BUTTONSBENCH: A CONTROLLABLE VISUAL REASONING BENCHMARK WITH CO-ORIGIN VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual puzzles offer an ideal testbed for visual reasoning due to their minimal knowledge priors and essential reliance on visual information . However, current evaluation practice on such puzzles exhibits two fundamental flaws: endpoint-only scoring inflates performance by masking unfaithful shortcuts behind illusory competence, while intuitive, non-parametric difficulty grading obscures models' concrete collapse thresholds and combinatorial limits. We present four contributions toward reliable evaluation: (1) BUTTONSBENCH-1, a diagnostic benchmark spanning 9 visual puzzle tasks parameterized by explicit difficulty knobs ("buttons") with discrete levels, transforming difficulty from a subjective tier into a measurable, programmatically grounded property; (2) **Co-Origin Probing Methodology**, an evaluation protocol that pairs base queries with decoupled intermediate behavioral checkpoints under majority consensus, formalizing verified joint accuracy ($\text{Acc}_{\text{joint}}$), falsification rate (FR), and structural crash points (CP); (3) **ButtonsBench Platform**, a modular three-stage system for on-demand programmatic generation of tailored visual materials, base questions, probes and final benchmark with ground-truth metadata to eliminate benchmark saturation and contamination; and (4) an audit of four frontier MLLMs across $1{,}594$ instances. We find that endpoint accuracy severely inflates perceived reasoning: probe auditing reveals widespread falsification rates (26.1% to 72.4%), overturning the apparent performance ranking by elevating `qwen-3.8-max` (46.1%) over `gpt-5.5` (45.1%). Furthermore, mapping multi-factor crash frontiers reveals a stark decoupling: while frontier models display high endurance along isolated 1D sweeps ($> L^{\max}$ on up to 11 slices), concurrent multi-factor scaling triggers catastrophic compositional collapse ($\text{CP}_{\text{all}} \leq 5$). The data of BUTTONSBENCH-1 benchmark and the source code of BUTTONSBENCH platform will be fully open-sourced upon publication.

1 INTRODUCTION

Visual puzzles serve as an ideal diagnostic for visual reasoning because they are inherently visual-dense and free of extensive domain knowledge: arriving at the solution strictly requires extracting and tracking fine-grained visual infomation, leaving little room for models to bypass visual perception via linguistic priors. However, existing puzzle benchmarks are predominantly static and one-off. Since their problems cannot be systematically varied or scaled from underlying materials, they quickly become "outdated" as model capabilities advance: saturated tasks lose discriminative power and are discarded in subsequent benchmarks. More critically, evaluating multimodal large language models (MLLMs) on such static puzzles exposes two fundamental evaluation deficiencies.

The first major deficiency lies in entangled variables and uncontrolled difficulty. In the majority of existing visual puzzle benchmarks, diverse visual conditions and structural factors are entangled within individual questions, leading to two major limitations: On the one hand, task difficulty is assigned by subjective human intuition rather than grounded in objective metrics. As models rapidly evolve, questions labeled 'difficult' by prior human intuition cease to pose a challenge, rendering such subjective difficulty gradings quickly obsolete. On the other hand, combinatorial problem spaces remain fundamentally uncharted. The factors governing difficulty are numerous and interact:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

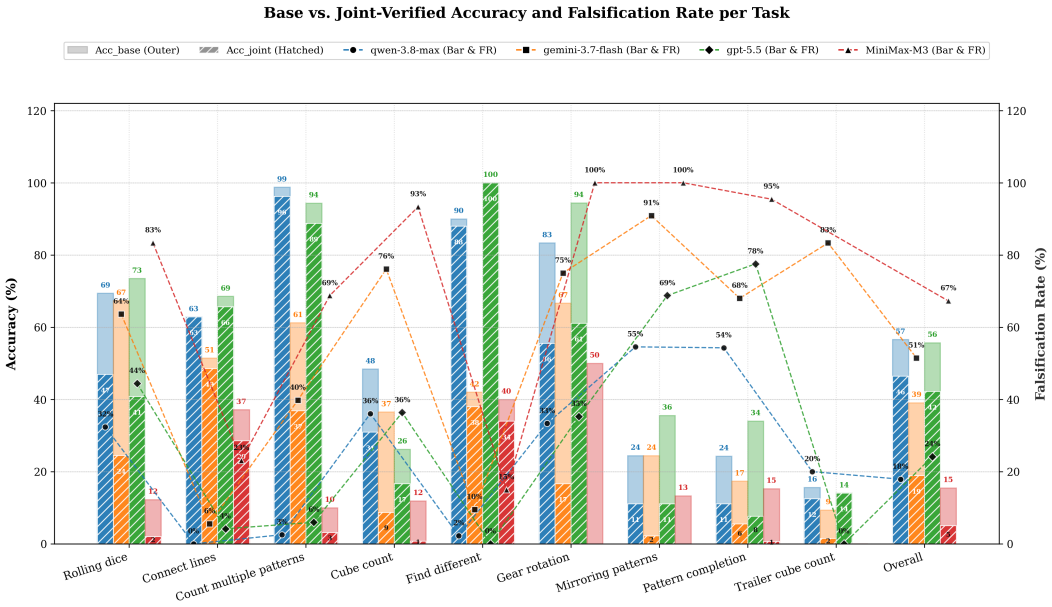


Figure 1: **Endpoint success masks severe reasoning collapse across multimodal tasks.** We contrast baseline endpoint accuracy (Acc_{base}, light outer bars) against probe-verified joint accuracy ($\text{Acc}_{\text{joint}}$, hatched inner bars) across four frontier MLLMs ($R = 5$ independent trials under majority consensus). The upper dashed curves quantify the Falsification Rate (FR, secondary axis). While static 2D search tasks (*e.g.*, Find different) exhibit high fidelity (FR $\leq 15\%$), dynamic state propagation tasks (*e.g.*, Gear rotation, Rolling dice) suffer catastrophic falsification (FR $> 35\%$, up to 100%), proving that conventional final-answer accuracy is heavily inflated by heuristic guessing.

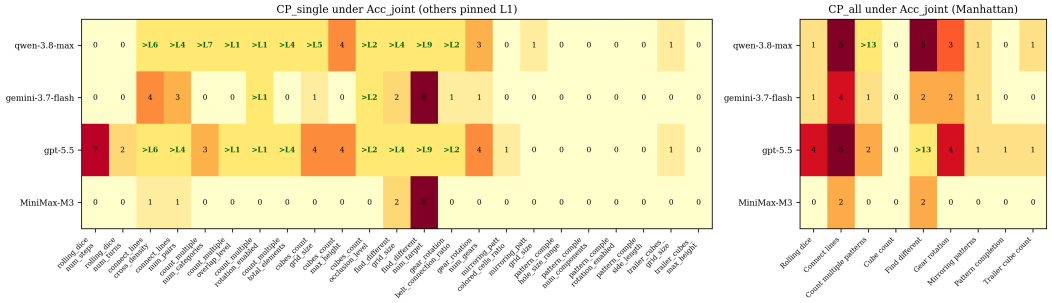


Figure 2: **Structural collapse frontiers on ButtonsBench-1.** We map the exact complexity thresholds (Crash Points) where verified deduction ($\text{Acc}_{\text{joint}}$) fails irreversibly. **(Left) 1D Sweeps ($\text{CP}_{\text{single}}$):** Frontier models exhibit widespread unsaturated resilience ($> L^{\max}$, gold cells) under isolated single-factor scaling, demonstrating the benchmark's procedural extensibility. **(Right) Combinatorial Sweeps (CP_{all}):** Concurrently perturbing multiple factors triggers catastrophic nonlinear collapse ($\text{CP}_{\text{all}} \leq 5$, dark cells), exposing severe compositional brittleness unobservable via conventional 1D evaluations.

changing one condition, combining a few, or scaling up the problem can turn an easy puzzle into a hard one. Manual authoring covers only a vanishing corner of this combinatorial space, leaving both the crash point and compositional generalization not merely unmeasured, but unmeasurable.

The second deficiency is that final-answer-only evaluation masks the absence of genuine reasoning behind a facade of illusory competence. A correct final answer can easily stem from lucky guessing, spurious visual cues, or shortcut heuristics. Relying on self-reported chains of thought (CoT) is insufficient, as models frequently produce unfaithful or post-hoc rationalizations (Lanham et al., 2023). Unlike text-based process supervision that merely inspects self-generated textual traces, be-

havioral checkpoints serve as a necessary falsification mechanism: stress-testing whether a model's step-by-step reasoning hold across intermediate checkpoints and contextual variants, thereby exposing brittle shortcuts.

To resolve these challenges, we present **ButtonsBench**, an extensible platform that programmatically synthesizes visual reasoning data to construct **ButtonsBench-1**. We anchored difficulty to quantitative structural parameters (*e.g.*, grid sizes, sequence lengths), which provides an objective measure of model capability scaling. We also mapped fine-grained crash points ($\mathrm{CP}_{\mathrm{single}}, \mathrm{CP}_{\mathrm{all}}$) with full factor sweeps, while on-demand parameter dial-up and material instances regeneration inherently resolve benchmark saturation and contamination. Furthermore, by logging complete state metadata during material synthesis, the platform automatically derives co-origin probes to audit latent reasoning fidelity, unmasking deceptive endpoint success without relying on unfaithful self-reported chains of thought.

In summary, our key contributions are fourfold:

- **A Procedural Benchmark Platform:**We designed BUTTONSBENCH, an extensible platform that decomposes visual reasoning data generation into a sequential three-stage pipeline: materials generation, questions construction , and exam assembly. This modular architecture enables scalable and customizable benchmark construction with deterministic ground-truth tracking, inherently mitigating both benchmark saturation and data contamination.
- **A Controlled Multi-Factor Benchmark:** We construct ButtonsBench-1, a calibrated diagnostic testbed comprising 1,594 procedurally synthesized question instances (including the basic question and the probe question) generated from 691 material instances across 9 visual puzzles. By systematically sweeping explicit factor levels, it enables fine-grained failure attribution; by coupling each puzzle instance with co-origin behavioral probes, it directly audits reasoning fidelity.
- **Co-Origin Evaluation Methodology:** We formulate an evaluation methodology using decoupled, multi-trial co-origin probes. Grounded in majority consensus, this protocol formalizes verified joint accuracy ($\mathrm{Acc}_{\mathrm{joint}}$), falsification rate (FR), and structural crash points (CP) to rigorously audit reasoning fidelity beyond endpoint success.
- **Diagnostic Empirical Findings:** A comprehensive behavioral audit across frontier MLLMs reveals two core phenomena: **(i) Unmasking Illusory Competence**: Conventional endpoint queries heavily mask defective reasoning through superficial guessing; probe verification strips away these unfaithful shortcuts and overturns benchmark rankings, with `qwen-3.8-max` surpassing `gpt-5.5` on verified joint accuracy. **(ii) 1D Parameter Endurance vs. Compositional Collapse**: While leading models exhibit high endurance along isolated parameter sweeps, concurrent combinatorial scaling triggers abrupt nonlinear collapse, demonstrating that single-variable robustness fails to generalize to coupled multi-factor environments.

2 THE BUTTONSBENCH PLATFORM

To construct diagnostic benchmarks that overcome the fundamental deficiencies of static datasets—namely, entangled variable conditions and illusory competence—we design and implement **ButtonsBench**, an extensible procedural platform for the scalable and customizable generation of visual reasoning data.

Rather than authoring evaluation problems monolithically—which irrevocably conflates visual rendering, task querying, and label assignment—the BUTTONSBENCH platform decomposes data construction into a sequential three-stage pipeline: **Material Generation**, **Question Construction**, and **Exam Assembly** (Figure 3). This decoupled yet phased workflow maximizes the diagnostic utility of underlying visual scenes, provides native support for generating paired co-origin probes, and enables programmatic ground-truth derivation with zero manual annotation. The details of each stage of the pipeline is elaborated in Appendix A.4.3

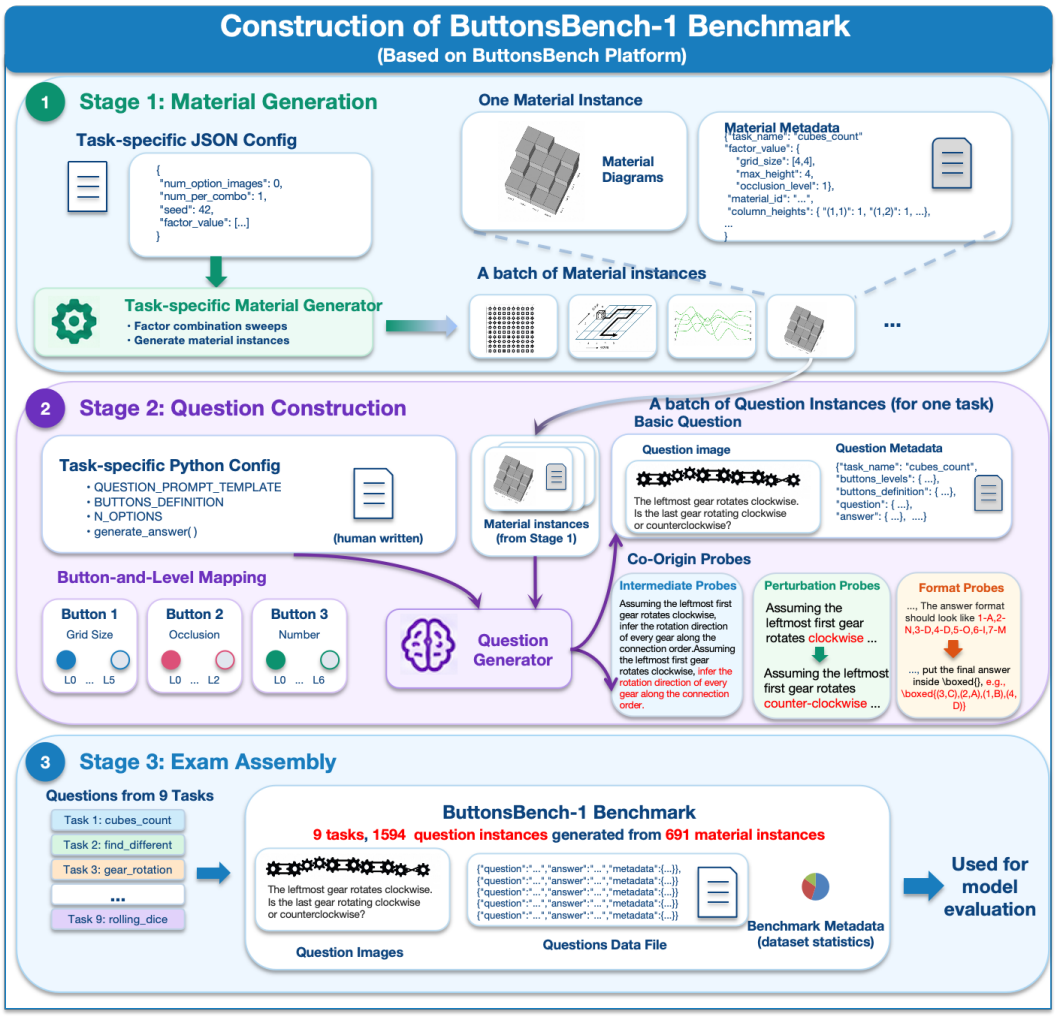


Figure 3: Construction of BUTTONSBENCH-1 benchmark based on ButtonsBench Platform.

3 BUTTONSBENCH-1 BENCHMARK DATASET

3.1 TASKS COMPOSITION

Table 1: Taxonomy and specifications of tasks in BUTTONSBENCH-1.

Core Domain	Task Identifier	Reasoning Target
Fine-Grained Discrim.	`find_different`	Locating outlier elements within dense 2D patterns
	`count_multiple_patterns`	Counting multi-class items under rotation & occlusion
	`pattern_completion`	Identifying pieces that complete structured holes
Spatial Geometry	`cubes_count`	Deducing 3D cube counts from isometric viewpoints
	`trailer_cubes_count`	Inferring 3D configurations via orthogonal projections
Pattern Trans.	`mirroring_patterns`	Inferring color-symmetric figures across reflection axes
Physical Rules	`gear_rotation`	Propagating kinematic rotations across meshed/belt gears
Topological	`connect_lines`	Tracing tangled continuous curves between endpoints
Sequential	`rolling_dice`	Tracking multi-step 3D face dynamics along grid paths

ButtonsBench-1 is built upon nine tasks, as shown in Figure 5. Originating from classic puzzle tasks including Babyvision (Chen et al., 2026) and MIRA (Zhou et al., 2026), the evaluation tasks in BUTTONSBENCH-1 collectively span six core visual reasoning domains, as summarized in Table 1.

These puzzles serve as an ideal testbed for multimodal evaluation. Their visual scenes are information-dense: arriving at the solution strictly demands reading fine-grained visual details from the pixels rather than exploiting linguistic priors. Crucially, each task can be governed by clear programmatic generation logic, enabling the platform to procedurally synthesize diverse material instances across scalable factor levels and derive paired co-origin probes.

3.2 BUTTONS

Every question in BUTTONSBENCH-1 is fully parameterized by task-specific *buttons*: independently adjustable factors that determine what the rendered diagram contains and how much reasoning it demands. A button takes one of an ordered set of discrete *levels* (e.g., Level 0 to Level 5), and each level maps to a concrete factor value consumed by the rendering and answer-derivation code. A question instance's full factor configuration is deterministically logged in its metadata as `button_levels`.

Appendix A.4.2 details the complete factor space across all nine tasks, enumerating each button alongside its underlying variable factor, assigned levels, factor value ranges, and resulting combinatorial coverage. These buttons are deliberately engineered to decouple visual conditions within each task.

3.3 COMBINATORIAL SWEEPS AND BENCHMARK STATISTICS

For each task, basic questions are curated through a combinatorial sweep across all factor levels: the platform enumerates every valid combination of button levels and generates exactly one basic question per combination (Table 5). Across all nine tasks, this yields a total of 691 basic questions. Every basic question instance explicitly records its button-level configuration in its metadata, which makes the single-factor control protocol directly applicable to downstream evaluation.

Beyond basic questions, each task is paired with dedicated *co-origin probes*. Each probe is constructed from the identical underlying *material instance* as its corresponding basic question, posing an alternative question derived programmatically from the scene's material metadata. As demonstrated in Appendix A.5, these 903 probes instantiate intermediate state queries (e.g., intermediate gear directions or coordinates of existing cubes), localized condition perturbations (e.g., reversing the driving gear's input direction), and format variants across the nine tasks.

The co-origin probes sweep the identical factor-level configurations as the basic questions, ensuring that each basic question is coupled with behavioral checkpoints instantiated on the exact same visual material instance.

In summary, BUTTONSBENCH-1 comprises 691 basic questions and 903 paired co-origin probes, totaling 1,594 evaluation question instances across the nine tasks (Table 4). Because every question is labeled with explicit button configurations and coupled with co-origin behavioral checkpoints, evaluation transcends traditional uninformative aggregate accuracy. Instead, researchers can perform fine-grained attribution across factor-level dimensions, pinpointing precise model crash points and falsifying spurious successes where ostensibly correct final answers collapse under intermediate checkpoints.

4 EVALUATION METHODOLOGY

4.1 METRICS

To move beyond uninformative aggregate scores and eliminate stochastic decoding variance, all models are evaluated across $R = 5$ independent trials for every question instance. Formally, let $\mathcal{T} = \{1, \ldots, |\mathcal{T}|\}$ index the 9 tasks ($|\mathcal{T}| = 9$), where each task $t \in \mathcal{T}$ comprises N_t unique material instances, yielding a total benchmark size of $N = \sum_{t \in \mathcal{T}} N_t = 691$ basic instances.

For the i-th material instance in task t ($i \in \{1, \ldots, N_t\}$), let $y^{\text{base}}_{t,i,r} \in \{0, 1\}$ denote whether the model correctly answers the basic question in the r-th trial ($r \in \{1, \ldots, R\}$). Similarly, if instance (t, i) is accompanied by $K_{t,i}$ co-origin probe questions, let $y^{\text{probe}}_{t,i,k,r} \in \{0, 1\}$ indicate correctness on the k-th probe question ($k \in \{1, \ldots, K_{t,i}\}$) during its corresponding r-th trial.

We first establish a deterministic mastery criterion using majority consensus to filter out accidental trial-level fluctuations. A question instance is considered mastered if it is solved in at least $\lceil R/2 \rceil = 3$ trials:

$$T^{\text{base}}_{t,i} = \mathbb{I}\left(\sum_{r=1}^{R} y^{\text{base}}_{t,i,r} \geq 3\right), \quad T^{\text{probe}}_{t,i,k} = \mathbb{I}\left(\sum_{r=1}^{R} y^{\text{probe}}_{t,i,k,r} \geq 3\right), \tag{1}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Composite probe mastery requires passing all designated behavioral checkpoints: $T^{\text{probe}}_{t,i} = \prod_{k=1}^{K_{t,i}} T^{\text{probe}}_{t,i,k}$.

Building upon these consensus states, we formulate our evaluation metrics across three hierarchical granularities: **Instance-level**, **Task-level**, and **Benchmark-level**.

Hierarchical Accuracy Metrics (Acc_{base} and $\text{Acc}_{\text{joint}}$). We distinguish between baseline endpoint accuracy (Acc_{base}) and probe-verified joint accuracy ($\text{Acc}_{\text{joint}}$), as shown in Table 2 and Appendix A.1.

Falsification Rate (FR). To unmask illusory competence hidden behind endpoint successes, FR quantifies the empirical conditional probability that seemingly mastered basic questions collapse under co-origin probe verification. It is computed at both task and benchmark levels:

$$\text{FR}^{(t)} = \frac{\sum_{i=1}^{N_t} \mathbb{I}\left(T^{\text{base}}_{t,i} = 1 \ \wedge \ T^{\text{probe}}_{t,i} = 0\right)}{\sum_{i=1}^{N_t} \mathbb{I}\left(T^{\text{base}}_{t,i} = 1\right)}, \quad \overline{\text{FR}} = \frac{\sum_{t \in \mathcal{T}} \sum_{i=1}^{N_t} \mathbb{I}\left(T^{\text{base}}_{t,i} = 1 \ \wedge \ T^{\text{probe}}_{t,i} = 0\right)}{\sum_{t \in \mathcal{T}} \sum_{i=1}^{N_t} \mathbb{I}\left(T^{\text{base}}_{t,i} = 1\right)}. \tag{2}$$

A high FR reveals that endpoint successes are deceptive—stemming from superficial heuristic shortcuts or random guessing rather than grounded visual reasoning.

Structural Crash Point Metrics (CP). To map the operational limits of MLLMs without relying on bounded normalizations—thereby preserving strict backward compatibility as benchmarks scale—we define crash points in the absolute coordinate space of discrete factor levels. For each task t, an instance corresponds to a discrete level vector $\mathbf{l} = (l_1, \ldots, l_{M_t}) \in \mathbb{N}^{M_t}$, where M_t is the number of configured buttons in task t. Because isolated failures may sporadically stem from localized rendering artifacts, we filter out *isolated flukes*: a zero-accuracy configuration $\text{Acc}(\mathbf{l}) = 0$ is considered isolated if all its immediate 1-step neighbors (differing by ± 1 level on any single button) have $\text{Acc} = 1$. All remaining zero-accuracy configurations constitute the persistent failure set $\mathcal{F}_t$.

Crash points are computed per task t under both evaluation criteria ($\text{Acc} \in \{\text{Acc}_{\text{base}}, \text{Acc}_{\text{joint}}\}$):

- **All-Buttons Crash Point ($\text{CP}^{(t)}_{\text{all}}$):** Reflecting the earliest collapse under multi-factor interaction, $\text{CP}^{(t)}_{\text{all}}$ is defined as the minimal button levels sum among all non-isolated failure configurations in task t:

$$\text{CP}^{(t)}_{\text{all}} = \min_{\mathbf{l} \in \mathcal{F}_t} \sum_{m=1}^{M_t} l_m. \tag{3}$$

  Higher values directly signify greater resilience against cumulative visual complexity.

- **Single-Button Crash Point ($\text{CP}^{(t,m)}_{\text{single}}$):** We also tried single-factor control by isolating button m in task t by anchoring all other buttons at their standardized baseline level (L1). Sweeping button m from L0 to its maximum level $L^{\max}_{t,m}$, $\text{CP}^{(t,m)}_{\text{single}}$ marks the first level where accuracy drops to zero for two consecutive levels:

$$\text{CP}^{(t,m)}_{\text{single}} = \min\left\{l \mid \text{Acc}(l) = 0 \ \wedge \ \text{Acc}(l+1) = 0\right\}. \tag{4}$$

Table 2: Hierarchical overview of evaluation metrics across instance, task, and benchmark granularities in BUTTONSBENCH-1.

Evaluation Granularity	Baseline Metric (Acc_{base})	Verified Joint ($\text{Acc}_{\text{joint}}$)	Falsification Rate (FR)	Structural Crash (CP)
1) Instance-level	$\text{Acc}_{\text{base}}(t,i) \in \{0,1\}$	$\text{Acc}_{\text{joint}}(t,i) \in \{0,1\}$	—	—
(Single instance)	Binary majority pass	Binary joint pass	—	—
2) Task-level	$\text{Acc}_{\text{base}}^{(t)}$	$\text{Acc}_{\text{joint}}^{(t)}$	$\text{FR}^{(t)}$	$\text{CP}_{\text{all}}^{(t)}$ / $\text{CP}_{\text{single}}^{(t,m)}$
(Within a task)	Mean over N_t instances	Mean over N_t instances	Probe failure rate given base correct	Discrete collapse frontiers
3) Benchmark-level	$\overline{\text{Acc}}_{\text{base}}$	$\overline{\text{Acc}}_{\text{joint}}$	$\overline{\text{FR}}$	$\overline{\text{CP}}_{\text{all}}$
(Overall benchmark)	Mean over 691 question groups	Mean over 691 question groups	Aggregate falsification rate over 691 groups	Mean across 9 tasks

Note: A "paired question group" refers to an evaluation unit comprising one basic question and all its paired co-origin behavioral probes derived from the identical material instance package. Across the 9 tasks, 691 paired question groups yield a total of 1,594 question instances.

If a model sustains performance without such consecutive failures up to $L_{t,m}^{\max}$, its threshold is designated as $> L_{t,m}^{\max}$. Across the overall benchmark, we report the task-averaged crash thresholds $\overline{\text{CP}}_{\text{all}} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \text{CP}_{\text{all}}^{(t)}$.

4.2 LLM-AS-A-JUDGE EVALUATION

We deployed an automated LLM-as-a-judge framework to parse model outputs. We employed `gemini-3.5-flash` as our judge model and instructed it to compare the extracted predicted final answers strictly against the labeled answer. The complete judge prompt template and its task-specific equivalence criteria are detailed in Appendix A.6.1.

5 EXPERIMENTS ON BUTTONSBENCH-1 BENCHMARK

We evaluated four frontier multimodal large language models (including `qwen-3.8-max`(Yang et al., 2025), `gemini-3.7-flash`(DeepMind, 2026), `gpt-5.5`(OpenAI, 2026), and `MiniMax-M3`(Lai et al., 2026)) on BUTTONSBENCH-1, covering all $N = 691$ paired question groups (1,594 instances). Under a strict fixed-denominator convention ($N = 691$), any evaluation query that fails to produce a parsable completion within 20 retries is scored as an explicit failure (0 accuracy). Crucially, we enforced two rigorous safeguards: **(i) Decoupled Single-Turn Probing**, where basic queries and co-origin probes are dispatched as isolated requests to preclude inter-turn information leakage, heuristic visual priming, or post-hoc self-correction; and **(ii) Consensus-Driven Multi-Trial Sampling**, where each instance is queried across $R = 5$ independent stochastic trials ($T = 0.7$) and aggregated via majority voting ($\geq 3/5$), filtering out accidental trial-level lucky guesses to capture the model's true, stable cognitive state.

5.1 THE ILLUSION OF ENDPOINT SUCCESS

Table 3 reports the comparative baseline consensus accuracy ($\overline{\text{Acc}}_{\text{base}}$), verified joint accuracy ($\overline{\text{Acc}}_{\text{joint}}$), and falsification rate ($\overline{\text{FR}}$) across all nine tasks.

Unmasking Superficial Endpoint Success. While conventional endpoint evaluation suggests that `gpt-5.5` and `qwen-3.8-max` are virtually neck-and-neck (60.1% vs. 57.4% macro $\overline{\text{Acc}}_{\text{base}}$), co-origin behavioral auditing substantially reverses this apparent hierarchy. Probe verification restores genuine model separation: on verified joint accuracy ($\overline{\text{Acc}}_{\text{joint}}$), `qwen-3.8-max` overtakes

Table 3: Comprehensive evaluation across 9 tasks on BUTTONSBENCH-1 (R=5 independent trials per instance under majority consensus). We report Baseline Consensus Accuracy (A_{base} ↑, %), Verified Joint Accuracy (A_{joint} ↑, %), and Falsification Rate (FR ↓, %). ↑ denotes higher is better; ↓ denotes lower is better. Bold values indicate the best verified joint performance per task ($N = 691$).

Task Name	qwen-3.8-max			gemini-3.7-flash			gpt-5.5			MiniMax-M3		
	A_{base} ↑	A_{joint} ↑	FR ↓	A_{base} ↑	A_{joint} ↑	FR ↓	A_{base} ↑	A_{joint} ↑	FR ↓	A_{base} ↑	A_{joint} ↑	FR ↓
Rolling Dice	69.4	**46.9**	32.4	67.3	24.5	63.6	73.5	40.8	44.4	12.2	2.0	83.3
Connect Lines	62.9	62.9	0.0	51.4	48.6	5.6	68.6	**65.7**	4.2	37.1	28.6	23.1
Count Patterns	98.8	**96.2**	2.5	61.2	36.9	39.8	94.4	88.8	6.0	10.0	3.1	68.8
Cube Count	48.4	**30.9**	36.1	36.5	8.7	76.1	26.2	16.7	36.4	11.9	0.8	93.3
Find Different	90.0	88.0	2.2	42.0	38.0	9.5	100.0	**100.0**	0.0	40.0	34.0	15.0
Gear Rotation	83.3	55.6	33.3	66.7	16.7	75.0	94.4	**61.1**	35.3	50.0	0.0	100.0
Mirroring Patterns	24.4	**11.1**	54.5	24.4	2.2	90.9	35.6	**11.1**	68.8	13.3	0.0	100.0
Pattern Completion	24.3	**11.1**	54.3	17.4	5.6	68.0	34.0	7.6	77.5	15.3	0.7	95.5
Trailer Cubes	15.6	12.5	20.0	9.4	1.6	83.3	14.1	**14.1**	0.0	0.0	0.0	—
Overall (Macro)	57.4	**46.1**	**26.1**	41.7	20.3	56.9	**60.1**	45.1	30.3	21.1	7.7	72.4

Note: Evaluated under an immutable master denominator policy ($N = 691$). Overall scores report unweighted task-level macro-averages (equal weight per task domain). Any unobserved instance or incomplete roll execution failing after 20 retries is strictly counted as failure (0). "—" denotes undefined falsification rates where the model scored zero baseline accuracy ($A_{\text{base}} = 0.0\%$), yielding an indeterminate zero denominator; macro-averaged FR for MiniMax-M3 is calculated over the 8 well-defined tasks.

`gpt-5.5` (46.1% vs. 45.1%) by maintaining a significantly lower falsification rate ($\overline{\text{FR}} = 26.1\%$ vs. 30.3%), confirming that co-origin probes expose latent internal reasoning consistency obscured by endpoint queries. Crucially, deceptive endpoint competence is pervasive across architectures; over half of the correct answers from `gemini-3.7-flash` and `MiniMax-M3` fail intermediate auditing ($\overline{\text{FR}} = 51.5\%$ and 67.3%, respectively), and even `qwen-3.8-max` experiences a 17.9% collapse in its baseline successes.

5.2 STRUCTURAL CRASH POINT FRONTIERS

The complete discrete crash frontiers ($\text{CP}_{\text{single}}$ and CP_{all}) across the active button parameter space under $\text{Acc}_{\text{joint}}$ are mapped in Figure 2. By contrasting isolated 1-Dimension sweeps against full combinations, two contrasting architectural regimes emerge:

Decoupling 1-Dimension Endurance from Multi-Factor Compositional Collapse. Contrasting single-button sweeps against combinatorial stress in Figure 2 reveals two critical insights into frontier model robustness. First, unsaturated 1D frontiers demonstrate benchmark extensibility (Figure 2, left). When perturbations are isolated along single factor axes ($\text{CP}_{\text{single}}$), frontier architectures display remarkable resilience: both `qwen-3.8-max` and `gpt-5.5` sustain unsaturated frontiers ($> L^{\max}$, gold cells) across 11 and 9 of the 23 task-button slices, respectively (*e.g.*, withstanding maximum target scaling beyond L9 in `find_different`). This empirical divergence validates the procedural extensibility of BUTTONSBENCH-1: unlike static benchmarks that rapidly saturate, our programmatic generation engine allows difficulty thresholds to be dialed up to track advancing model capabilities without re-authoring tasks. Conversely, we observe severe compositional collapse under joint multi-factor stress (Figure 2, right). Crucially, single-factor endurance fails to generalize to combinatorial environments. Across dynamic tracking (`gear_rotation`) and volumetric spatial reasoning (`cubes_count`), where individual dimensions exhibit decent survival thresholds ($\text{CP}_{\text{single}} \geq 4$), the joint Manhattan crash points abruptly contract to $\text{CP}_{\text{all}} \leq 5$. This nonlinear degradation indicates that contemporary MLLM deduction is fragile to factor coupling; while models can compensate for isolated visual noise or step extensions, their latent multi-step tracking completely shatters when visual and dynamical complexities scale concurrently.

Hard Walls vs. Unsaturated Frontiers. A *Hard Wall* ($\text{CP}_{\text{single}} = 0$, indicating consecutive failures beginning at L0) highlights tasks where model capabilities instantly dissolve. `MiniMax-M3` hits hard walls on 19 of 23 task-button slices, `gemini-3.7-flash` on 14, `qwen-3.8-max` on 8, and `gpt-5.5` on 6. In particular, `pattern_completion` and `trailer_cubes_count` (height) constitute universal hard walls across all four models. Conversely, *Unsaturated Frontiers* ($\text{CP} > L^{\max}$) mark factors where models withstand the maximum complexity evaluated. `qwen-3.8-max` remains unsaturated across 11 slices (including all 4 buttons in `count_multiple_patterns` and both in `connect_lines`), while `gpt-5.5` is unsaturated across 9. This asymmetry validates the *procedural extensibility* of BUTTONSBENCH-1: unlike static benchmarks that become obsolete once saturated, our procedural generation framework allows researchers to seamlessly dial up button ranges (e.g., scaling gear sequences from 50 to 100) to trace future model frontiers without benchmark redesign.

Verification Layer Shifts the Collapse Frontier. Auditing models via joint consensus significantly contracts their perceived resilience. In `gear_rotation`, `gpt-5.5` appears unsaturated across all button dimensions under baseline evaluation ($\text{CP}_{\text{all}} > 7$), yet plummets to $\text{CP}_{\text{all}} = 4$ under joint verification. Similarly, `qwen-3.8-max`'s crash point in `rolling_dice` regresses from complexity level 11 to 1. Traditional benchmarks substantially overestimate the operational safety margins of vision-language models.

Further results of domain-specific failure analysis are demonstrated in Appendix A.2.1.

6 RELATED WORK

Visual Reasoning Benchmark An increasing body of multimodal benchmarks focuses on the foundational visual perception and visual reasoning capabilities of MLLMs. One line of work focused on visual puzzles and curates datasets by manually collecting and annotating problems from public children's puzzle challenges, logic reasoning books, IQ tests and early-grade mathematics competitions, or via handcrafted expert design (Chen et al., 2026; Zhou et al., 2026; Yan et al., 2025; He et al., 2026; Xu et al., 2025; Khezresmaeilzadeh et al., 2026). Although these benchmarks encompass a wide spectrum of puzzle types, each individual task typically offers only a sparse number of instances alongside an uneven difficulty distribution. Another line of research investigates domain-specific visual perception and reasoning faculties, establishing dedicated evaluations for spatial perception (Rim et al., 2026), topology (Maniparambil et al., 2026; Mohseni et al., 2026), fine-grained perception (Grover et al., 2025), abstract visual reasoning (Zhang et al., 2024; Yan et al., 2025; Lin et al., 2026; Sinha et al., 2026), mental imagery (Zeller et al., 2026), and atomic visual abilities (Mai et al., 2026). Concurrently, recent efforts such as Kanade & Ganu (2026); Ghosh et al. (2025); Grover et al. (2025); Ge et al. (2026); Sinha et al. (2026) have begun leveraging programmatic rendering and parametric control to generate diverse data automatically, effectively scaling visual diversity while substantially reducing dataset construction costs.

7 CONCLUSION

In this work, we presented BUTTONSBENCH, an extensible procedural platform, and BUTTONSBENCH-1, a diagnostic benchmark that parameterizes visual puzzle complexity via discrete, controllable difficulty knobs. By pairing base queries with decoupled co-origin behavioral checkpoints, our framework moves beyond outcome-only scoring to expose pervasive shortcut heuristics and reveal that isolated parameter endurance abruptly collapses under combinatorial scaling. These diagnostic findings indicate that future model training must prioritize intermediate state reasoning and the maintenance of continuous deduction chains, rather than solely rewarding final-answer prediction. More broadly, by drastically reducing manual annotation overhead through programmatic generation, our framework establishes a versatile operational workbench for the community—serving both as an extensible, living testbed for dynamic evaluation and as an automated data engine for process-supervised reasoning training.

LIMITATIONS

Several limitations of this work chart valuable directions for future inquiry. First, regarding task breadth, while BUTTONSBENCH-1 spans nine core puzzle families covering six reasoning domains, it remains confined to structured, stylized visual mechanics and does not yet capture messy, open-world physical dynamics or embodied interactions. Second, the benchmark is currently restricted to Chinese instructions, leaving model resilience under multilingual or cross-cultural prompt variations unexplored. Third, our empirical audit centered on four leading frontier MLLMs to establish proof of concept; broadening this behavioral audit across diverse parameter scales, open-weight architectures, and reasoning-specialized models constitutes an immediate avenue for future work.

AI USE STATEMENT

Generative AI was used only for writing polishing and figure and table formatting. We take responsibility for the final content of this work.

REFERENCES

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Yiping Bao, et al. Babyvision: Visual reasoning beyond language. *arXiv preprint arXiv:2601.06521*, 2026.

Google DeepMind. deepmind.google. https://deepmind.google/models/model-cards/gemini-3-7-flash/, 2026.

Yunfei Ge, Anbang Liu, Qineng Wang, Johnalbert Garnica, Jianwen Lyu, Zihan Wang, Reuben Tan, Jianfeng Gao, Ruohan Zhang, Yining Hong, Jiajun Wu, and Manling Li. Mindtopo: Can foundation models reason in topological space?, 2026. URL https://arxiv.org/abs/2609.11900.

Samrajnee Ghosh, Naman Agarwal, Hemanshu Garg, Chinmay Mittal, Parag Singla, et al. The percept-v challenge: Can multimodal llms crack simple perception problems? *arXiv preprint arXiv:2508.21143*, 2025.

Rynaa Grover, Jayant Sravan Tamarapalli, Sahiti Yerramilli, and Nilay Pande. Huemanity: Probing fine-grained visual perception in mllms, 2025. URL https://arxiv.org/abs/2506.03194.

Xixiang He, Baiqi Wu, Xingming Li, Ao Cheng, Qiyao Sun, Xuanyu Ji, and Qingyong Hu. Stembind: When mllms get lost between rules and instances in abstract visual reasoning, 2026. URL https://arxiv.org/abs/2606.00148.

Aditya Sanjiv Kanade and Tanuja Ganu. Do you see me: A multidimensional benchmark for evaluating visual perception in multimodal llms. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7285–7326, 2026.

Tina Khezresmaeilzadeh, Jike Zhong, and Konstantinos Psounis. Vriq: Benchmarking and analyzing visual-reasoning iq of vlms, 2026. URL https://arxiv.org/abs/2602.05382.

Xunhao Lai, Weiqi Xu, Yufeng Yang, Qiaorui Chen, Yang Xu, Lunbin Zeng, Xiaolong Li, Haohai Sun, Haichao Zhu, Vito Zhang, Jinkai Hu, Jiayao Li, Rui Gao, Zekun Li, Songquan Zhu, Jingkai Zhou, and Pengyu Zhao. Minimax sparse attention, 2026. URL https://arxiv.org/abs/2606.13392.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning, july 2023. *URL http://arxiv. org/abs/2307.13702*, 2023.

Zichao Lin, Yifeng Xie, Bowen Qu, Haiming Wang, Jia Li, Haoning Wu, Yuhao Dong, Zuhao Yang, Jinguo Zhu, Haoyu Lu, Zijia Zhao, Tongtian Yue, Zhangyang Qi, Junwei Yang, Mengfan Dong, Peizhou Cao, Chenzhuang Du, Zaida Zhou, Haotian Yao, Hao Yang, Hongcheng Gao, Lin Sui, Weihong Li, Xinxing Zu, Jia Chen, Yao Wang, Xiaoxue Wu, Yalin Wang, Y. Charles, Yiping Bao, Yangyang Liu, Zhiqi Huang, and Xinyu Zhou. Perceptionbench: Evaluating atomic visual perception in multimodal large language models, 2026. URL https://arxiv.org/abs/2607.24957.

Zheda Mai, Arpita Chowdhury, Zihe Wang, Sooyoung Jeon, Lemeng Wang, Jiacheng Hou, Jihyung Kil, and Wei-Lun Chao. Ava-bench: Atomic visual ability benchmark for vision foundation models, 2026. URL https://arxiv.org/abs/2506.09082.

Mayug Maniparambil, Nils Hoehing, Janak Kapuriya, Arjun Karuvally, Ellen Rushe, Anthony Ventresque, Noel O'Connor, and Fergal Reid. Topobench: Benchmarking llms on hard topological reasoning, 2026. URL https://arxiv.org/abs/2603.12133.

Amirreza Mohseni, Mona Mohammadi, Morteza Saghafian, and Naser Talebizadeh Sardari. Curvebench: A benchmark for exact topological reasoning over nested jordan curves, 2026. URL https://arxiv.org/abs/2605.14068.

OpenAI. Introducing gpt-5.5. https://openai.com/index/gpt-5-5-system-card/, 2026. [Accessed 26-09-2026].

Patrick Rim, Tom Long, Ekta Prashnani, Ruth Rosenholtz, Ben Boudaoud, Peter Xenopoulos, Alex Wong, Joohwan Kim, and Jae-Hyun Jung. Spatial-iq: Deconstructing spatial intelligence via hierarchical capability tests, 2026. URL https://arxiv.org/abs/2607.22864.

Rohit Sinha, Aditya Kanade, Sai Srinivas Kancheti, Vineeth N Balasubramanian, and Tanuja Ganu. Mind's eye: A benchmark of visual abstraction, transformation and composition for multimodal llms, 2026. URL https://arxiv.org/abs/2604.16054.

Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu, Houqiang Li, Xiaohua Wang, Xizhou Zhu, Wenhai Wang, Jifeng Dai, and Jinguo Zhu. Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models, 2025. URL https://arxiv.org/abs/2504.15279.

Hao Yan, Xingchen Liu, Hao Wang, Zhenbiao Cao, Handong Zheng, Liang Yin, Xinxing Su, Zihao Chen, Jihao Wu, Minghui Liao, Chao Weng, Wei Chen, Yuliang Liu, and Xiang Bai. Visuriddles: Fine-grained perception is a primary bottleneck for multimodal large language models in abstract visual reasoning, 2025. URL https://arxiv.org/abs/2506.02537.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Jana Zeller, Thaddäus Wiedemer, Fanfei Li, Thomas Klein, Prasanna Mayilvahanan, Matthias Bethge, Felix Wichmann, Ryan Cotterell, and Wieland Brendel. Mentisoculi: Revealing the limits of reasoning with mental imagery, 2026. URL https://arxiv.org/abs/2602.02465.

Yizhe Zhang, He Bai, Ruixiang Zhang, Jiatao Gu, Shuangfei Zhai, Josh Susskind, and Navdeep Jaitly. How far are we from intelligent visual deductive reasoning?, 2024. URL https://arxiv.org/abs/2403.04732.

Yiyang Zhou, Haoqin Tu, Zijun Wang, Zeyu Wang, Niklas Muennighoff, Fan Nie, Chaorui Deng, Shen Yan, Haoqi Fan, Yejin Choi, et al. When visualizing is the first step to reasoning: Mira, a benchmark for visual chain-of-thought. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26154–26164, 2026.

A APPENDIX

A.1 EVALUTATION DETAILS

Our accuracy metrics of Acc_{base} and $\text{Acc}_{\text{joint}}$ are defined as follows:

- **Instance-level (Single Question):** For a specific question instance (t, i), its baseline and verified masteries are deterministic binary indicators:

$$\text{Acc}_{\text{base}}(t,i) = T_{t,i}^{\text{base}} \in \{0,1\}, \quad \text{Acc}_{\text{joint}}(t,i) = \mathbb{I}\left(T_{t,i}^{\text{base}} = 1 \ \wedge \ T_{t,i}^{\text{probe}} = 1\right) \in \{0,1\}. \tag{5}$$

  An instance is jointly mastered if and only if the model reliably solves both the primary question and every paired co-origin probe under majority voting.

- **Task-level (Within a Specific Task):** For a specific task t, its task-level accuracy represents the empirical success rate averaged over all N_t material instances belonging to that task:

$$\text{Acc}_{\text{base}}^{(t)} = \frac{1}{N_t}\sum_{i=1}^{N_t}\text{Acc}_{\text{base}}(t,i), \quad \text{Acc}_{\text{joint}}^{(t)} = \frac{1}{N_t}\sum_{i=1}^{N_t}\text{Acc}_{\text{joint}}(t,i). \tag{6}$$

- **Benchmark-level (Overall Benchmark Average):** Across the entire benchmark, overall competence is reported as the global micro-average across all N instances (or equivalently, instance-weighted macro-average across all $|\mathcal{T}|$ tasks):

$$\overline{\text{Acc}}_{\text{base}} = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{i=1}^{N_t}\text{Acc}_{\text{base}}(t,i), \quad \overline{\text{Acc}}_{\text{joint}} = \frac{1}{N}\sum_{t\in\mathcal{T}}\sum_{i=1}^{N_t}\text{Acc}_{\text{joint}}(t,i). \tag{7}$$

A.2 FURTHER EXPERIMENTS RESULTS

A.2.1 DOMAIN-SPECIFIC FAILURE ANALYSIS

By categorizing the nine tasks into core cognitive domains, we observe clear structural boundaries in contemporary multimodal deduction:

Static 2D Perceptual Search is Probe-Resilient. In pure 2D visual search tasks, models exhibit genuine perceptual grounding. On `find_different`, `gpt-5.5` achieves a flawless 100.0% base and joint accuracy (FR $= 0.0\%$), while `qwen-3.8-max` maintains 88.0% joint accuracy (FR $= 2.2\%$). Similarly, on `count_multiple_patterns`, top models exhibit low falsification (2.5% for Qwen, 6.0% for GPT). This indicates that frontier models possess reliable static 2D perceptual grounding, effectively performing fine-grained pattern discrimination and visual search without relying on spurious heuristics. The sole exception is `pattern_completion`, where even frontier models suffer extreme probe destruction (FR from 54.3% up to 95.5%).

Dynamical Tracking Suffers Intermediate Propagation Collapse. In physical and sequential tracking domains (`gear_rotation`, `rolling_dice`), the divergence between endpoint accuracy and verified deduction reaches its extreme. In `gear_rotation`, while `gpt-5.5` and `gemini-3.7-flash` register high baseline scores (94.4% and 66.7%), their probe failure rates surge to 35.3% and 75.0%, while `MiniMax-M3` collapses entirely ($50.0\% \rightarrow 0.0\%$, FR $= 100.0\%$). Crucially, probe-level breakdown reveals that models fail specifically on state-enumeration probes, which query intermediate gear orientations and connection types. Despite modest endpoint accuracy, models fail to maintain consistent rotational states across intermediate gears, demonstrating that deductions collapse along the transmission chain rather than at the final prediction.

3D Spatial Geometry Hits Structural Hard Walls. In `cubes_count` and `trailer_cubes_count`, severe floor effects dominate. Under multi-layer isometric occlusion, `MiniMax-M3` scores 0.0% base on `trailer_cubes_count`, and `gemini-3.7-flash` experiences an 83.3% falsification drop. Current MLLMs exhibit catastrophic blind spots when required to infer occluded spatial volume beneath visible surface layers.



Figure 4: **Endpoint competence vs. probe-verified competence profiles.** Radar plots delineate task-level performance across the nine puzzles for each evaluated architecture. Solid outer contours denote accuracy of base questions only (Acc_{base}), while dashed inner contours denote verified joint accuracy ($\text{Acc}_{\text{joint}}$). The shaded divergence highlights architectural fragility: frontier models maintain robust mastery over top-hemisphere perceptual tasks (*e.g.*, Count multiple patterns), but uniformly collapse toward the center origin on bottom-hemisphere 3D spatial and sequential dynamics.

A.2.2 Competence Radar plots for all models

Figure 4shows the the endpoint prediction competence and prob-verified competence of 4 models across 9 tasks.

A.3 ButtonsBench-1 Details

A.4 Data Compositions of ButtonsBench-1

The data compositions of ButtonsBench-1 is shown in Table 4

A.4.1 Examples of 9 tasks in ButtonsBench-1 Benchmark

Examples of 9 tasks in ButtonsBench-1 benchmark are shown in Figure 5.

A.4.2 Button configurations

Buttons compositions and their levels configurations are shown in Figure 5.

Table 4: Data Compositions of BUTTONSBENCH-1.

Task Name	Basic Ques.	Probe Types	Probe Ques.	Total
pattern_completion	144	1	144	288
find_different	50	2	100	150
count_multiple_patterns	160	1	160	320
trailer_cubes_count	64	1	64	128
cubes_count	126	2	252	378
gear_rotation	18	3	72	90
connect_lines	35	1	35	70
mirroring_patterns	45	1	45	90
rolling_dice	49	1	49	98
Total	**691**	**14**	**903**	**1,594**

Probe types indicate the number of distinct probe templates evaluated per task; each probe template evaluates exactly one probe question across button-level combinations. Across the benchmark, 13 probe types instantiate 903 co-origin probes to audit the 691 basic questions.

Table 5: Button configurations across all nine tasks in BUTTONSBENCH-1.

Task Name	Configured Button (`Factor`)	Levels	Value Range	Comb.
find_different	Grid Size (`grid_size`)	L0–L4	10, 15, 20, 25, 30	50
	Target Count (`num_target`)	L0–L9	1, 2, …, 10	
count_multiple_patterns	Rotation (`rotation_enabled`)	L0, L1	False, True	160
	Occlusion (`overlap_level`)	L0, L1	0, 1	
	Quantity (`total_elements`)	L0–L4	20, 25, …, 40	
	Categories (`num_categories`)	L0–L7	1, 2, …, 8	
pattern_completion	Rotation (`rotation_enabled`)	L0, L1	False, True	144
	Grid Size (`side_length`)	L0–L5	10, 15, 20, 25, 30, 40	
	Quantity (`num_components`)	L0–L3	1, 3, 5, 7	
	Hole Size (`hole_size_range`)	L0–L2	[7, 8], [9, 10], [11, 12]	
cubes_count	Occlusion (`occlusion_level`)	L0–L2	0, 1, 2	126
	Grid Size (`grid_size`)	L0–L5	2, 3, 4, 5, 6, 7	
	Quantity (`max_height`)	L0–L6	1, 2, …, 7	
trailer_cubes_count	Grid Size (`grid_size`)	L0–L7	5, 10, 15, 20, 25, 30, 40, 50	64
	Quantity (`max_height`)	L0–L7	5, 10, 15, 20, 25, 30, 40, 50	
mirroring_patterns	Grid Size (`grid_size`)	L0–L8	10, 15, …, 50	45
	Colored Ratio (`colored_cells_ratio`)	L0–L4	0.1, 0.3, 0.5, 0.7, 0.9	
gear_rotation	Gear Count (`num_gears`)	L0–L5	10, 15, 20, 30, 40, 50	18
	Belt Conn. (`belt_connection_ratio`)	L0–L2	0.1, 0.3, 0.5	
connect_lines	Occlusion (`cross_density`)	L0–L6	0, 1, …, 6	35
	Quantity (`num_pairs`)	L0–L4	2, 3, …, 6	
rolling_dice	Turns (`num_turns`)	L0–L6	0, 1, …, 6	49
	Reasoning Steps (`num_steps`)	L0–L9	1, 2, …, 10	

Comb. = number of unique button-level combinations (each basic exam contains exactly one basic question per unique combination). Grid sizes denote square side lengths. For *rolling_dice*, 21 of the 70 nominal combinations violating $\texttt{num_steps} \geq \texttt{num_turns} + 1$ are excluded.

A.4.3 BUTTONSBENCH PLATFORM AND BENCHMARK CONSTRUCTION PIPELINE

A.4.4 STAGE 1: MATERIAL GENERATION

The core premise of fine-grained diagnosis is isolating the exact structural factors governing visual complexity. For each task, we identify the explicit visual conditions and structural attributes that impact perception or reasoning (e.g., grid dimensions, sequence lengths, or distractor counts), formalized as quantitative **factors**.

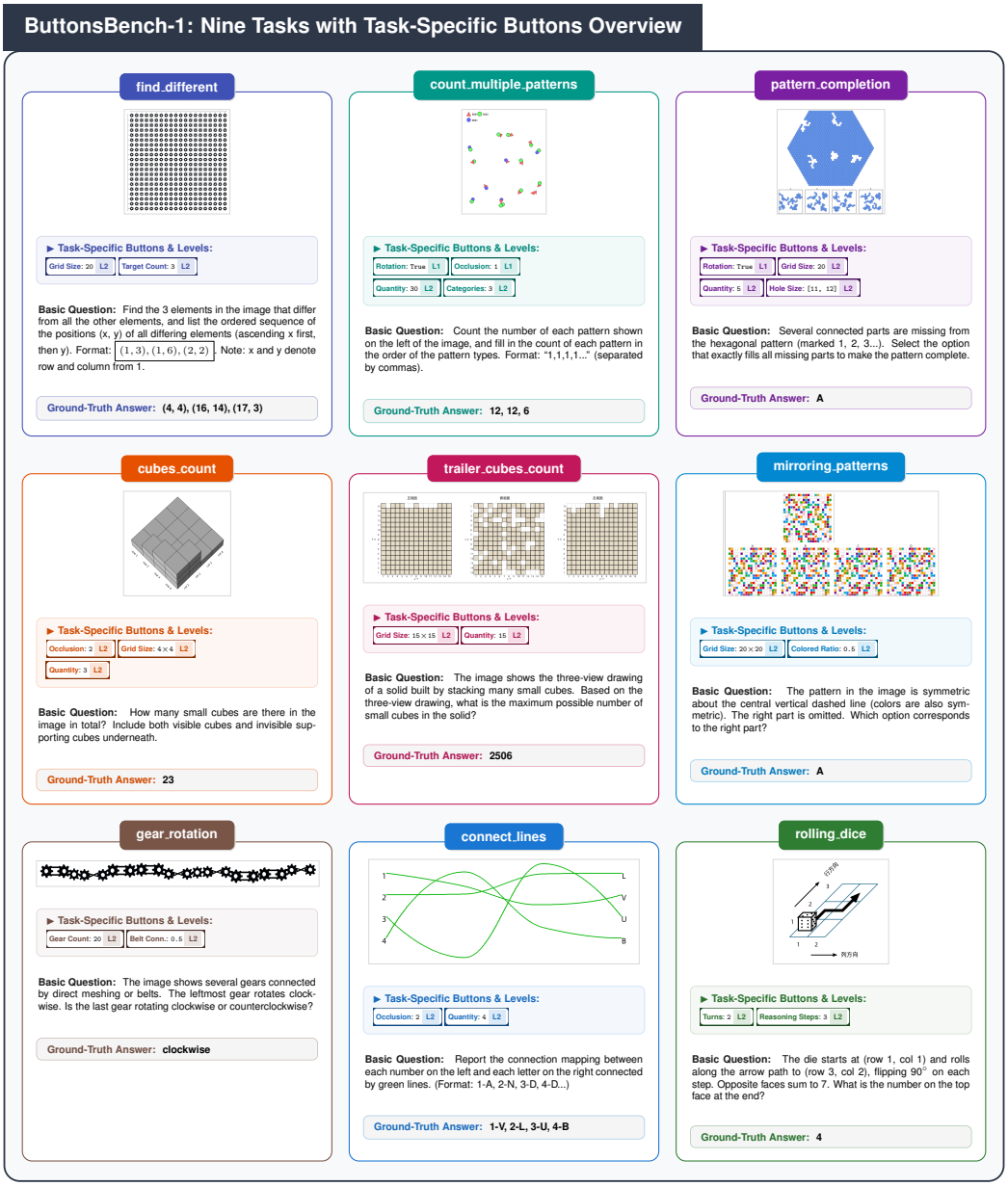


Figure 5: Overview of the nine tasks in BUTTONSBENCH-1. Each panel demonstrates a material diagram, its corresponding basic question, and the configured button levels.

The Material Generation stage synthesizes pure visual material instances through programmatic factor sweeps. For any target task, the platform takes a generation script alongside a task-specific JSON configuration template that defines: a) the target factors to sweep alongside their concrete values, b) the factor-level combinations to cover, and c) the number of distinct stochastic instantiations generated per factor configuration.

For every valid factor configuration, this stage generates a self-contained *Material Instance Package*. Crucially, this package contains not only the rendered images (including the primary diagram and, where applicable, sub-diagrams or option images), but also an accompanying **Material Metadata** file. This file records two tiers of information: (1) instance-level properties, such as instance IDs, relative image paths, and task identifiers; and (2) fine-grained structural and state generation logs, such as step-by-step state transitions, the exact spatial coordinates, colors, and counts of key

elements. Retaining these underlying state records is the bedrock that allows subsequent stages to instantiate rigorous behavioral checkpoints without manual annotation.

Following generation, benchmark creators can optionally review the material instances to filter out visually flawed samples, ensuring a high-quality material pool for downstream question construction.

A.4.5 STAGE 2: QUESTION CONSTRUCTION

The Question Construction stage ingests the validated Material Instance Packages and programmatically instantiates concrete evaluation items. Unlike conventional authoring pipelines that construct a single question per image, BUTTONSBENCH exploits the rich material metadata of each material instance to instantiate both the **basic question** and a set of **co-origin behavioral probes**.

To enable controlled evaluation, the platform establishes a structured button-and-level mapping by formalizing the decoupled, independent factors within each task as dedicated **buttons**, with their discrete values mapped to difficulty **levels** (*e.g.*, Level 0 to Level 5). In this stage, configuration files specify these factor-to-level assignments for the target task. This design allows researchers to simulate "tuning a knob"—systematically varying one button while holding all other factors constant to isolate single-factor failure modes.

Building upon these parameter definitions, deterministic ground-truth derivation ensures absolute label fidelity. Rather than relying on error-prone human labeling or ambiguous natural language descriptions, question prompts and answers are instantiated programmatically. Benchmark designers specify a question prompt template alongside a ground-truth derivation function that directly queries the underlying material metadata to compute the provably correct answer. For multiple-choice tasks, the configuration specifies the candidate options and rendering layouts, ensuring deterministic consistency across all generated items.

Crucially, retaining complete state logs unlocks programmatic co-origin probe derivation, extending question construction well beyond basic target queries. The platform automatically derives task-specific *co-origin probes* from the very same material instance:a) **Intermediate State Probes**, which query intermediate deductions along the visual reasoning path (e.g., the rotation direction of an intermediate gear in a transmission chain); b) **Local Condition Perturbations**, which perturb localized textual conditions without altering the visual asset (e.g., inverting an initial input state) to test model robustness; and c) **Format Variants**, which modify question structures or answer formats to evaluate inference consistency against variations in prompt formatting. Concrete examples of basic questions paired with their co-origin probes across all tasks are provided in Appendix A.5. The output of this stage is an integrated *Question Item Package*, comprising the final composite image (containing prompt context, diagrams, and options), verified answer labels, paired probe items, and comprehensive question instance metadata.

A.4.6 STAGE 3: EXAM ASSEMBLY

In this final stage, the platform curates ready-to-evaluate benchmarks, or "exams," from the pool of instantiated question items based on designated evaluation objectives. Benchmark creators can flexibly assemble homogeneous test suites drawn from a single task with aligned difficulty levels, or heterogeneous benchmarks that span multiple tasks to support comprehensive visual reasoning evaluation. The exported benchmark package comprises three components: a) the rendered question images, b) a consolidated questions data file aggregating all questions' metadata, and c) a benchmark metadata file documenting dataset-level summary statistics, including task compositions and question counts.

A.4.7 PLATFORM EXTENSIBILITY AND BENCHMARK SUSTAINABILITY

The modular design of BUTTONSBENCH directly resolves two chronic failure modes of contemporary visual reasoning evaluations:

Modular Task Extensibility and Flexible Benchmarking. Beyond the default task suite, the modularity of BUTTONSBENCH provides native extensibility for the community. First, researchers can readily incorporate new tasks by implementing a custom material generator and a corresponding

configuration file, or author novel question prompts and co-origin probe variants on top of existing material instances. Second, the decoupled assembly stage allows benchmark creators to flexibly tailor evaluation benchmarks to specific research goals—whether isolating a single factor within a homogeneous task or constructing a broad, heterogeneous test suite across diverse tasks.

Automated Resistance to Benchmark Saturation. As multimodal architectures advance, traditional static benchmarks rapidly suffer ceiling effects and are discarded. In BUTTONSBENCH-1, task difficulty is objectively grounded in factor parameters. When contemporary models saturate a given difficulty level, creators need not design new tasks from scratch; they simply extend the button configurations in the generation template (e.g., increasing grid dimensions from 5×5 to 8×8, or lengthening topological dependency chains), programmatically scaling difficulty upward while preserving task semantics.

Dynamic Immunity to Data Contamination. Because all materials, questions, and co-origin probes are generated programmatically from parameterized algorithms, the entire visual testbed can be refreshed on demand. By varying the pseudo-random generation seeds, the platform instantiates brand-new visual scenes, distractor layouts, and topological configurations that share identical distribution properties and factor levels with the original benchmark but possess completely novel pixel realizations. This guarantees minimal test-set leakage and ensures reliable zero-shot evaluation for future MLLM iterations.

A.5 EXAMPLES OF QUESTION INSTANCES

The detailed question prompts of the base question and paired co-origin questions of each task in BUTTONSBENCH-1 are shown in Figures from Figure 6 to Figure 14 .

A.5.1 BUTTONSBENCH PLATFORM UI INTERFACE

Figure 15 to Figure 17 illustrate the user interfaces of the BUTTONSBENCH platform across its three data construction stages.

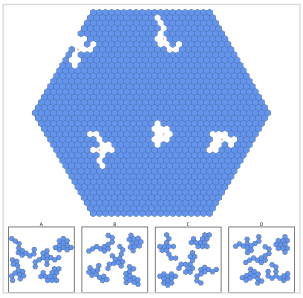
A.6 EXPERIMENT DETAILS

A.6.1 LLM-AS-JUDGE PROMPT

Our prompt for LLM-as-judge is demonstrated in A.6.1.

Task: Pattern Completion (Difficulty Level 2)

Configurations (L2): • **Rotation**: `on (L1)` • **Grid Size**: `20 (L2)` • **Components**: `5 (L2)` • **Hole Size**: `[11, 12] (L2)`



(a) Base Task Diagram

(b) Co-Origin Probe (Variant sub1.0)

Base Question: Several connected parts are missing from the large hexagonal pattern in the image, and each missing part is marked with a red numeric label (1, 2, 3, etc.). Analyze each option one by one and determine whether it can exactly fill in all the missing parts to make the pattern a complete large hexagon. Select the option that exactly fills in all the blanks and completes the large hexagon, and put its letter label inside `\boxed{}`.

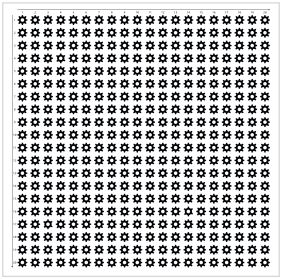
Ground Truth: `A`

Co-Origin Probe 1 (State Enumeration Probing, sub1.0):
Several connected parts are missing from the large hexagonal pattern in the image, and each missing part is marked with a red numeric label (1, 2, 3, etc.). Analyze each option one by one and determine whether it can exactly fill in all the missing parts to make the pattern a complete large hexagon, and report the analysis result of each option as a pair `(option letter, conflicting_hole_id)`: the option letter is the label of that option (A, B, C, etc.), and `conflicting_hole_id` is the index of the missing part that the option's pattern fails to fill; if the option's pattern can exactly fill in all the missing parts, set `conflicting_hole_id` to `none` (if the pattern fails to fill multiple missing parts, join their indices with a plus sign, e.g., `1+3`). List all pairs separated by English commas in ascending order of the option letters, and put the final answer inside `\boxed{}`, e.g., `\boxed{(A, 1), (B, none), (C, 3), (D, none)}`.

Ground Truth: `(A, none), (B, 5), (C, 1), (D, 5)`

Figure 6: Base question and co-origin probe specification for the **Pattern Completion** task under Level 2 configurations.

Task: Find Different (Difficulty Level 2)



Configurations (L2): • **Grid Size**: `20 (L2)` • **Target Element Count**: `3 (L2)`

Base Question: Find the 3 elements in the image that differ from all the other elements, and list the ordered sequence of the positions (x, y) of all the differing elements, separated by commas. Order the coordinates by 'ascending x first, then ascending y'. The final answer should look like: `\boxed{(1, 3), (1, 6), (2, 2)}`. Note: x and y denote the row and column of a differing element, counted from 1.

Ground Truth: `(4, 4), (16, 14), (17, 3)`

Co-Origin Probe 1 (Row-wise State Enumeration, var1.0):
There are 3 minority outlier elements in the image whose patterns differ from the majority. Count how many outlier elements there are in each row of the grid, and write the per-row counts as a sequence of pairs ordered by row index. In each pair (row, count), row is the row index and count is the number of outlier elements in that row; if a row contains no outlier elements, its count is 0 and that row must still be listed. Join the pairs with English commas in ascending row order, and put the complete pair sequence inside `\boxed{}`, e.g., `\boxed{(1, 3), (2, 0), (3, 1)}`.

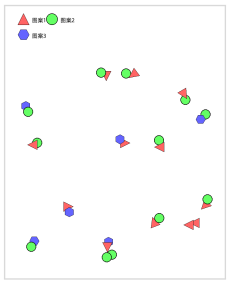
Ground Truth: `(1, 0), (2, 0), (3, 0), (4, 1), (5, 0), (6, 0), (7, 0), (8, 0), (9, 0), (10, 0), (11, 0), (12, 0), (13, 0), (14, 0), (15, 0), (16, 1), (17, 1), (18, 0), (19, 0), (20, 0)`

Co-Origin Probe 2 (Column-wise State Enumeration, var2.0):
There are 3 minority outlier elements in the image whose patterns differ from the majority. Count how many outlier elements there are in each column of the grid, and write the per-column counts as a sequence of pairs ordered by column index. In each pair (column, count), column is the column index and count is the number of outlier elements in that column; if a column contains no outlier elements, its count is 0 and that column must still be listed. Join the pairs with English commas in ascending column order, and put the complete pair sequence inside `\boxed{}`.

Ground Truth: `(1, 0), (2, 0), (3, 1), (4, 1), (5, 0), (6, 0), (7, 0), (8, 0), (9, 0), (10, 0), (11, 0), (12, 0), (13, 0), (14, 1), (15, 0), (16, 0), (17, 0), (18, 0), (19, 0), (20, 0)`

Figure 7: Base question and co-origin probes specification for the **Find Different** task under Level 2 configurations.

Task: Count Multiple Patterns (Difficulty Level 2)



Configurations (L2): • **Rotation**: `on (L1)` • **Overlap Level**: `1 (L1)` • **Total Elements**: `30 (L2)` • **Categories**: `3 (L2)`

Base Question: Count the number of each pattern shown on the left of the image, and fill in the count of each pattern in the order of the pattern types. The final answer format is like "1,1,1,1..." (the counts of the patterns separated by commas).

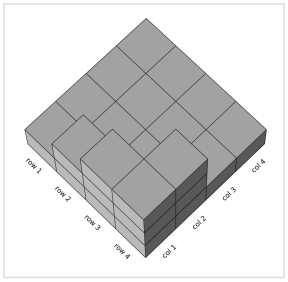
Ground Truth: `12, 12, 6`

Co-Origin Probe 1 (Pattern Count Verification & Correction, var1.0):
The legend on the left of the image lists the patterns by index (Pattern 1, Pattern 2, ...). Count the number of each pattern on the left of the image and check the given sequence of element-count pairs for each pattern, correcting any incorrect counts. In each pair `(pattern_id, quantity)`, `pattern_id` is the legend index of the pattern (Pattern 1 corresponds to 1, Pattern 2 to 2, and so on) and `quantity` is the element count of that pattern. The possibly incorrect sequence of pattern-count pairs is: `(1, 14),(2, 11),(3, 4)`. Join the pairs with English commas in ascending pattern order, and put the corrected complete sequence inside `\boxed{}`, e.g., `\boxed{(1, 5),(2, 6),(3, 4)}`.

Ground Truth: `(1, 12),(2, 12),(3, 6)`

Figure 8: Base question and co-origin probe specification for the **Count Multiple Patterns** task under Level 2 configurations.

Task: Cubes Count (Difficulty Level 2)



Configurations (L2): • **Occlusion Level**: `2 (L2)` • **Grid Size**: `4x4 (L2)` • **Stack Height**: `3 (L2)`

Base Question: How many small cubes are there in the image in total?

Ground Truth: `23`

Co-Origin Probe 1 (Column-wise Height Enumeration, sub_v3.0):
The image shows a cube stack built from small cubes, with row and column indices labeled on the edges; the coordinate (x, y) denotes row x and column y, with both x and y counted from 1. Write the column height at each coordinate. Express the height at each coordinate as a triple (x, y, height): height is the height (number of layers) of the cube column at that position; if no cube exists at a coordinate, its height is 0. Join the triples with English commas ordered by 'ascending x first, then ascending y', and put the final answer inside `\boxed{}`, e.g., `\boxed{(1, 1, 1), (1, 2, 1), (1, 3, 1)}`.

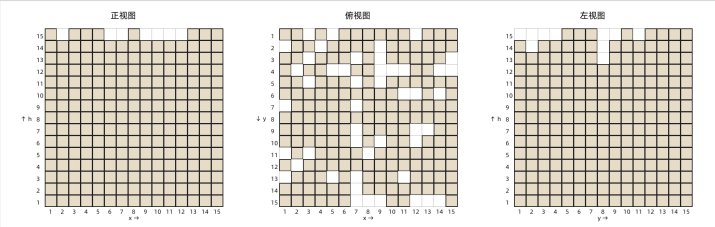
Ground Truth: `(1, 1, 1), (1, 2, 1), (1, 3, 1), (1, 4, 1), (2, 1, 2), (2, 2, 1), (2, 3, 1), (2, 4, 1), (3, 1, 3), (3, 2, 1), (3, 3, 1), (3, 4, 1), (4, 1, 3), (4, 2, 3), (4, 3, 1), (4, 4, 1)`

Co-Origin Probe 2 (Spatial Coordinates Enumeration, sub_v4.0):
The image shows a cube stack built from small cubes, with row and column indices labeled on the edges; the spatial coordinate (x, y, z) denotes row x, column y, and the z-th layer counted from bottom to top, with x, y, z all at least 1. List the spatial coordinates of every small cube in the stack (including occluded ones); if no cube exists at a spatial coordinate, omit that position. Express each cube's spatial coordinate as a triple (x, y, z). Join the triples with English commas ordered by 'descending z first, then ascending x, then ascending y', and put the final answer inside `\boxed{}`, e.g., `\boxed{(1, 1, 2), (1, 1, 1), (1, 2, 1), (1, 3, 1)}`.

Ground Truth: `(3, 1, 3), (4, 1, 3), (4, 2, 3), (2, 1, 2), (3, 1, 2), (4, 1, 2), (4, 2, 2), (1, 1, 1), (1, 2, 1), (1, 3, 1), (1, 4, 1), (2, 1, 1), (2, 2, 1), (2, 3, 1), (2, 4, 1), (3, 1, 1), (3, 2, 1), (3, 3, 1), (3, 4, 1), (4, 1, 1), (4, 2, 1), (4, 3, 1), (4, 4, 1)`

Figure 9: Base question and co-origin probes specification for the **Cubes Count** task under Level 2 configurations.

Task: Trailer Cubes Count (Difficulty Level 2)



Configurations (L2): • **Grid Size**: `15x15 (L2)` • **Stack Height**: `15 (L2)`

Base Question: The image shows the three-view drawing of a solid built by stacking many small cubes. Based on the three-view drawing, what is the maximum possible number of small cubes in the solid?

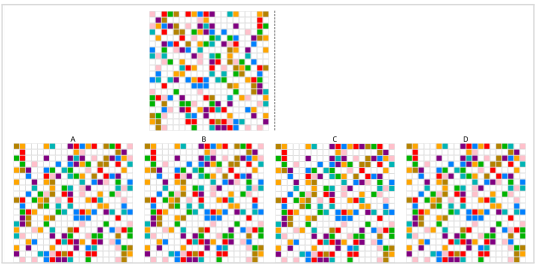
Ground Truth: `2506`

Co-Origin Probe 1 (Max Column Height Mapping, sub1.0):
The image shows the three-view drawing of a solid built by stacking many small cubes. Based on the three-view drawing, determine the maximum possible number of small cubes at each (x, y) coordinate position in the top view (i.e., the maximum possible column height at each (x, y) coordinate). Write the column height at each top-view coordinate as a triple (x, y, height), where x and y are counted from 1 and are consistent with the coordinate labels in the drawing; if no cube can exist at a coordinate, its height is 0. Order all triples by ascending x first, then ascending y, join them with English commas, and put the complete triple sequence inside `\boxed{}`, e.g., `\boxed{(1, 1, 1), (1, 2, 1), (1, 3, 0), ..., (2, 1, 1), (2, 2, 1), ...}`.

Ground Truth: `(1, 1, 14), (1, 2, 0), (1, 3, 14), (1, 4, 14), (1, 5, 15), (1, 6, 15), (1, 7, 0), (1, 8, 12), (1, 9, 14), (1, 10, 15), (1, 11, 0), ..., (15, 14, 15), (15, 15, 15)`

Figure 10: Base question and co-origin probe specification for the **Trailer Cubes Count** task under Level 2 configurations.

Task: Mirroring Patterns (Difficulty Level 2)



Configurations (L2): • **Grid Size**: `20x20 (L2)` • **Coloring Ratio**: `0.5 (L2)`

Base Question: The pattern in the image is symmetric about the central vertical dashed line (the colors are also symmetric). The right part of the pattern is now omitted. Which of the options should be the right part?

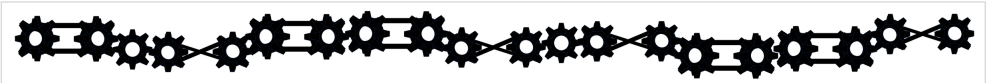
Ground Truth: `A`

Co-Origin Probe 1 (Violating Color & Correction Probing, sub1.0):
The pattern in the image is symmetric about the central vertical dashed line (the colors are also symmetric), and the right part of the pattern is now omitted. Exactly one of the option images is a right part fully symmetric to the part left of the dashed line; each of the remaining options contains exactly one cell that violates the symmetry with the part left of the dashed line. For each option, report the color of the cell that violates the symmetry, and the color that cell should have to preserve the symmetry. Express each option as a triple `(option_label, incorrect_grid_color, correct_grid_color)`: `option_label` is the letter of the corresponding option image (A, B, C, D, etc.), `incorrect_grid_color` is the color of the cell that violates the symmetry, and `correct_grid_color` is the color that cell should have to preserve the symmetry (cell colors are expressed with these English words: red, blue, green, orange, purple, pink, cyan, brown, white); if an option is fully symmetric to the part left of the dashed line (no violating cell), fill both `incorrect_grid_color` and `correct_grid_color` with `none`. Join the triples with English commas in ascending option-letter order, and put the final answer inside `\boxed{}`, e.g., `\boxed{(A, orange, blue),(B, none, none),(C, red, white),(D, white, green)}`.

Ground Truth: `(A, cyan, red),(B, none, none),(C, purple, white),(D, red, pink)`

Figure 11: Base question and co-origin probe specification for the **Mirroring Patterns** task under Level 2 configurations.

Task: Gear Rotation (Difficulty Level 2)



Configurations (L2): • **Gear Count**: `20 (L2)` • **Belt Connection Ratio**: `0.5 (L2)`

Base Question: The image shows several gears connected by direct meshing or by belts. The leftmost gear rotates clockwise. Is the last gear rotating clockwise or counterclockwise? Put the final answer inside `\boxed{}`, e.g., `\boxed{clockwise}`.

Ground Truth: `clockwise`

Co-Origin Probe 1 (Counterfactual Initial Direction Probing, counter_clockwise):
The image shows several gears connected by direct meshing or by belts. The leftmost gear rotates counterclockwise. Is the last gear rotating clockwise or counterclockwise? Put the final answer inside `\boxed{}`, e.g., `\boxed{clockwise}`.

Ground Truth: `counterclockwise`

Co-Origin Probe 2 (Pairwise Transmission Mechanism Probing, sub_v2.0):
The image shows a chain of gears where adjacent gears are connected by direct meshing or by belts. Starting from the leftmost first gear, number the gears 1, 2, 3, ... along the connection order, and list the connection relation of each pair of adjacent gears in connection order. Express each connection as a triple `(from_gear, to_gear, connection_type)`: `from_gear` and `to_gear` are the indices of the two adjacent gears (`from_gear < to_gear`), and `connection_type` is one of `mesh` (direct meshing), `parallel` (parallel belt connection), or `cross` (crossed belt connection). Join the triples with English commas in connection order and put the final answer inside `\boxed{}`, e.g., `\boxed{(1, 2, mesh), (2, 3, cross), (3, 4, parallel)}`.

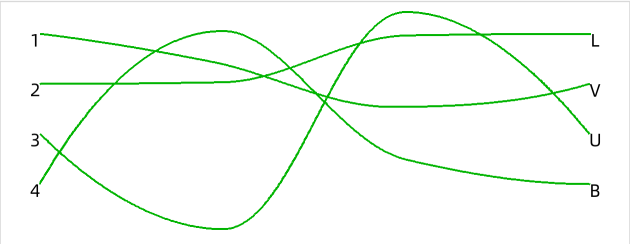
Ground Truth: `(1, 2, parallel), (2, 3, mesh), (3, 4, mesh), (4, 5, cross), (5, 6, mesh), (6, 7, parallel), (7, 8, mesh), (8, 9, parallel), (9, 10, mesh), (10, 11, cross), (11, 12, mesh), (12, 13, mesh), (13, 14, cross), (14, 15, mesh), (15, 16, parallel), (16, 17, mesh), (17, 18, parallel), (18, 19, mesh), (19, 20, cross)`

Co-Origin Probe 3 (Intermediate Rotation Trajectory Probing, sub_v3.0):
The image shows a chain of gears where adjacent gears are connected by direct meshing or by belts. Assuming the leftmost first gear rotates clockwise, infer the rotation direction of every gear along the connection order. Express each gear's rotation direction as a pair `(gear_id, direction)`: `gear_id` is the gear index starting from 1 along the connection order, and `direction` is `clockwise` or `counterclockwise`. Join the pairs with English commas in connection order and put the final answer inside `\boxed{}`, e.g., `\boxed{(1, counterclockwise), (2, clockwise), (3, counterclockwise)}`.

Ground Truth: `(1, clockwise), (2, clockwise), (3, counterclockwise), (4, clockwise), (5, counterclockwise), (6, clockwise), (7, clockwise), (8, counterclockwise), (9, counterclockwise), (10, clockwise), (11, counterclockwise), (12, clockwise), (13, counterclockwise), (14, clockwise), (15, counterclockwise), (16, counterclockwise), (17, clockwise), (18, clockwise), (19, counterclockwise), (20, clockwise)`

Figure 12: Base question and co-origin probes specification for the **Gear Rotation** task under Level 2 configurations.

Task: Connect Lines (Difficulty Level 2)



Configurations (L2): • **Cross Density**: `2 (L2)` • **Pair Count**: `4 (L2)`

Base Question: Report the connection mapping between each number on the left of the image and each letter on the right that are connected by the green lines. (The answer format should look like `1-A,2-N,3-D,4-D,5-O,6-I,7-M`)

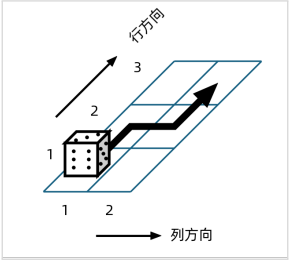
Ground Truth: `1-V, 2-L, 3-U, 4-B`

Co-Origin Probe 1 (Reverse Letter-to-Number Mapping, var1.0):
The image shows a column of numbers on the left and a column of letters on the right, connected by green curves. For each letter from top to bottom on the right, find the number on the left that is connected to it by a green line. Express each connection as a pair `(number, letter)`, where `number` is the left number and `letter` is the right letter. Join the pairs with English commas in top-to-bottom order of the right letters (no spaces inside a pair), and put the final answer inside `\boxed{}`, e.g., `\boxed{(3,C),(2,A),(1,B),(4,D)}`.

Ground Truth: `(2,L),(1,V),(3,U),(4,B)`

Figure 13: Base question and co-origin probe specification for the **Connect Lines** task under Level 2 configurations.

Task: Rolling Dice (Difficulty Level 2)



Configurations (L2): • **Reasoning Steps**: `3 (L2)` • **Turn Count**: `2 (L2)`

Base Question: The die starts on the grid cell at row 1, column 1. In the image the die rests on the starting cell and rolls cell by cell along the black arrow path, flipping 90 degrees over a cell edge on each roll. The numbers on opposite faces of the die always sum to 7. Roll the die from the starting position along the route shown in the image to the destination cell indicated by the arrow (destination: row 3, column 2). What is the number on the top face of the die at the end? Put the final answer inside `\boxed{}`, e.g., `\boxed{1}`.

Ground Truth: `4`

Co-Origin Probe 1 (Step-by-Step 6-Face State Tracking, sub1.0):
The die starts at row 1, column 1. Roll the die from the starting position along the route shown in the image to the destination indicated by the arrow (destination: row 3, column 2). Starting from the initial position, write out the numbers on the 6 faces of the die after each roll: a die state is a 7-tuple `(step_id, top, front, right, bottom, back, left)` giving the numbers on the top, front, right, bottom, back, and left faces after roll `step_id`, where `step_id` starts at 0 and `step_id = 0` denotes roll 0 (the initial face state). List, separated by English commas, all die-state 7-tuples from the initial position (roll 0) to the final position (roll 3) in ascending `step_id` order, and put the final answer inside `\boxed{}`, e.g., `\boxed{(step0, 5, 1, 4, 2, 6, 3), (step1, 6, 5, 4, 1, 2, 3)}`.

Ground Truth: `(step0, 3, 6, 5, 4, 1, 2), (step1, 6, 4, 5, 1, 3, 2), (step2, 2, 4, 6, 5, 3, 1), (step3, 4, 5, 6, 3, 2, 1)`

Figure 14: Base question and co-origin probe specification for the **Rolling Dice** task under Level 2 configurations.

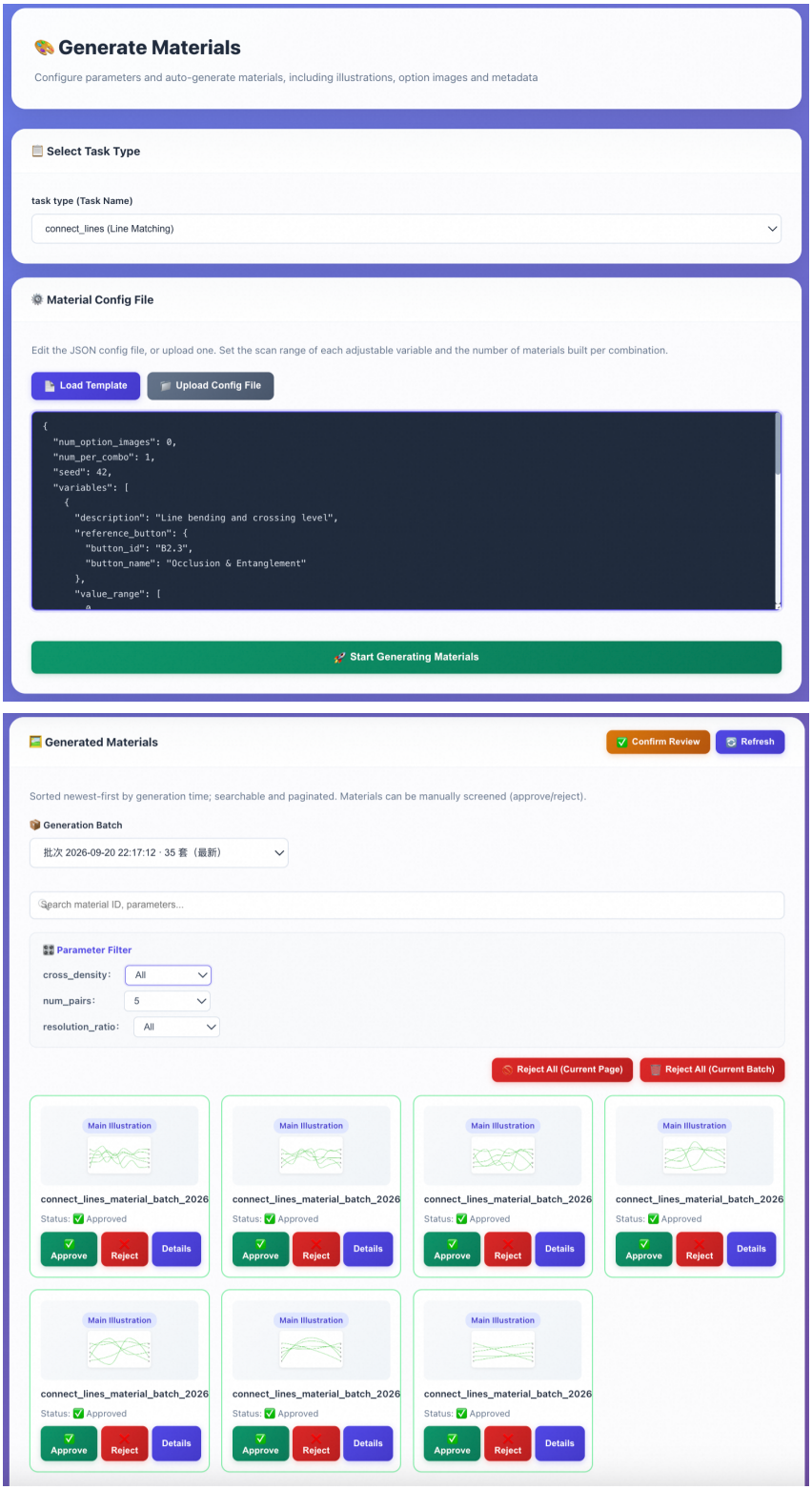


Figure 15: Stage 1: Material Generation.

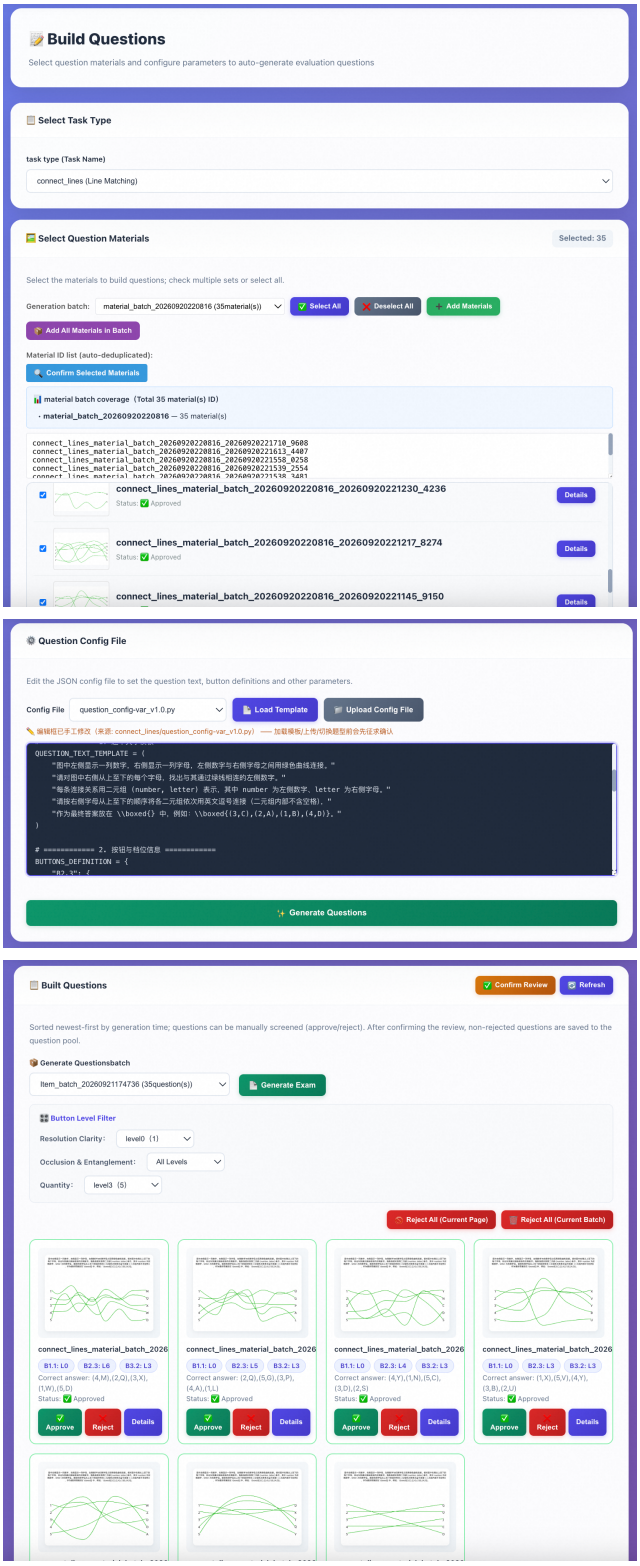


Figure 16: Stage 2: Question Construction.

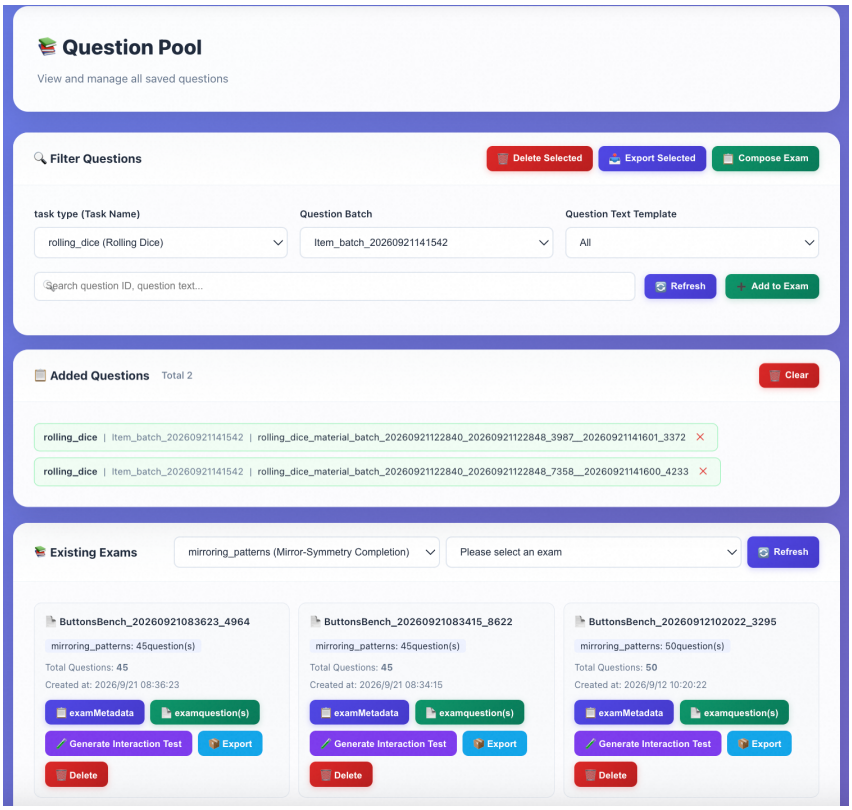


Figure 17: Stage 1: Material Generation.

Listing 1: LLM-as-a-Judge Prompt Template

```
JUDGE_PROMPT_TEMPLATE = """You are a strict answer evaluator.
You will receive:
1. A problem description and the corresponding image.
2. The correct answer (ground truth) given as a label or exact value.
3. The model's predicted answer
Your task:
- Decide whether the model's predicted answer is equivalent to the correct answer.
- For mathematical expressions, check equivalence:
    * Fractions can be equivalent if numerator/denominator are scaled equally.
    * Expressions like 3/5, \frac{3}{5}, 0.6 are considered equivalent.
    * For symbolic answers, treat trivial reorderings as equal if mathematically identical.
    * Ignore obvious format differences (spaces, LaTeX vs plain text).
- For multiple-choice questions, check if the model's choice letter matches the ground truth
    letter, OR if its value matches the correct option's value.
- Be careful: do NOT mark as correct if the value is only similar in magnitude but not exactly
     the same mathematically.
- If no valid answer is present, mark "Incorrect".
You must reply with only one word: "Correct" or "Incorrect".
Question:
{question}
Standard Answer:
{gdt}
Model Answer:
{box_answer}
"""
```