_2(c \mid d) = \frac{\sum_{i:d_i=d} \mathbf{1}\{|\text{bucket}_c^{(i)}| \geq 2\} + \alpha}{\sum_{i:d_i=d} 1 + 2\alpha},$$

   where $\alpha = 1$. The per-depth *target mix* is 0.25 for each class (one slot) plus the extra 0.25 allocated proportionally to $\hat{p}_2(\cdot \mid d)$; we pick the doubled class that is most underrepresented vs. this target.

3. **Slot balancing.** The slot holding the doubled class rotates (A→B→C→D), and the remaining moves are greedily assigned to flatten historical slot×class frequencies.

C.8 Learning Curve (Closed-Loop MCQ)

C.8.1 Definition

Let $\mathcal{S}$ be cube states, $\mathcal{M}$ the standard move set (quarter/half turns in Singmaster), and $m(s)$ the deterministic transition obtained by applying $m \in \mathcal{M}$ to $s \in \mathcal{S}$. A fixed oracle distance $d : \mathcal{S} \to \mathbb{N}_0$ (FTM/HTM; chosen once per run) gives the *distance-to-solved*. The start state s_0 is produced by a depth-d scramble with per-episode seed; the *teacher plan* π_0 is the inverse scramble (shortest plan from s_0 to solved), with head $m^\star$.

Progress-making moves. For state s, the progress set $G(s)$ is

$$\{\, m \in \mathcal{M} \;:\; d(m(s)) < d(s) \;\text{ or }\; m(s) \text{ is solved} \,\}.$$

An option is *progress-making* iff it is in $G(s)$.

MCQ construction. At attempt t, from state s_t we construct an MCQ with four distinct options: one option is sampled from $G(s_t)$ (when $G(s_t) \neq \varnothing$), and the remaining three are sampled from $\mathcal{M} \setminus G(s_t)$ when feasible; options are then uniformly shuffled into A-D using an RNG independent of the scramble seed. (If $|\mathcal{M} \setminus G(s_t)| < 3$, a rare fallback samples without the progress/non-progress restriction; evaluation remains defined.)

Parsing and attempts. The model outputs a single letter $X_t \in \{\text{A}, \text{B}, \text{C}, \text{D}\}$ using either `<ANSWER>X</ANSWER>` or `ANSWER: X` (case-insensitive). A *parse-fail* is recorded if neither pattern is found. An *attempt* is counted for every query (including parse-fails).

Accept-progress control policy. Let $m^\star$ be the head of π_t (teacher next move), and let a_t be the concrete move mapped from the chosen option X_t (undefined on parse-fail). The environment update is:

Let $m^\star$ be the head of π_t (teacher next move), and let a_t be the concrete move mapped from the chosen option X_t (undefined on parse-fail). Writing $s'_t = a_t(s_t)$ for brevity, the environment update is:

$$(s_{t+1}, \pi_{t+1}) = \begin{cases} (m^\star(s_t),\, \text{tail}(\pi_t)) & a_t = m^\star \\ (s'_t,\, \text{Solve}(s'_t)) & a_t \in G(s_t) \\ (s'_t,\, a_t^{-1} \cdot \pi_t) & a_t \notin G(s_t) \\ (s_t,\, \pi_t) & \text{parse-fail.} \end{cases}$$

Here, $a_t^{-1} \cdot \pi_t$ denotes prepending the group-theoretic inverse of a_t to the plan π_t (equivalently $\langle a_t^{-1} \rangle \| \pi_t$). Here, $\text{Solve}(\cdot)$ is the oracle shortest-plan solver (returns an empty plan at solved), $\langle a_t^{-1} \rangle$ is the one-move deque containing the group-theoretic inverse of a_t, and $\|$ denotes deque concatenation (push-front).

Depth (d)	Closed-Loop Horizon	Recovery Budget
1	1 step	4
2	2 steps	5
3	3 steps	6
4	4 steps	7

Table 7: Task parameters by depth. The **Closed-Loop Horizon** defines the maximum steps allowed in the standard Step-by-Step task (strictly equal to d). The **Recovery Budget** defines the maximum additional moves allowed to solve the cube after the first deviation in the Learning Curve test.

Stopping rule and outcomes. An episode *solves* if $d(s_t) = 0$ for some $t \leq$ `max_attempts`; otherwise it is a *failure*. The number of attempts used is Let $K =$ `max_attempts` denote the attempt budget. The number of attempts used is

$$T = \begin{cases} \min\{t : d(s_t) = 0\} & t \leq K \\ \infty & \text{otherwise.} \end{cases}$$

For reporting, the solved-attempts histogram is $H(k) = \big|\{\text{ episodes with } T = k\}\big|$. Episode-level proportions such as $P(1)$ and $P(\leq 3)$ use the unconditional denominator (all N episodes; unsolved contribute 0). The *Avg@All* statistic treats failures as `max_attempts` when averaging attempts across episodes; *Med@Solved* is the median of T over solved episodes only (NA if none).

C.8.2 Generation & Fairness Controls

Deterministic scrambles & shared decoding. Each episode i is generated with a fixed scramble depth d and a deterministic seed equal to the episode index (`random_seed=idx`). All models use the same prompt, renderer settings, temperature $= 0.0$, and a large `max_new_tokens`, ensuring identical inputs and decoding policy.

Canonical option set, with a minimal fallback. From state s_t, we compute the *progress set*

$$G(s_t) = \{\, m : \; d(m(s_t)) < d(s_t) \;\text{ or } m(s_t) \text{ is solved} \,\},$$

using the same oracle distance $d(\cdot)$ as for scoring. When feasible, we build a *canonical* MCQ with exactly one progress-making option sampled from $G(s_t)$ and three distinct distractors sampled from $\mathcal{M} \setminus G(s_t)$ (never equal to the correct move). If $|\mathcal{M} \setminus G(s_t)| < 3$ (rare), we use a *fallback* that samples the remainder from $\mathcal{M} \setminus \{\text{correct}\}$ without enforcing progress/non-progress; options are still unique and evaluable.

Uniform letter shuffling, independent RNG. Options are uniformly permuted into A-D using a full-entropy RNG (System Random) that is independent of both the scramble seed and episode index. This prevents fixed letter positions from correlating with correctness and reduces positional cueing. We do not otherwise regularize or audit slot frequencies; independence plus sample size keeps slot priors approximately flat in expectation.

Parse protocol & penalties. The model must output exactly one letter via either `<ANSWER>X</ANSWER>` or `ANSWER: X` (case-insensitive), $X \in \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$. Any other output is a *parse-fail*. Parse-fails count as attempts and leave the environment unchanged, applying a consistent penalty without introducing spurious transitions.

Fixed control policy (accept-progress). The same policy is applied to every model's choice a_t: (i) if a_t matches the teacher head, apply and advance the plan; (ii) else if $a_t \in G(s_t)$, apply and replan from the new state; (iii) else apply and prepend a_t^{-1} to the plan. This creates a uniform opportunity to recover from near-misses and a uniform consequence for regressions.

Attempts budget parity & stopping. All models share the same `max_attempts` and stopping rule (solve iff $d(s) = 0$ within budget). Aggregate metrics (SR, $P(1)$, $P(\leq 3)$, Med@Solved, Avg@All) are computed over the same episode set with the unconditional denominator, so unsolved runs contribute 0 to proportion metrics and `max_attempts` to Avg@All.

Scope of randomness. We intentionally resample distractors each attempt (subject to the uniqueness and canonical/fallback rules) rather than fixing distractor identities across episodes or models. This avoids overfitting to specific move tuples while keeping evaluation comparable through shared scrambles, shared decoding, and a fixed control policy.

D Detailed Prompts

D.1 Cube Face Reconstruction

System Prompt

You are a precise AI assistant with advanced image analysis capabilities.
The official Rubik's Cube sticker colors and their RGB values are:

- White: (255,255,255)
- Yellow: (255,255,0)
- Red: (255,0,0)
- Orange: (255,128,0)
- Blue: (0,0,255)
- Green: (0,255,0)

When given the cube-net image:

1. Locate the 'Front' face (center of the net).
2. Extract its 3*3 sticker colors, row by row.
3. Verify each cell by cross-referencing its RGB against the table above.
4. Output strictly in this format:
   ANSWER:
   Row 1: [Color1, Color2, Color3]
   Row 2: [Color4, Color5, Color6]
   Row 3: [Color7, Color8, Color9]
5. Finally, append exactly this line:
   Answer verified for correctness.

Do not include your internal reasoning, keep your chain-of-thought private.

User Prompt

Here is the image of the Rubik's Cube laid out in a net diagram. The front face is labeled as 'Front'. Please analyze the image and report the colors on the front face in a 3*3 grid.

D.2 Cross-Modal Verification

System Prompt

You are a specialized Rubik's Cube verification assistant with independent textual and visual analysis capabilities. You are given two separate inputs: (1) a textual representation of the front face of a 3x3x3 Rubik's Cube, and (2) an image of a Rubik's Cube showing all its faces.
Tasks:

1. Independently identify the front face in the image.
2. Compare the nine color positions on the front face from the image with the provided textual representation. Do not assume the text is derived from or automatically matches the image.
3. If all nine positions match exactly in color, respond with Answer: Yes. If any position does not match, respond with Answer: No.

Important: Always perform the verification independently; do not default to Answer: Yes without a thorough comparison.

User Prompt

Inputs

1. A textual representation of the front face of a 3x3x3 Rubik's Cube: {front_face}
2. An image of a Rubik's Cube showing all its faces.

Task

- Determine the front face of the Rubik's Cube in the image.
- Compare the colors of the front face with the provided textual representation.
- Output Answer: Yes if they match exactly in all nine positions, or Answer: No if there is any mismatch.

D.3 Optimal Move Prediction

D.3.1 Image + Text Prompt

System Prompt

You are an expert Rubik's Cube solver. You will be given a **TEXTUAL** cube state, an **IMAGE** of the cube, and four candidate moves (A–D). The cube is exactly 1 move from solved. Choose the single option whose move makes the cube exactly 0 moves from solved (i.e., solves it).

CONTRACT (read carefully):

- Exactly ONE of A/B/C/D is correct. Never answer AMBIGUOUS.
- Output must be **EXACTLY** one of A, B, C, or D wrapped in <ANSWER> tags. Nothing else.
- Your entire output MUST match this regex: ^\\s*<ANSWER>\\s*[ABCD]\\s*</ANSWER>\\s*$
- Do NOT include explanations, move strings (e.g., R, U2, R'), or any extra text inside or outside the tags.

Notation reminder:

- X = 90° clockwise quarter-turn of face X.
- X' = 90° counter-clockwise quarter-turn of face X.

- X2 = a single 180° half-turn of face X (not two 90s).

Grounding & authority (internal):

- Faces are identified by their centers (isomorphic color scheme).
- If TEXT and IMAGE disagree, the **TEXTUAL** state is authoritative; the IMAGE is for reference only.

Parsing rules (internal; do not restate):

- If the textual state is an ASCII net with tokens [w,y,o,r,g,b], map tokens via centers: w→white, y→yellow, o→orange, r→red, g→green, b→blue. Do NOT invent or output Kociemba strings.
- If the textual state is already URFDLB facelets, use it directly.

Decision procedure (internal only):

1. Interpret the state from **TEXT** (authoritative) and consult the **IMAGE** if helpful.
2. Mentally apply each candidate (A–D) to this state.
3. Select the single candidate that yields the solved cube (distance = 0).
4. If a tie somehow occurs, break ties by letter: $A \prec B \prec C \prec D$.
5. Think silently; do not reveal reasoning.
6. Output only the LETTER in the required format.

User Prompt

Textual Cube State (authoritative):
{textual_representation}
Candidate moves:
A: {move_A}
B: {move_B}
C: {move_C}
D: {move_D}
Reply with exactly one line:
<ANSWER> A/B/C/D </ANSWER>

D.3.2 Text Only Prompt

System Prompt

You are an expert Rubik's Cube solver. You will be given a **TEXTUAL** cube state and four candidate moves (A–D). The cube is exactly 1 move from solved. Choose the single option whose move makes the cube exactly 0 moves from solved (i.e., solves it).

CONTRACT (read carefully):

- Exactly ONE of A/B/C/D is correct. Never answer AMBIGUOUS.
- Output must be **EXACTLY** one of A, B, C, or D wrapped in <ANSWER> tags. Nothing else.
- Your entire output MUST match this regex: ^\\s*<ANSWER>\\s*[ABCD]\\s*</ANSWER>\\s*$
- Do NOT include explanations, intermediate moves, or any other text.

Notation reminder:

- X = 90° clockwise quarter-turn of face X.
- X' = 90° counter-clockwise quarter-turn of face X.
- X2 = a single 180° half-turn of face X (not two 90s).

Grounding & authority (internal):

- The **TEXTUAL** state is the single source of truth. Faces are identified by their centers.

Parsing rules (internal; do not restate):

- If the textual state is an ASCII net with tokens [w,y,o,r,g,b], map tokens via centers: w→white, y→yellow, o→orange, r→red, g→green, b→blue. Do NOT invent or output Kociemba strings.
- If the textual state is already URFDLB facelets, use it directly.

Decision procedure (internal only):

1. Interpret the state from TEXT using the center mapping.
2. Mentally apply each candidate (A–D) to this state.
3. Select the single candidate that yields the solved cube (distance = 0).
4. If a tie somehow occurs, break ties by letter: $A \prec B \prec C \prec D$.
5. Think silently; do not reveal reasoning.
6. Output only the LETTER in the required format.

User Prompt

Textual Cube State (ground truth):
{textual_representation}
Candidate moves:
A: {move_A}
B: {move_B}
C: {move_C}
D: {move_D}
Reply with exactly one line:
<ANSWER> A/B/C/D </ANSWER>

D.3.3 Image Only Prompt

System Prompt

You are an expert Rubik's Cube solver. You will be given an **IMAGE** of the cube and four candidate moves (A–D). The cube is exactly 1 move from solved. Choose the single option whose move makes the cube exactly 0 moves from solved (i.e., solves it).

CONTRACT (read carefully):

- Exactly ONE of A/B/C/D is correct. Never answer AMBIGUOUS.
- Output must be **EXACTLY** one of A, B, C, or D wrapped in <ANSWER> tags. Nothing else.
- Your entire output MUST match this regex: ^\\s*<ANSWER>\\s*[ABCD]\\s*</ANSWER>\\s*$
- Do NOT include explanations, move strings, or any other text.

Notation reminder:

- X = 90° clockwise quarter-turn of face X.
- X' = 90° counter-clockwise quarter-turn of face X.
- X2 = a single 180° half-turn of face X (not two 90s).

Grounding & authority (internal):

- No textual state is provided. Infer the state from the **IMAGE** only. Faces are identified by their centers; the cube is solved when every face matches its center color.

Parsing rules (internal; do not restate):

- None (no textual state). Use the pixels only.

Decision procedure (internal only):

1. Infer the current state from the IMAGE using center colors.
2. Mentally apply each candidate (A–D) to this state.
3. Select the single candidate that yields the solved cube (distance = 0).
4. If a tie somehow occurs, break ties by letter: $A \prec B \prec C \prec D$.
5. Think silently; do not reveal reasoning.
6. Output only the LETTER in the required format.

User Prompt

Candidate moves:
A: {move_A}
B: {move_B}
C: {move_C}
D: {move_D}
Reply with exactly one line:
<ANSWER> A/B/C/D </ANSWER>

D.4 Reflections

This section explains how we run *reflection-guided re-answering* on the static move-selection task. Each reflection trial begins from the same **Image+Text** prompt we use for move prediction test (Prompt D.3.1). Concretely, every item shows (i) a rendered image of the cube for context and (ii) the authoritative textual cube state, along with four candidate move.

D.4.1 Guided Unredacted Reflection

System Prompt

You are an expert Rubik's Cube solver. You previously answered the following multiple-choice question incorrectly.Your task is to reflect on your mistake, understand why it occurred, and improve your reasoning for the future.
Complete the following steps clearly and concisely:

1. Explain why you answered incorrectly.
2. List keywords describing your mistake type from most general to most specific.
3. Solve the problem again step-by-step using the correct answer provided.
4. Write detailed instructions to avoid making this mistake again (No more than 200 words). Provide general advice to improve your (Large Language Model) performance on similar problems in the future.

User Prompt

The cube's textual state was: {cube_state}. An image of the cube in this state is attached. The cube was only one move away from being solved.
The four options were:
A: {option_A}
B: {option_B}
C: {option_C}
D: {option_D}

You chose the option: {model_choice}, which did not solve the cube. The correct solution was option: {correct_answer}.

D.4.2 Guided Redacted Reflection

System Prompt

You are an expert Rubik's Cube solver. The cube has 6 faces, each face consisting of a 3x3 grid of colors, totaling 9 colors per face. The faces are labeled as follows: Front (F), Back (B), Left (L), Right (R), Up (U), and Down (D). On each face, colors are arranged in 3 rows, each containing 3 colors. Colors are read from left to right within each row, starting from the top row and proceeding downward. Thus, each face has exactly 9 colors in a fixed order.
You previously answered the following multiple-choice question incorrectly.
Your task is to reflect on your mistake, understand why it occurred, and improve your reasoning for the future.
Complete the following steps clearly and concisely:

1. Explain why you answered incorrectly.
2. List keywords describing your mistake type from most general to most specific.
3. Solve the problem again step-by-step using the correct answer provided.
4. Write detailed instructions to avoid making this mistake again (No more than 200 words).
5. Provide general advice to improve your (Large Language Model) performance on similar problems in the future.

User Prompt

The cube's textual state was: {cube_state}. An image of the cube in this state is attached.
The cube was only one move away from being solved.
The four options were:
A: {option_A}
B: {option_B}
C: {option_C}
D: {option_D}

You chose the option: {model_choice}, which did not solve the cube. The correct solution was option: [REDACTED]. Correct answer is hidden, to avoid answer leakage into the reflection.

D.5 Optimal-Move Adherence vs Depth

System Prompt

You are an expert Rubik's Cube solver. Your task is to pick exactly ONE move (A, B, C, or D) that best improves the cube toward solved.
Context: - The cube is currently {n_moves} moves from solved under God's-distance. - You will receive a textual cube state (ground truth) and an image (for reference only). Ignore any discrepancies: the text is authoritative.
Decision Rule (no ambiguity allowed):

1. For each candidate (A,B,C,D), internally simulate applying the move to the textual state and estimate the resulting God's-distance d_1.
2. Prefer the move with the **lowest** d_1 (equivalently, the largest decrease from the current distance).
3. If multiple candidates yield the same minimal d_1, break ties **deterministically by letter**: $A \prec B \prec C \prec D$.
4. Never output anything other than A/B/C/D.

Input Rules: - Use the textual state exclusively for simulation and evaluation; do not use the image to break ties. - Think step-by-step internally but do **not** reveal your chain-of-thought.
Output Format (STRICT): Return exactly one line containing only this XML tag:
<ANSWER> A/B/C/D </ANSWER>
No explanations, no extra text, no AMBIGUOUS.

User Prompt

The cube is {n_moves} moves away from being solved.
Textual Cube State (Ground Truth): {textual_representation}
Candidate Moves: A: {move_A}
B: {move_B}
C: {move_C}
D: {move_D}
Output exactly one line:
<ANSWER> X </ANSWER>

D.6 Causal Move-Effect Probe

System Prompt

You are an expert Rubik's Cube evaluator. For **each** option (A–D), predict how applying that move to the textual cube state changes God's-distance.
Rules

- Textual cube state is ground truth; ignore any images.
- Use face centers to map colors to faces.
- Parse moves in Singmaster notation: quarter X, inverse X', half $X2$; apply sequences left-to-right.
- Report only the sign: DECREASE, NO_CHANGE, or INCREASE.
- If a move is invalid or cannot be parsed, use NO_CHANGE.
- Think step-by-step internally but do not reveal reasoning.

Output (STRICT): exactly four lines, in order A,B,C,D:

|<A> DECREASE|NO_CHANGE|INCREASE </A>|
|<B> DECREASE|NO_CHANGE|INCREASE </B>|

|<C> DECREASE|NO_CHANGE|INCREASE </C>|
|<D> DECREASE|NO_CHANGE|INCREASE </D>|

User Prompt

Face Centers
U: {U_color} R: {R_color} F: {F_color}
D: {D_color} L: {L_color} B: {B_color}
Textual Cube State (Ground Truth)
{state_text}
Candidate Moves
A: {move_A}
B: {move_B}
C: {move_C}
D: {move_D}
Return exactly four lines (uppercase labels), in this order:

<A> DECREASE	NO_CHANGE	INCREASE </A>
<B> DECREASE	NO_CHANGE	INCREASE </B>
<C> DECREASE	NO_CHANGE	INCREASE </C>
<D> DECREASE	NO_CHANGE	INCREASE </D>

D.7 Recovery dynamics

System Prompt

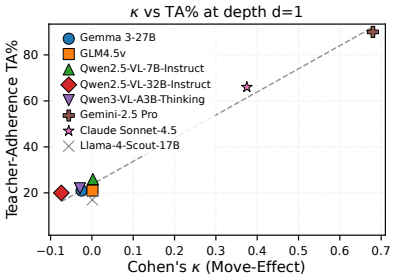
You are an expert Rubik's Cube solver. Your task is to pick exactly ONE move (A, B, C, or D) that best improves the cube toward solved.
Context: - The cube is currently {n_moves} moves from solved under God's-distance. - You will receive a textual cube state (ground truth) and an image (reference only). If they disagree, the text is authoritative.
Decision Rule (no ambiguity):

1. For each candidate (A,B,C,D), internally simulate applying the move to the textual state and estimate the resulting God's-distance d_1.
2. Choose the move with the **lowest** d_1 (largest decrease from current distance).
3. If multiple candidates tie at the minimal d_1, break ties **deterministically by letter**: $A < B < C < D$.
4. Never output anything other than A/B/C/D.

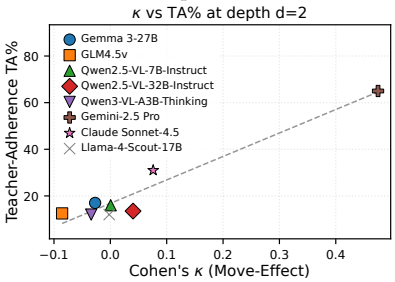
Input Rules: - Use the textual state exclusively for simulation and evaluation; do not use the image to break ties. - Think step-by-step internally but do **not** reveal your chain-of-thought.
Output Format (STRICT): Return exactly one line containing only this XML tag:
<ANSWER> A/B/C/D </ANSWER>
No explanations, no extra text, no ambiguous output.

User Prompt

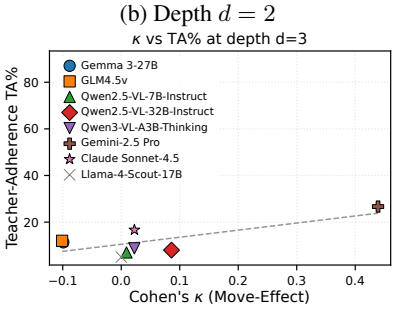
The cube is {n_moves} moves away from being solved.
Textual Cube State (Ground Truth): {textual_representation}
Candidate Moves: A: {move_A}
B: {move_B}
C: {move_C}
D: {move_D}
Output exactly one line:
<ANSWER> X </ANSWER>



(a) Depth $d = 1$



(b) Depth $d = 2$



(c) Depth $d = 3$

Figure 4: Move-Effect κ vs. Closed-Loop TA% at scramble depths $d \in \{1, 2, 3\}$; least-squares fit per panel. Pearson $r = 0.995$ at d=1.

E Additional Quantitative Results

E.1 Per-Depth Correlation Plots

To investigate the link between local causal understanding and sequential control, we plot the relationship between the *Causal Move-Effect* metric (Chance-corrected Cohen's κ) and *Closed-Loop Teacher Adherence* (TA%) across scramble depths $d \in \{1, 2, 3\}$.

As illustrated in Figure 4, the strength of this association varies significantly with task horizon:

- **Depth 1 (Strong Alignment):** At $d = 1$, we observe a near-perfect linear correlation ($r = 0.995$) between pre-action evaluation and control performance. This indicates that for single-step problems, a model's ability to forecast move consequences ($\kappa > 0$) directly translates to successful execution.

- **Depths 2 and 3 (Decoupling):** As the sequence length increases, the linear relationship weakens. At $d = 2$ and $d = 3$, open-weight models cluster near the origin (indicating chance-level performance on both metrics), while the frontier model (Gemini 2.5 Pro) degrades but remains distinct. This "collapse" suggests that while local causal understanding is a prerequisite for control, it is insufficient to prevent trajectory divergence in multi-step planning without robust error recovery mechanisms.

		Guided (Unredacted)				
Model	N	Init. (%)	Final (%)	Δ	EFR (%)	OTR (%)
Gemma 3-27B	100	29.0	81.0	+52.0	81.69	20.69
GLM-4.5v	25	36.0	100.0	+64.0	100.00	0.00
Llama 4-Scout 17B	50	10.0	100.0	+90.0	100.00	0.00
Qwen 2.5-VL-7B-Inst.	100	17.0	59.0	+42.0	59.04	41.18
Qwen 2.5-VL-32B-Inst.	100	22.0	70.0	+48.0	71.79	36.36
Qwen3-VL-A3B-Thinking	50	22.0	72.0	+50.0	71.79	27.27

Table 8: Guided (Unredacted) reflection on Optimal Move Prediction: the gold option is exposed in the reflection prompt, serving as an oracle upper bound. Columns and protocol match Table 4.

E.2 Unredacted Reflection Results

In the main text (§4.4), we presented "Guided (Redacted)" reflection, where the model must critique its own output without access to the ground truth. Here, we present the results of the "Guided (Unredacted)" ablation. In this regime, the reflection prompt explicitly includes the correct option as a hint, serving as an oracle upper bound to determine if models can recognize and justify the correct move when it is pointed out to them. Table 8 summarizes these findings. In stark contrast to the label-safe setting i.e., where gains were modest (+14% for Llama 4-Scout-17B) or negative (−20% for GLM-4.5v) the unredacted setting yields massive improvements across all architectures.

- **Perfect Rationalization:** GLM-4.5v and Llama 4-Scout-17B achieve **100% Final Accuracy** (gains of +64.0 and +90.0 points, respectively). This confirms that these models possess sufficient world knowledge to *verify* a solution when given the answer, even though they fail to *derive* it autonomously.

- **Broad Improvement:** Even the models that struggled most in the standard setting show strong recovery. Gemma 3-27B, which showed zero net gain in the redacted setting, jumps from 29% to 81% accuracy in the unredacted setting. Similarly, the Qwen family sees improvements ranging from +42 to

+50 points.

These results highlight the "Post-Hoc Rationalization" capabilities of MLLMs. The discrepancy between the Redacted and Unredacted scores indicates that the bottleneck in spatial reasoning is not the inability to understand the move mechanics, but the lack of a robust internal verification signal. When that signal is provided externally (the oracle label), models can successfully align their reasoning, but without it, they are prone to overthinking and instability.