

# MORPHOBENCH: A Benchmark with Difficulty Adaptive to Model Reasoning

Xukai Wang<sup>1,2,5\*</sup>, Xuanbo Liu<sup>3,5\*</sup>, Mingrui Chen<sup>1,2,5\*</sup>, Haitian Zhong<sup>1,2,5\*</sup>, Xuanlin Yang<sup>4,5\*</sup>, Bohan Zeng<sup>4\*</sup>, Jinbo Hu<sup>4</sup>, Hao Liang<sup>4,5</sup>, Junbo Niu<sup>4</sup>, Xuchen Li<sup>1,2,5</sup>, Ruitao Wu<sup>3,5</sup>, Ruichuan An<sup>4</sup>, Yang Shi<sup>4</sup>, Liu Liu<sup>3</sup>, Xu-Yao Zhang<sup>1,2</sup>, Qiang Liu<sup>1,2</sup>, Zhouchen Lin<sup>4</sup>, Wentao Zhang<sup>4,5†</sup>, Bin Dong<sup>4,5†</sup>

<sup>1</sup>NLPR&MAIS, Institute of Automation, CAS   <sup>2</sup>School of Artificial Intelligence, UCAS  
<sup>3</sup>Beihang University   <sup>4</sup>Peking University   <sup>5</sup>Zhongguancun Academy  
wangxukai2025@ia.ac.cn, wentao.zhang@pku.edu.cn, dongbin@math.pku.edu.cn

## Abstract

With the advancement of powerful large-scale reasoning models, effectively evaluating the reasoning capabilities of these models has become increasingly important. However, existing benchmarks designed to assess the reasoning abilities of large models tend to be limited in scope and lack the flexibility to adapt their difficulty according to the evolving reasoning capacities of the models. To address this, we propose MORPHOBENCH, a benchmark that incorporates multidisciplinary questions to evaluate the reasoning capabilities of large models and can adjust and update question difficulty based on the reasoning abilities of advanced models. Specifically, we curate the benchmark by selecting and collecting complex reasoning questions from existing benchmarks and sources such as Olympiad-level competitions. Additionally, MORPHOBENCH adaptively modifies the analytical challenge of questions by leveraging key statements generated during the model’s reasoning process. Furthermore, it includes questions generated using simulation software, enabling dynamic adjustment of benchmark difficulty with minimal resource consumption. We have gathered over 1,300 test questions and iteratively adjusted the difficulty of MORPHOBENCH based on the reasoning capabilities of models such as GPT-5 and Gemini-3-Pro. MORPHOBENCH enhances the comprehensiveness and validity of model reasoning evaluation, providing reliable guidance for improving both the reasoning abilities and scientific robustness of large models. Code and datasets are released at <https://github.com/OpenDCAI/MorphoBench>.

## 1 Introduction

In recent years, large-scale pre-trained models have achieved remarkable progress, demonstrating unprecedented capabilities across natural language

processing, code generation, and multimodal understanding (Devlin et al., 2019; Achiam et al., 2023; Guo et al., 2024b; Bai et al., 2023; Guo et al., 2025a; Chen et al., 2025). Besides, there is a growing emphasis on strengthening their reasoning capabilities, especially in specialized academic domains such as mathematics, physics, logic, and related fields (Zhou et al., 2024; Muennighoff et al., 2025; Liu et al., 2023; Xu et al., 2025). This shift reflects the broader ambition of artificial intelligence: to move from surface-level understanding to robust and generalizable reasoning.

To effectively evaluate large models, several benchmarks such as MME-Reasoning (Yuan et al., 2025), SeePhys (Xiang et al., 2025), and HLE (Phan et al., 2025) have been proposed to measure reasoning abilities. Some models have even achieved gold-medal performance in competitions like the IMO (Huang and Yang, 2025) and IPHO (Qiu et al., 2025a). However, these benchmarks are static and cannot adapt to changes in a model’s reasoning proficiency. Moreover, although specialized agents may perform well in certain domains such as the IMO or IPHO, the coverage of current reasoning benchmarks remains narrow, as most focus on mathematics or physics problems. Many existing benchmarks, while intended for reasoning assessment, such as HLE (Phan et al., 2025), still rely on domain-specific knowledge, which tends to overestimate factual recall instead of true reasoning ability. Genuine reasoning should be evaluated through problems that involve complex logical inference based on simple or universally understood knowledge rather than the memorization of rare concepts. Therefore, a benchmark capable of dynamically adjusting difficulty according to a model’s reasoning ability, while covering multiple academic domains and emphasizing reasoning over knowledge rarity, is essential for accurate and stable evaluation.

To address these limitations, we propose MOR-

\*Contributed equally.

†Corresponding authors.

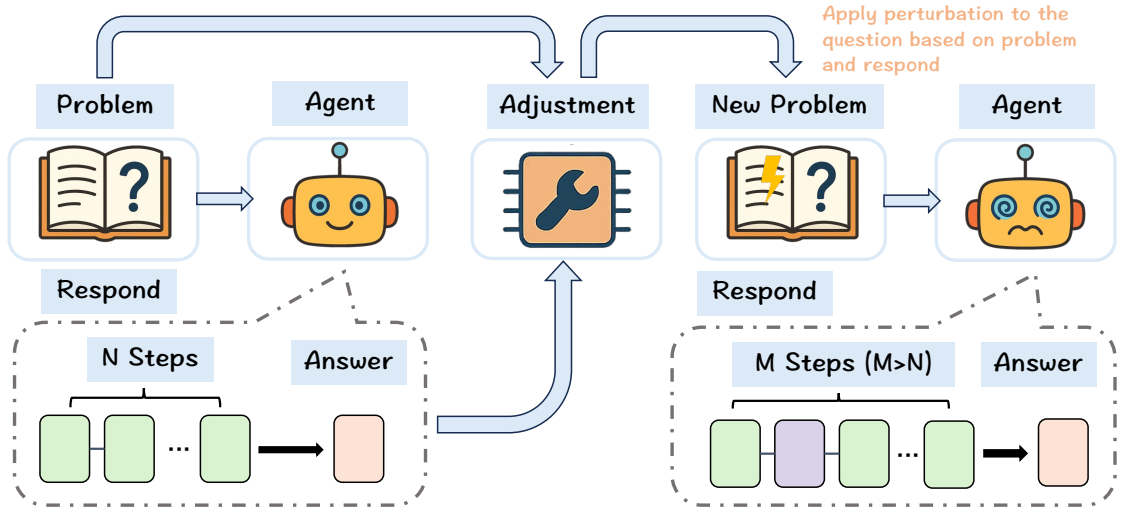


Figure 1: Overview of MORPHOBENCH.

MORPHOBENCH, a multi-disciplinary reasoning benchmark with difficulty adaptive to model performance. Unlike existing benchmarks, MORPHOBENCH dynamically adjusts question difficulty along two key dimensions: understanding conditions and constructing reasoning chains, enabling fair and comparable evaluation across models of different proficiency levels. It achieves this by modifying key statements within the model’s reasoning process, varying the clarity of problem conditions and introducing either guiding hints or distracting information to regulate reasoning complexity.

The main contributions of this paper can be summarized as follows:

- We introduce MORPHOBENCH, a novel benchmark that includes complex, reasoning-intensive problems in multiple disciplines. The benchmark supports adaptive difficulty calibration based on the model’s reasoning process, enabling fair and comparable evaluation across models with different levels of reasoning ability.
- MORPHOBENCH adapts difficulty along two dimensions: recognizing given conditions and constructing reasoning chains. MORPHOBENCH identifies critical points in the model’s problem-analysis process and adjusts questions accordingly. It widens the reasoning gap through targeted perturbations of the textual grounding linked to model-identified visual cues, adjustments to the form and structure of intermediate reasoning, and parameterized controls under verified automatic generation, enabling principled and scalable diffi-

culty evolution.

- We introduce a reviewer-judge agent, a modular LLM-as-judge evaluation framework that decomposes multimodal model outputs into step-level evidence and scores them along complementary axes of correctness, reasoning quality, and hint-following. Our framework enables fine-grained, interpretable diagnosis beyond final-answer accuracy.

## 2 Related Work

### 2.1 Large Reasoning Models

Frontier models increasingly treat reasoning as a first-class capability: they are designed and trained to allocate more inference-time computation to difficult prompts, perform multi-step deduction, and verify intermediate steps before producing a final answer (OpenAI, 2025a; Comanici et al., 2025; Guo et al., 2025a, 2024a; Su et al., 2025). In the multimodal regime, recent reasoning-optimized systems further encourage longer deliberation, including variants that incorporate images into intermediate reasoning (OpenAI, 2025b; Comanici et al., 2025; Bai et al., 2025; Liang et al., 2025).

These trends make it increasingly important to adopt diagnostic evaluations: moving beyond a single accuracy metric and using controlled test suites to better characterize model reasoning behavior and generalization.

### 2.2 Evaluation Benchmarks for Large Models

Evaluating large models requires robust benchmarks that reflect their capabilities across knowledge and reasoning (Hendrycks et al., 2020; Wang

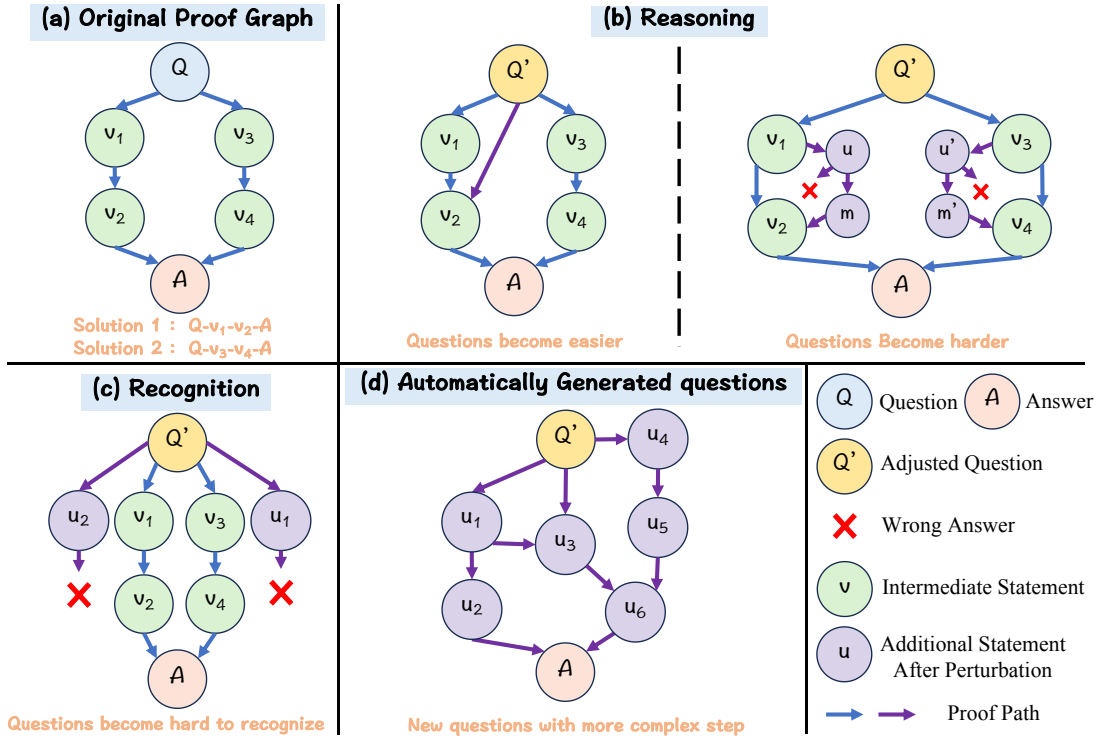


Figure 2: Demonstration of MORPHOBENCH’s problem difficulty adjustment pipelines.

et al., 2024). As models evolve, benchmarks for multimodal understanding and reasoning have become increasingly important, including benchmarks that test integration across visual and textual modalities (Lu et al., 2023; Yu et al., 2023; Yue et al., 2024; Zhang et al., 2024; Fu et al., 2025; Shi et al., 2025; Hu et al., 2025). In parallel, reasoning-focused benchmarks evaluate complex reasoning and practical problem solving (e.g., code and long-horizon tasks) (Zheng et al., 2025; Yuan et al., 2025; Guo et al., 2025b), while domain-specific benchmarks probe specialized scientific QA abilities (Phan et al., 2025; Ruan et al., 2025; Shen et al., 2025; Xiang et al., 2025; Li et al., 2024). However, many benchmarks remain largely static and provide limited support for capability-adaptive evaluation via controlled difficulty adjustments, which makes fair comparison across models with different reasoning profiles challenging.

### 3 MORPHOBENCH

#### 3.1 Data Collection

To evaluate large-scale models’ reasoning capabilities across disciplines, MORPHOBENCH aggregates explicit-reasoning questions from three sources (see Fig. 3).

**(1) Open-source benchmarks.** We incorporate reasoning-oriented items from *Humanity’s*

*Last Exam (HLE)* (Phan et al., 2025), *MME-Reasoning* (Yuan et al., 2025), and a subset of historical reasoning questions from *HistBench* (Qiu et al., 2025b).

**(2) Olympiad-level competition problems.** We collect challenging problems in mathematics, physics, and chemistry from established competitions and national Olympiads (e.g., CMO/Putnam/IMO/USAMO; CPhO/CChO), together with coach-designed Olympiad training problems.

**(3) Expert-designed complex reasoning scenarios.** We additionally construct new questions via template-based automatic generation for structured reasoning settings (e.g., black-box circuits), with answers verified by simulation for objectivity and reproducibility. The generation pipeline is described in Sec. 3.3.

All questions are standardized with a unified style guide and undergo at least two rounds of expert review for correctness and consistency. This initial dataset is denoted as MORPHO-R(v0).

#### 3.2 Preliminary Analysis

To illustrate how question difficulty can be systematically adjusted according to the reasoning capabilities of large-scale models, we first define the difficulty levels of questions in MORPHOBENCH.

Model	Engineering	Mathematics	Natural Sciences	Social Sciences	Other
Total (share)	439 (28.38%)	560 (36.21%)	250 (16.16%)	86 (5.56%)	212 (13.70%)
Gemini-2.5-flash	13.64	45.29	28.80	59.34	34.02
Gemini-2.5-pro	15.91	48.91	28.80	<b>65.93</b>	40.72
Gemini-3-pro	24.09	<b>66.12</b>	<b>33.60</b>	53.85	<b>48.45</b>
GPT-5.1	<b>48.64</b>	61.52	32.40	51.65	45.88
Grok-4	8.13	60.93	31.06	49.45	14.67
o3	32.27	51.45	32.40	54.95	33.51
o4-mini	15.91	54.35	28.00	49.45	32.99

Table 1: **Subject-level answer correctness on MORPHO-R(v0)**. We report accuracy (Acc; %, higher is better) across five disciplines. **Total (share)** reports the number of questions and their proportion in the full dataset ( $N = 1307$ ). Best results per column are in bold.

Recent LLMs increasingly demonstrate planning-like behaviors, outlining intermediate steps before producing the final solution (Gui et al., 2025; Rawat et al., 2025). Inspired by this observation, we formalize the solving process as a search problem on a *directed proof graph* (Wei et al., 2023; Yao et al., 2023) and analyze how the complexity of this graph, which reflects the model’s reasoning depth and branching structure, can be adjusted to control the difficulty of a question.

### 3.2.1 Reasoning as Path Search in a Proof Graph

For a reasoning question  $Q$ , we construct a directed proof graph

$$G_Q = (V, E, c). \quad (1)$$

Each vertex  $v \in V$  encodes an intermediate statement or subconclusion, and each directed edge  $e = (v, v') \in E$  represents a single logically valid inference step. The edge weight  $c(e) > 0$  captures the expected computational difficulty for an LLM to move from state  $v$  to  $v'$  without introducing additional intermediate statements.

The start vertex  $s(Q)$  corresponds to the original problem statement and the terminal vertex  $t(Q)$  denotes a fully verified answer. For any valid reasoning path

$$\pi = (v_0, \dots, v_k) \quad \text{with} \quad v_0 = s(Q), v_k = t(Q), \quad (2)$$

the accumulated cost is

$$\text{Cost}(\pi) = \sum_{i=0}^{k-1} c(v_i, v_{i+1}). \quad (3)$$

We define the intrinsic difficulty of  $Q$  under a model’s reasoning policy as the expected cost of

correctly deriving the answer over valid paths:

$$\begin{aligned} L(Q) &= \mathbb{E}_{\pi \sim P(\pi|Q)} [\text{Cost}(\pi)] \\ &= \sum_{\pi: s \rightarrow t} P(\pi | Q) \text{Cost}(\pi) \end{aligned} \quad (4)$$

where  $P(\pi | Q)$  denotes the model-assigned probability of following a valid reasoning path  $\pi$ . This expectation-based formulation captures both step-level computational costs and the diversity of plausible reasoning trajectories.

### 3.2.2 Question Modification and Information Gap

We formalize question modification as an algorithm  $\mathcal{R}$  that appends a hint  $\tau$  to the original question, yielding  $Q' = \mathcal{R}(Q, \tau)$ . With respect to the target answer  $A$ , we define the *information gap* induced by the modification as

$$\Delta I = K(A | Q') - K(A | Q), \quad (5)$$

where  $K(A | Q)$  serves as an information-theoretic proxy for the effective complexity of producing  $A$  given  $Q$ . Intuitively,  $\Delta I \leq 0$  corresponds to helpful or redundant modifications, whereas  $\Delta I > 0$  indicates misleading or irrelevant adjustments. In what follows, we focus on misleading modifications with  $\Delta I > 0$ .

### 3.2.3 Impact of Misleading Modifications

Let  $\text{Fail}(Q, B)$  denote the event that an agent fails to reach  $t(Q)$  within a fixed compute budget  $B$  under its reasoning policy. Misleading modifications ( $\Delta I > 0$ ) expand the effective search space of the proof graph by introducing spurious alternatives, thereby increasing the expected traversal cost and making failures more likely under a fixed budget:

$$\Pr[\text{Fail}(Q', B)] > \Pr[\text{Fail}(Q, B)]. \quad (6)$$

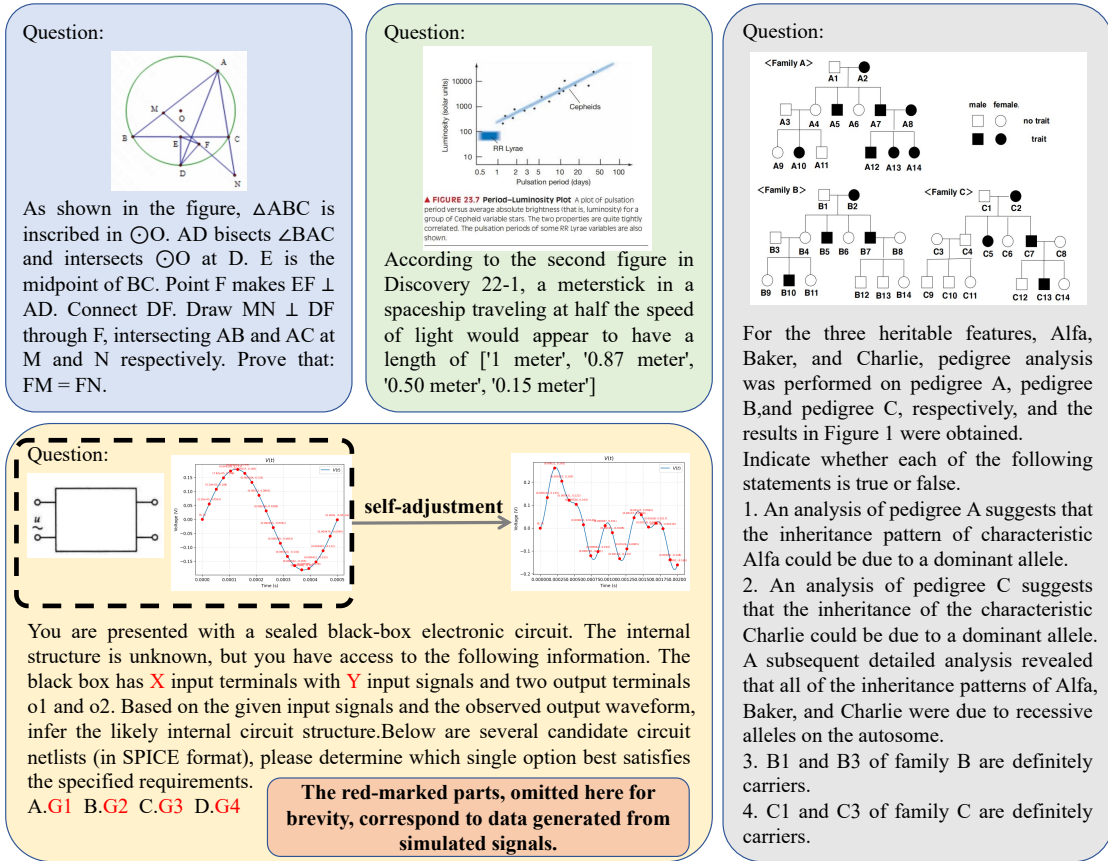


Figure 3: Testing examples from MORPHOBENCH.

We provide formal assumptions and proof sketches in Appendix A.2.

### 3.3 Difficulty Adaptation

We provide our three main difficulty adaptation schemes. Detailed implementation procedures are deferred to Appendix A.3.

**Adaptation based on agent reasoning.** Shaping the agent reasoning process is a direct and effective way to control problem difficulty and widen the gap between question and answer. As shown in Fig. 2(b), we adjust difficulty by introducing progressively revealable hints derived from key reasoning steps: higher-disclosure, more explicit hints lower difficulty, whereas lower-disclosure hints leave more exploration space; under the hard setting, we additionally inject plausible but misleading hints to increase difficulty without changing the original question or answer. To make this process controllable and auditable, we first build a proof graph from the reference solution trace: intermediate statements are extracted and linked according to their inferential dependencies, and each graph element is attached with the originating step indices for traceability. Based on the proof graph, we

then generate a tiered hint set ordered from coarse guidance to increasingly specific cues. Early tiers mainly indicate what objects to focus on and what basic setup or auxiliary construction to introduce, while later tiers progressively reveal pivotal intermediate relations or conclusions that unlock the main solution path, yet still avoid stating the final answer. Each hint is aligned with evidence steps in the trace, and we apply post-generation validation and repair to ensure reliable downstream use. The algorithm therefore enables configurable control of problem complexity through hint design and makes the generated hints interpretable and actionable, supporting finer-grained difficulty evolution.

**Adaptation based on agent recognition.** MORPHOBENCH increases the reasoning cost between questions and answers by perturbing the visual cues most critical to the model, making the model more prone to reasoning errors as illustrated in Fig. 2(c). Instead of relying on predefined annotations, the model itself first indicates which elements it considers essential. These elements are then deliberately obfuscated at the text level, for example by introducing ambiguous wording or partially masking

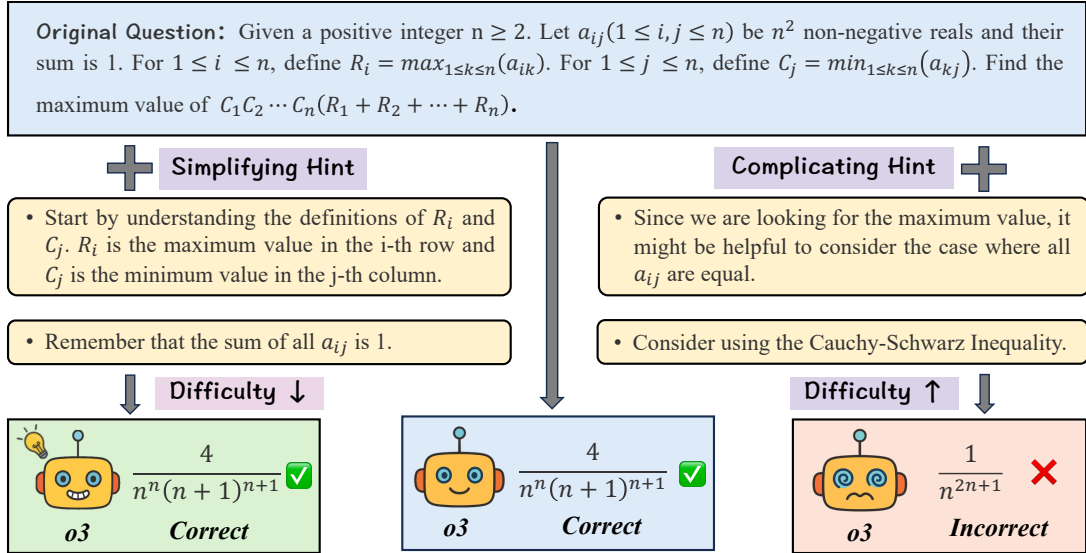


Figure 4: Examples for adaptation based on agent reasoning.

key terms, thereby hindering precise interpretation. Unlike random textual noise, such agent-driven perturbations directly target the linguistic features most relied upon, making them more challenging. If the model continues to answer correctly under these conditions, it demonstrates strong robustness and generalization; conversely, performance degradation reveals over-dependence on localized textual cues. This strategy thus provides a principled means of difficulty adjustment, testing whether the model remains effective when its key features are perturbed.

**Adaptation for automatically generated questions.** In MORPHOBENCH, automatic question generation involves two central challenges: ensuring validity and regulating difficulty, as demonstrated in Fig. 2(d). To guarantee validity, we incorporate external simulation software, such as circuit simulators, to systematically verify the correctness of generated outputs. To regulate difficulty, we adjust key generation parameters. Specifically, in circuit black-box tasks, difficulty is modulated by varying the number of exposed terminals, with a larger number increasing the complexity of inferring the internal structure. In “spot the different one” tasks, difficulty is controlled either by selecting character pairs with higher visual similarity or by expanding the grid size, thereby imposing greater demands on visual discrimination. These mechanisms allow MORPHOBENCH to evolve difficulty automatically: as terminal counts or grid complexity grow, the tasks become progressively harder. This enables continuous challenge for mod-

els and supports scalable evaluation of reasoning and multimodal understanding.

## 4 Experiment

### 4.1 Implementation Details

We evaluate frontier multimodal reasoning models, including Gemini-2.5-Flash, Gemini-2.5-Pro, GPT-5.1, Grok-4, and the OpenAI o-series (o3, o4-mini).

We first benchmark all models on the original dataset MORPHO-R(v0) and report discipline-level results across mathematics, engineering, natural sciences, and social sciences. Starting from MORPHO-R(v0), we construct two adaptation families by editing each instance based on o3’s responses to expose different failure modes.

**Agent-reasoning adaptation:** The MORPHO-R family keeps the same underlying problems as MORPHO-R(v0) but adjusts the reasoning demands of the prompts and hints. Specifically, we build MORPHO-R(Lite), which simplifies the required reasoning chain, and MORPHO-R(Complex), which increases reasoning depth by rewriting lemma hints to control the granularity and number of intermediate steps.

For each instance in MORPHO-R(v0), we create two derived instances for MORPHO-R(Lite) and two for MORPHO-R(Complex); thus, the derived datasets are each twice the size of MORPHO-R(v0).

**Agent-recognition adaptation:** The MORPHO-P family targets multimodal perception robustness. We derive MORPHO-P(Perturbed) by perturbing critical textual and visual cues in 476 multimodal instances, while keeping the underlying task in-

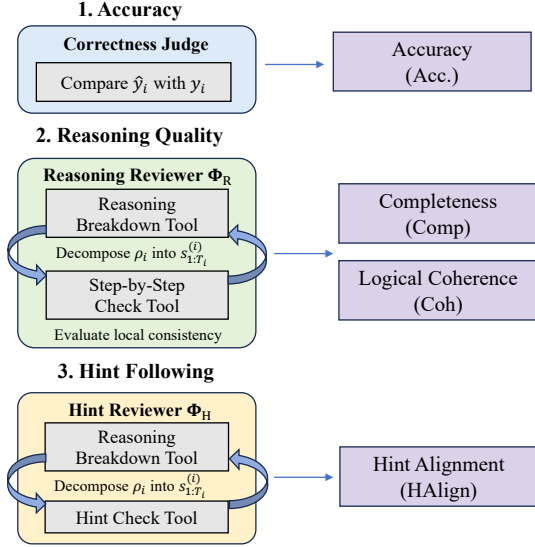


Figure 5: Evaluation framework of MORPHOBENCH.

tent unchanged, to evaluate whether models remain reliable under controlled perception disturbances.

**Automatic-generation adaptation.** We additionally create a graded circuit-reasoning benchmark, denoted as MORPHO-G, by varying the number of terminals in black-box circuit questions to obtain a spectrum of difficulty.

## 4.2 Evaluation Metrics

We evaluate multimodal reasoning models on MorphoBench along three complementary dimensions: answer correctness, reasoning quality, and hint following. For each example  $d_i = (x_i, y_i, h_i)$ ,  $x_i$  denotes the multimodal input,  $y_i$  the ground-truth answer, and  $h_i$  an optional set of provided hints (set to  $\emptyset$  when absent). Given a model  $M$ , we obtain the output  $a_i = M(x_i) = (\hat{y}_i, \rho_i)$ , where  $\hat{y}_i$  is the final answer and  $\rho_i$  is the model’s reasoning trace. All judge-based metrics follow a unified LLM-as-judge formulation:

$$\begin{aligned} \text{Score} &= \text{LLM}(p, a, r) = G_\phi(p, \text{Evidence}), \\ \text{Evidence} &= F(\Phi(p, a)), \end{aligned} \quad (7)$$

where  $p$  is the metric-specific rubric prompt,  $\Phi$  is a tool-based reviewer that converts the model output into structured step-level reviews  $r$ ,  $F$  aggregates  $r$  into evidence, and  $G_\phi$  is the judge model producing the final score. A more detailed formalization is provided in Appendix B.

**Accuracy.** Correctness is computed by directly comparing the predicted final answer with the

ground truth:

$$\text{Acc} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}[\hat{y}_i = y_i]. \quad (8)$$

**Reasoning Quality.** To evaluate reasoning beyond final-answer correctness, we report two scores: Completeness and Logical Coherence. For each example, the model produces a reasoning trace  $\rho_i$ . We define the reasoning reviewer  $\Phi_R$  as a two-stage review procedure: (1) step breakdown, which decomposes  $\rho_i$  into an ordered step sequence  $\mathbf{s}_{1:T_i}^{(i)} = (s_1^{(i)}, \dots, s_{T_i}^{(i)})$ ; (2) step checking, which evaluates local consistency between adjacent steps  $(s_{t-1}^{(i)}, s_t^{(i)})$ . The resulting step-level reviews are aggregated into an evidence summary  $\text{Evidence}_i^R$ , which is scored under the rubric prompt  $p_R$ . Intuitively, Completeness measures whether essential intermediate steps are covered, while Logical Coherence measures whether the chain is internally consistent throughout.

$$\begin{aligned} (\text{Comp}_i, \text{Coh}_i) &= G_\phi(p_R, \text{Evidence}_i^R), \\ \text{Comp}_i, \text{Coh}_i &\in [0, 100]. \end{aligned} \quad (9)$$

**Hint Following.** For hint-bearing examples ( $h_i \neq \emptyset$ ), we additionally measure Hint Alignment, which evaluates whether the model’s reasoning behavior follows the provided hints. We define the hint reviewer  $\Phi_H$  as: (1) step breakdown, producing  $\mathbf{s}_{1:T_i}^{(i)}$ ; and (2) hint checking, which examines each step  $s_t^{(i)}$  conditioned on the question and the hint set  $h_i$ , and assesses whether any deviation is justified. Step-level hint reviews are aggregated into  $\text{Evidence}_i^H$ , which is scored under  $p_H$  to produce the final score.

$$\begin{aligned} \text{HALign}_i &= G_\phi(p_H, \text{Evidence}_i^H), \\ \text{HALign}_i &\in [0, 100]. \end{aligned} \quad (10)$$

Notably, some hints are intentionally misleading. Therefore, a higher Hint Alignment score is not universally better, but rather reflects whether the model can recognize misleading hints and still maintain a correct reasoning trajectory.

## 4.3 Main Comparison Results

**Cross-disciplinary performance.** We report a discipline-level breakdown on MORPHO-R(v0) in Table 1 to characterize cross-disciplinary reasoning performance. The table also reports **Total (share)** to show the subject-wise distribution of questions

Model	MORPHO-R (Lite)	MORPHO-R (v0)	MORPHO-R (Complex)	MORPHO-P (v0)	MORPHO-P (Perturbed)
Gemini-2.5-flash	39.67 (+3.56)	36.11	30.72 (-5.39)	38.24	34.24 (-4.00)
Gemini-2.5-pro	44.87 (+5.39)	39.48	37.49 (-1.99)	39.71	37.61 (-2.10)
Gemini-3-pro	51.26 (+1.91)	49.35	<b>42.96 (-6.39)</b>	<b>51.47</b>	37.82 (-13.65)
GPT-5.1	<b>55.28 (+4.51)</b>	<b>50.77</b>	40.11 (-10.66)	45.80	39.71 (-6.09)
Grok-4	41.87 (+4.14)	37.73	31.34 (-6.39)	42.48	37.77 (-4.71)
o3	46.06 (+3.90)	42.16	33.32 (-8.84)	46.64	<b>40.55 (-6.09)</b>
o4-mini	42.69 (+3.36)	39.33	31.37 (-7.96)	46.64	39.29 (-7.35)

Table 2: **Answer correctness on MORPHOBENCH.** We report accuracy (Acc; %, higher is better) across MORPHO-R and MORPHO-P splits; parentheses denote absolute changes (percentage points) relative to the corresponding v0 baseline. Best results are in bold.

Model	MORPHO-R (Lite)	MORPHO-R (v0)	MORPHO-R (Complex)	MORPHO-P (v0)	MORPHO-P (Perturbed)
Gemini-2.5-flash	60.33 (+0.47) / 67.89 (+0.86)	59.86 / 67.03	57.56 (-2.30) / 64.57 (-2.46)	67.64 / 73.64	65.88 (-1.76) / 72.62 (-1.02)
Gemini-2.5-pro	59.19 (-0.50) / 66.06 (-0.67)	59.69 / 66.73	58.32 (-1.37) / 65.03 (-1.70)	66.64 / 71.49	64.58 (-2.06) / 69.99 (-1.50)
Gemini-3-pro	62.06 (+2.81) / 65.41 (+2.70)	59.25 / 62.71	62.86 (+3.61) / 66.01 (+3.30)	66.78 / 70.07	58.09 (-8.69) / 61.36 (-8.71)
GPT-5.1	63.66 (+0.37) / 71.49 (+1.36)	63.29 / 70.13	58.00 (-5.29) / 66.16 (-3.97)	66.93 / 74.08	68.55 (+1.62) / 74.68 (+0.60)
Grok-4	70.94 (+0.07) / 73.32 (-0.23)	70.87 / 73.55	64.96 (-5.91) / 66.80 (-6.75)	76.41 / 78.63	78.25 (+1.84) / 80.31 (+1.68)
o3	76.93 (+3.73) / 80.14 (+4.20)	73.20 / 75.94	<b>73.86 (+0.66) / 77.05 (+1.11)</b>	78.40 / 80.61	79.67 (+1.27) / 81.88 (+1.27)
o4-mini	<b>78.37 (+0.63) / 83.71 (+0.61)</b>	<b>77.74 / 83.10</b>	73.58 (-4.16) / <b>78.96 (-4.14)</b>	<b>82.67 / 86.78</b>	<b>84.13 (+1.46) / 88.24 (+1.46)</b>

Table 3: **Reasoning quality on MORPHOBENCH.** Each cell reports two reasoning quality scores (left: Completeness, right: Logical Coherence; both in  $[0, 100]$ ); parentheses denote absolute changes (points) relative to the corresponding v0 baseline. Best results are in bold.

Model	MORPHO-R (Lite)	MORPHO-R (Complex)
Gemini-2.5-flash	86.73	67.74 (-18.99)
Gemini-2.5-pro	86.48	60.48 (-26.00)
Gemini-3-pro	80.43	57.11 (-23.32)
GPT-5.1	88.72	68.87 (-19.85)
Grok-4	78.46	56.29 (-22.17)
o3	88.30	69.21 (-19.09)
o4-mini	90.12	69.72 (-20.40)

Table 4: **Hint following on MORPHOBENCH hint-enabled splits.** We report Hint Alignment (HAlign;  $[0, 100]$ ); parentheses after R(Complex) indicate the absolute difference to R(Lite).

in MORPHOBENCH. The leading model varies by discipline: GPT-5.1 achieves the highest accuracy in Engineering (48.64%), Gemini-3-Pro leads Mathematics (66.12%), and Gemini-2.5-Pro attains the best performance in Social Sciences (65.93%). Notably, Engineering exhibits a more polarized pattern: GPT-5.1 and o3 substantially outperform the rest (48.64% and 32.27%, respectively), while other models remain below 25%.

**Influence of adjustment based on agent recognition and reasoning.** As shown in Table 2, we compare models across the two adaptation families. Within the MORPHO-R family, where questions are held fixed and only lemma-level hints are modified, all models improve on R(Lite) relative to

R(v0), but degrade on R(Complex). This pattern suggests that helpful hints improve performance, whereas misleading hints effectively increase task difficulty. In terms of absolute accuracy, GPT-5.1 leads on R(Lite) and R(v0) (55.28% and 50.77%), while Gemini-3-Pro attains the best performance on the misleading-hint split R(Complex) (42.96%), suggesting that models differ substantially in their accuracy retention under misleading hints.

For recognition-driven perturbations, Table 2 shows a consistent accuracy drop from P(v0) to P(Perturbed) across all models, with notably different drop magnitudes. While Gemini-3-Pro attains the highest accuracy on the unperturbed split P(v0) (51.47%), o3 achieves the best performance under perturbation on P(Perturbed) (40.55%), indicating that strong clean performance does not necessarily translate to robustness when key cues are perturbed.

To disentangle correctness from reasoning behaviors, we further report reasoning-quality diagnostics (Table 3). Across splits, the top reasoning-quality scores are generally achieved by the OpenAI o-series (especially o4-mini), while other models show larger fluctuations under hint/recognition adaptations. Importantly, the change in reasoning quality does not always mirror the change in accuracy. For example, GPT-5.1 exhibits a marked accuracy drop on R(Complex), accompanied by a decrease in both Completeness and Coherence, whereas on P(Perturbed) its reasoning-quality

Difficulty Level	o3 Acc. (%)	Gemini-2.5-Pro Acc. (%)
1	48.3	75.9
2	30.0	36.7
3	48.0	16.0
4	23.1	7.7
5	40.7	0.0
6	39.3	7.1
7	54.2	12.5
8	57.7	7.7
9	44.0	0.0
10	34.8	13.0

Table 5: Model performance of o3 and Gemini-2.5 Pro on the MORPHO-G. The circuit black-box problem is a single-choice question with six options in total.

scores slightly increase despite reduced accuracy, illustrating a partial decoupling between “producing a well-structured trace” and “arriving at the correct answer” under perturbations.

Finally, we report Hint Alignment on the hint-enabled splits (Table 4). On R(Lite), all models achieve high alignment (all above 78%). When hints become intentionally misleading in R(Complex), alignment drops markedly for every model, suggesting that models can partially recognize and deviate from misleading hints. We emphasize that hints are not instructions; thus, Hint Alignment is a behavioral diagnostic of how a model’s reasoning trace depends on or resists external guidance.

**Influence of adjustment for automatically generated questions.** For the circuit black-box tasks, we conducted evaluations on o3 and Gemini-2.5-Pro. Before testing, we systematically defined difficulty levels for black-box problems. Specifically, the difficulty was divided into ten levels based on the number of external terminals. Each level corresponds to the number of input terminals on the black box, which in turn specifies the number of alternating current (AC) voltages simultaneously applied to these terminals. As the number of terminals increases, the reasoning process becomes inherently more complex, resulting in progressively more challenging tasks. The experimental results are summarized in Table 5.

As shown in the results, difficulty stratification strongly affects Gemini-2.5-Pro: as difficulty increases from level 1 to 10, its accuracy drops sharply from 75.9% to 0–13%, remaining low at higher levels. In contrast, o3’s accuracy fluctuates between 30% and 58% without a clear down-

ward trend. This shows that the designed difficulty partition effectively suppresses Gemini-2.5-Pro’s performance, confirming the sensitivity of the difficulty design, while o3 exhibits weaker sensitivity. The difference likely results from distinct training distributions and inference strategies, as o3 can utilize external tools for analysis and problem solving, whereas Gemini-2.5-Pro aligns more closely with the intended progressive difficulty response.

## 5 Conclusion

In this paper, we present MORPHOBENCH, a multimodal benchmark designed to evaluate frontier models on challenging, cross-disciplinary reasoning. Starting from MORPHO-R(v0), we construct difficulty-controlled variants that probe distinct failure modes, including (i) reasoning sensitivity to supportive versus misleading lemma-level hints and (ii) robustness to recognition-level perturbations of critical cues.

Alongside these adaptations, we provide structured attribute annotations to support fine-grained analysis of model behaviors. Our experiments across state-of-the-art models demonstrate that performance differs substantially across disciplines and difficulty settings, and that standard accuracy can diverge from reasoning-quality and hint-related diagnostics. We hope MORPHOBENCH serves as a useful testbed for developing more reliable multimodal reasoning systems and for diagnosing their robustness under controlled difficulty shifts.

## 6 Limitations

While MORPHOBENCH encompasses a wide variety of problem types and allows for dynamic difficulty adjustment, it primarily relies on adapting existing problems. Although modifying test questions according to the model’s reasoning process appropriately tailors the difficulty to its abilities, our current framework still falls short of generating entirely novel scientific reasoning scenarios from scratch. In future work, we plan to build upon our methodology by leveraging the failure modes observed in model reasoning to enable the automated generation of entirely new, complex questions grounded in reference literature.

## 7 Acknowledgments

This work was supported by the Zhongguancun Academy Project C20250204.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.
- Tianyi Bai, Zengjie Hu, Fupeng Sun, Jiantao Qiu, Yizhen Jiang, Guangxin He, Bohan Zeng, Conghui He, Binhang Yuan, and Wentao Zhang. 2025. Multi-step visual reasoning with visual tokens scaling and verification. *arXiv preprint arXiv:2506.07235*.
- Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. 2025. Sft or rl? an early investigation into training rl-like reasoning large vision-language models. *arXiv preprint arXiv:2504.11468*.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.
- Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118.
- Runquan Gui, Zhihai Wang, Jie Wang, Chi Ma, Huiling Zhen, Mingxuan Yuan, Jianye Hao, Defu Lian, Enhong Chen, and Feng Wu. 2025. *Hypertree planning: Enhancing llm reasoning via hierarchical thinking*. Preprint, arXiv:2505.02322.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025a. Deepseek-rl: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, and 1 others. 2024a. Deepseek-coder: When the large language model meets programming—the rise of code intelligence. *arXiv preprint arXiv:2401.14196*.
- Meng-Hao Guo, Jiajun Xu, Yi Zhang, Jiayi Song, Haoyang Peng, Yi-Xuan Deng, Xinzhi Dong, Kiyohiro Nakayama, Zhengyang Geng, Chen Wang, and 1 others. 2025b. R-bench: Graduate-level multi-disciplinary benchmarks for llm & mllm complex reasoning evaluation. *arXiv preprint arXiv:2505.02018*.
- Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024b. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. 2025. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos. *arXiv preprint arXiv:2501.13826*.
- Yichen Huang and Lin F. Yang. 2025. *Gemini 2.5 pro capable of winning gold at imo 2025*. Preprint, arXiv:2507.15855.
- Zekun Li, Xianjun Yang, Kyuri Choi, Wanrong Zhu, Ryan Hsieh, HyeonJung Kim, Jin Hyuk Lim, Sungyoung Ji, Byungju Lee, Xifeng Yan, and 1 others. 2024. Mmsci: A multimodal multi-discipline dataset for phd-level scientific comprehension. In *AI for Accelerated Materials Design-Vienna 2024*.
- Hao Liang, Ruitao Wu, Bohan Zeng, Junbo Niu, Wentao Zhang, and Bin Dong. 2025. Multimodal reasoning for science: Technical report and 1st place solution to the icml 2025 seephys challenge. *arXiv preprint arXiv:2509.06079*.
- Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.
- OpenAI. 2025a. *Gpt-5*.

- OpenAI. 2025b. [o3: Advanced reasoning model](#).
- Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.
- Jiahao Qiu, Jingzhe Shi, Xinzhe Juan, Zelin Zhao, Jiayi Geng, Shilong Liu, Hongru Wang, Sanfeng Wu, and Mengdi Wang. 2025a. [Physics supernova: Ai agent matches elite gold medalists at ipho 2025](#). *Preprint*, arXiv:2509.01659.
- Jiahao Qiu, Fulian Xiao, Yimin Wang, Yuchen Mao, Yijia Chen, Xinzhe Juan, Shu Zhang, Siran Wang, Xuan Qi, Tongcheng Zhang, and 1 others. 2025b. On path to multimodal historical reasoning: Histbench and histagent. *arXiv preprint arXiv:2505.20246*.
- Mrinal Rawat, Ambuje Gupta, Rushil Goomer, Alessandro Di Bari, Neha Gupta, and Roberto Pieraccini. 2025. [Pre-act: Multi-step planning and reasoning improves acting in llm agents](#). *Preprint*, arXiv:2505.09970.
- Jiacheng Ruan, Dan Jiang, Xian Gao, Ting Liu, Yuzhuo Fu, and Yangyang Kang. 2025. Mme-sci: A comprehensive and challenging science benchmark for multimodal large language models. *arXiv preprint arXiv:2508.13938*.
- Hui Shen, Taiqiang Wu, Qi Han, Yunta Hsieh, Jizhou Wang, Yuyue Zhang, Yuxin Cheng, Zijian Hao, Yuansheng Ni, Xin Wang, and 1 others. 2025. Phyx: Does your model have the "wits" for physical reasoning? *arXiv preprint arXiv:2505.15929*.
- Yang Shi, Huanqian Wang, Wulin Xie, Huanyao Zhang, Lijie Zhao, Yi-Fan Zhang, Xinfeng Li, Chaoyou Fu, Zhuoer Wen, Wenting Liu, and 1 others. 2025. Mme-videoocr: Evaluating ocr-based capabilities of multimodal llms in video scenarios. *arXiv preprint arXiv:2505.21333*.
- Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, and 1 others. 2025. Openthinking: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. [Chain-of-thought prompting elicits reasoning in large language models](#). *Preprint*, arXiv:2201.11903.
- Kun Xiang, Heng Li, Terry Jingchen Zhang, Yinya Huang, Zirong Liu, Peixin Qu, Jixi He, Jiaqi Chen, Yu-Jie Yuan, Jianhua Han, and 1 others. 2025. Seep-hys: Does seeing help thinking?—benchmarking vision-based physics reasoning. *arXiv preprint arXiv:2505.19099*.
- Ran Xu, Yuchen Zhuang, Yishan Zhong, Yue Yu, Xianguo Tang, Hang Wu, May D Wang, Peifeng Ruan, Donghan Yang, Tao Wang, and 1 others. 2025. Medagentgym: Training llm agents for code-based medical reasoning at scale. *arXiv preprint arXiv:2506.04405*.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. [Tree of thoughts: Deliberate problem solving with large language models](#). *Preprint*, arXiv:2305.10601.
- Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.
- Jiakang Yuan, Tianshuo Peng, Yilei Jiang, Yiting Lu, Renrui Zhang, Kaituo Feng, Chaoyou Fu, Tao Chen, Lei Bai, Bo Zhang, and 1 others. 2025. Mme-reasoning: A comprehensive benchmark for logical reasoning in mllms. *arXiv preprint arXiv:2505.21327*.
- Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.
- Jianrui Zhang, Mu Cai, and Yong Jae Lee. 2024. Vinoground: Scrutinizing llms over dense temporal reasoning with short videos. *arXiv preprint arXiv:2410.02763*.
- Zihan Zheng, Zerui Cheng, Zeyu Shen, Shang Zhou, Kaiyuan Liu, Hansen He, Dongruixuan Li, Stanley Wei, Hangyi Hao, Jianzhu Yao, and 1 others. 2025. Livecodebench pro: How do olympiad medalists judge llms in competitive programming? *arXiv preprint arXiv:2506.11928*.
- Kun Zhou, Beichen Zhang, Zhipeng Chen, Xin Zhao, Jing Sha, Zhichao Sheng, Shijin Wang, Ji-Rong Wen, and 1 others. 2024. Jiu3.0: Efficiently improving mathematical reasoning by training small data synthesis models. *Advances in Neural Information Processing Systems*, 37:1854–1889.

# Appendices

<b>A More Details about MORPHOBENCH</b>	<b>12</b>
A.1 Details of Taxonomy . . . . .	12
A.2 More Proofs of Question Modification . . . . .	13
A.3 More Details of Difficulty Adjustment . . . . .	14
<b>B More Details of Evaluation Metric</b>	<b>17</b>
B.1 Basic Objects and Notation . . . . .	17
B.2 A Unified “Tool Review → Evidence → LLM Scoring” Framework . . . . .	17
B.3 Correctness (Accuracy) . . . . .	18
B.4 Reasoning Quality (Completeness / Logical Coherence) . . . . .	18
B.5 Hint Follow (Hint Alignment; Hint Subsets Only) . . . . .	19
B.6 Final Outputs (Four Metrics) . . . . .	19
<b>C More Evaluation Results</b>	<b>20</b>
C.1 Evaluations on Additional Models . . . . .	20
C.2 Diversity Analysis . . . . .	20
C.3 Visualized Examples of Difficulty Adjustments . . . . .	20
<b>D Broader Impact and Ethical Considerations</b>	<b>21</b>
D.1 Societal Impact . . . . .	21
D.2 Data Sourcing, Privacy, and Quality Assurance . . . . .	21
D.3 Potential Risks and Disclosure . . . . .	22

## A More Details about MORPHOBENCH

### A.1 Details of Taxonomy

We organize each sample into a three-level taxonomy. We denote each category by its full name followed by its abbreviation in parentheses. The leaf category of any sample is given by the tuple  $\langle L1, L2, L3 \rangle$ .

#### Level 1 (L1): Task Nature

- **Perception / Extraction (PERC)**. Low-level understanding and signal extraction from inputs, including recognition, reading diagrams/OCR, locating entities, and basic counting.
- **Retrieval / Matching (RETR)**. Locating or aligning information either provided in the prompt/evidence or drawn from external resources/commonsense; emphasis on correspondence and lookup.
- **Reasoning / Synthesis (RSYN)**. Multi-step deduction or constraint satisfaction that integrates pieces of evidence (e.g., flow/conservation rules, multi-hop logic chains) to reach a conclusion.

#### Level 2 (L2): Knowledge Closure

- **Closed (CLO)**. The answer is fully determined by the prompt and provided evidence; no outside knowledge is required.
- **Open (OPE)**. Solving requires external knowledge beyond what is given (e.g., background facts, domain conventions).
- **Hybrid (HYB)**. Primarily evidence-driven but benefits from a small amount of common or world knowledge (e.g., everyday conventions) to bridge gaps.

#### Level 3 (L3): Reasoning Primitive

- **Flow / Conservation (FLOW)**. Applying conservation or balance principles (e.g., circuit KCL/KVL, mass/energy balance, network flow).
- **Path / Reachability (PATH)**. Determining connectivity, routes, or shortest hops in graphs, mazes, or grids.

- **Chronology / Timeline (TIME).** Ordering events, aligning dates/eras/dynasties, or constructing consistent timelines.
- **Taxonomy / Hierarchy (TAXO).** Working with classification trees, phylogeny, or family hierarchies to place or infer relations.
- **Probability / Statistics / Estimation (PROB).** Handling uncertainty, intervals, likelihoods, sampling, or simple statistical summaries.
- **Arithmetic / Equation Solving (ARITH).** Performing numeric operations or solving algebraic equations/constraints.
- **Counting / Sets (COUNT).** Basic combinatorics, set relations/operations, and discrete enumerations.
- **Compare / Order (COMP).** Ranking or pairwise comparison tasks (greater/less, sorting by a criterion).
- **Geometry / Measurement (GEOM).** Reasoning about shapes, angles, areas/lengths, units/conversions, and geometric relations.
- **Topology / Matching / Pairing (MATCH).** Assignment, bijection/invariant-based pairing, or structure-preserving correspondence.
- **Textual Entailment / Logical Consistency (ENTAIL).** Checking whether statements are supported, contradicted, or mutually consistent with given text/evidence.

Each sample is labeled at all three levels; its leaf label is the concatenation of their abbreviations, formatted as L1-L2-L3 (e.g., RSYN-CLO-FLOW). When ambiguity arises, we prioritize (i) the dominant *task nature* (L1), then (ii) *knowledge closure* (L2), and finally (iii) the *primary reasoning primitive* (L3).

## A.2 More Proofs of Question Modification

This appendix formalizes why misleading modifications ( $\Delta I > 0$  in Eq. (5)) can enlarge the effective search space in the proof graph and thereby increase reasoning difficulty under a fixed compute budget. We use a minimal construction to make the argument explicit: a modification embeds additional incompressible information into the search graph by introducing spurious outgoing edges that do not lie on the unique goal path.

**Toy setting.** Consider an original proof graph whose goal-reaching structure is a single directed path

$$P = (v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_k),$$

which is the *unique* route from the start vertex  $v_0$  to the goal vertex  $v_k$ . A modification appends a hint  $\tau$  and induces a perturbed graph  $G_{Q'}$  by attaching dead-end (out-degree-one) edges while preserving  $P$  as the only goal path. Intuitively, these dead ends represent misleading branches that look plausible locally but cannot reach the verified terminal state.

Crucially, the terminal vertex  $v_k$  (representing the ground-truth answer) remains strictly invariant under these modifications; we are exclusively manipulating the complexity of the intermediate search space, not the fundamental semantic or mathematical terminus.

**Lemma 1** *Let the original search graph be a single directed path*

$$P = (v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_k),$$

*which is the unique route from the start vertex  $v_0$  to the goal vertex  $v_k$ . Embed an incompressible binary string  $\tau$  of length  $|\tau| = \Delta I$  bits into the graph by attaching  $m$  dead-end (out-degree-one) edges while preserving  $P$  as the only goal path. Then*

$$m \geq \Delta I - O(1).$$

**Proof 1** *Fix a universal prefix Turing machine  $U$ . Implicitly, we assume that the embedding is performed by a fixed computable map  $\mathcal{E} : \{0, 1\}^{\Delta I} \rightarrow \mathcal{G}$  that sends a bitstring  $\tau$  to a graph  $G = \mathcal{E}(\tau)$  obtained from  $P$  by attaching  $m$  dead-end edges while keeping  $P$  as the unique goal path. This ensures there is a fixed decoding procedure of constant size used in the complexity argument below.*

*For each vertex  $v_i$  on  $P$  let  $d_i$  be its out-degree in  $G$  and set  $s_i := d_i - 1 \geq 0$ ; thus*

$$m = \sum_i s_i$$

*is the total number of added edges. At vertex  $v_i$  there are exactly  $d_i = 1 + s_i$  possibilities for which outgoing edge continues along  $P$ , so the number of distinct graphs obtainable by choosing, at every vertex, which outgoing edge is the path-edge is at most*

$$\prod_i (1 + s_i).$$

Using the inequality  $1 + x \leq 2^x$  (valid for all  $x \geq 0$ ) we get

$$\prod_i (1 + s_i) \leq \prod_i 2^{s_i} = 2^{\sum_i s_i} = 2^m.$$

Hence there are at most  $2^m$  distinct graphs that can result from adding  $m$  dead-end edges to  $P$  while preserving  $P$  as the unique goal path.

Since  $\mathcal{E}$  is a fixed computable embedding, different inputs  $\tau$  must produce different output graphs; therefore the number of different  $\tau$  representable with  $m$  added edges is at most  $2^m$ . It follows that  $\tau$  has Kolmogorov complexity bounded by

$$K_U(\tau) \leq m + O(1),$$

where the  $O(1)$  term accounts for the fixed-size description of the decoding routine and the book-keeping needed to recover  $\tau$  from the index of the graph.

On the other hand, by the incompressibility assumption  $K_U(\tau) \geq \Delta I - O(1)$ . Combining the two bounds yields  $m \geq \Delta I - O(1)$ , as claimed.

### From dead ends to higher failure probability.

We next connect Lemma 1 to the budgeted failure event  $Fail(Q, B)$  used in the main text. The key additional ingredient is a *local indistinguishability* condition: when facing  $d_i$  outgoing edges at  $v_i$ , a model that cannot reliably discriminate the unique path-edge from spurious edges based on local information will allocate non-trivial probability mass to dead ends.

[Local indistinguishability on perturbed vertices]

For each vertex  $v_i$  on  $P$  with out-degree  $d_i = 1 + s_i$ , the policy selects the unique path-edge with probability at most  $\alpha_i \leq 1/(1 + s_i)$ . (Equality corresponds to uniform choice among outgoing edges.)

**Lemma 2** *Under Assumption A.2, the probability of reaching  $v_k$  from  $v_0$  without traversing any dead-end edge is upper bounded by*

$$\Pr[\text{reach } v_k \text{ in one try}] \leq \frac{1}{1 + m}.$$

**Proof 2** *Let  $\alpha_i$  be the probability of choosing the path-edge at  $v_i$ . By Assumption A.2,  $\alpha_i \leq 1/(1 + s_i)$ , hence*

$$\Pr[\text{one-try success}] = \prod_{i=0}^{k-1} \alpha_i \leq \prod_{i=0}^{k-1} \frac{1}{1 + s_i}.$$

Finally, since  $(1 + a)(1 + b) \geq 1 + a + b$  for all  $a, b \geq 0$ , by induction we obtain  $\prod_i (1 + s_i) \geq 1 + \sum_i s_i = 1 + m$ , which implies the stated bound.

[Budgeted failure increases with  $m$ ] Consider a budget  $B$  that allows at most  $T$  independent attempts (e.g., via restart after hitting a dead end), and let  $Fail(Q', B)$  denote failure to reach  $v_k$  within budget. Then

$$\Pr[Fail(Q', B)] \geq \left(1 - \frac{1}{1 + m}\right)^T.$$

In particular, larger  $m$  yields larger failure probability.

**Proof 3** *By Lemma 2, each attempt succeeds with probability at most  $p \leq 1/(1 + m)$ . With  $T$  independent attempts, success probability is at most  $1 - (1 - p)^T$ , hence failure probability is at least  $(1 - p)^T \geq (1 - \frac{1}{1+m})^T$ .*

**Linking back to the information gap.** Combining Corollary A.2 with Lemma 1, we see that a misleading modification with  $\Delta I > 0$  necessarily requires injecting at least  $\Omega(\Delta I)$  spurious edges in this construction, which in turn increases the failure probability under a fixed budget for any policy satisfying Assumption A.2. This supports the main-text claim that a positive information gap corresponds to an expansion of the effective search space and a higher likelihood of failure under limited compute.

### A.3 More Details of Difficulty Adjustment

**Agent Reasoning** Difficulty is a central factor in benchmark evaluation, yet it is often challenging to quantify due to its inherent subjectivity. Even when comparing problems within the same domain, it remains difficult to establish a rigorous partial order of difficulty; this challenge is further exacerbated when comparisons span across heterogeneous domains or disciplines. Conventional approaches typically resort to coarse-grained indicators, such as pass@N or weighted sums of chain-of-thought (CoT) lengths, which, while straightforward to compute, largely capture only superficial properties of model performance. Such measures fail to reflect more nuanced dimensions of reasoning, including the difficulty of exploration (the ability to branch into alternative solution paths), retrieval difficulty (the ability to identify relevant knowledge from prior context), and single-step reasoning difficulty (the precision of local logical inference).

To address these limitations, we propose the proof graph  $G$ . The graph serves as a modality that jointly encodes reasoning depth and exploration breadth, thereby offering a more fine-grained representation of problem-solving complexity. For a

given model  $M$ , we refer to its underlying knowledge system as axioms, while the intermediate conclusions derived throughout the reasoning process are termed lemmas. Formally, the proof graph is defined as  $G = (V, E)$ , where  $V$  denotes the set of lemmas and  $E$  represents the directed edges capturing inferential dependencies between them. The reasoning sub-process can thus be viewed as the progressive activation of new lemmas, based on both the initial axioms and previously established lemmas. We construct the proof graph  $G$  from an explicit reasoning trace. Given a step sequence  $S = (s_0, \dots, s_{n-1})$ , we convert it into a directed acyclic graph using lightweight heuristics together with an LLM-based parser. Each node corresponds to an intermediate statement, such as a given condition, a construction, an observation, a lemma, or the final goal. Each directed edge indicates that one statement supports or leads to another. Each node is linked back to the indices of the steps that gave rise to it, so the graph can be traced to the original text.

Building on the proof graph  $G$ , we further construct progressively revealable hints to enable controllable difficulty adjustment. Specifically, we generate a tiered hint set  $\mathcal{H} = \{H_1, \dots, H_T\}$  ordered from coarse guidance to increasingly specific cues, where each tier  $H_t$  contains several short hint statements. Hints are produced by selecting one or more salient lemmas from  $G$  and rewriting them into student-facing guidance. Early tiers mainly indicate what objects to focus on and what basic setup or auxiliary construction to introduce, while later tiers gradually disclose key intermediate relationships or important conclusions that can unlock the main solution path, yet still avoid stating the final answer. To ensure interpretability and auditability, each hint is aligned with evidence steps in the original reasoning trace, for example by recording the step indices that support the hint, so that the hint can be traced back to the specific trace statements it summarizes. For scalable generation and downstream use, we apply post-processing constraints to the hints, including length control for conciseness, mitigation of spoiler-like expressions that would directly reveal the final result, and consistent output formatting for reliable processing. Finally, difficulty is primarily adjusted by controlling the disclosure level of  $\mathcal{H}$ : low-disclosure settings provide only high-level orientation and leave more exploration space to the solver, whereas high-disclosure settings expose more pivotal intermediate structure

and thus reduce the remaining reasoning burden. In addition, under a hard setting, we can generate plausible but misleading hints that steer the solver toward an incorrect path, thereby increasing difficulty without changing the original question or answer. For each question, we analyze its unique reasoning trajectory and inject hints or perturbations targeting its exact cognitive bottlenecks, ensuring the benchmark dynamically tests logical robustness.

**Agent Recognition** In this stage, MORPHOBENCH adopts an image perturbation strategy based on the agent recognition of key visual information to increase task difficulty. We provide existing question–answer pairs to the agent and require it to identify and return the core visual elements within the corresponding images. Subsequently, as shown in Fig. 6, we perform text processing on these key pieces of information by obfuscating their textual descriptions in the question and the image, thereby introducing interference at the textual level.

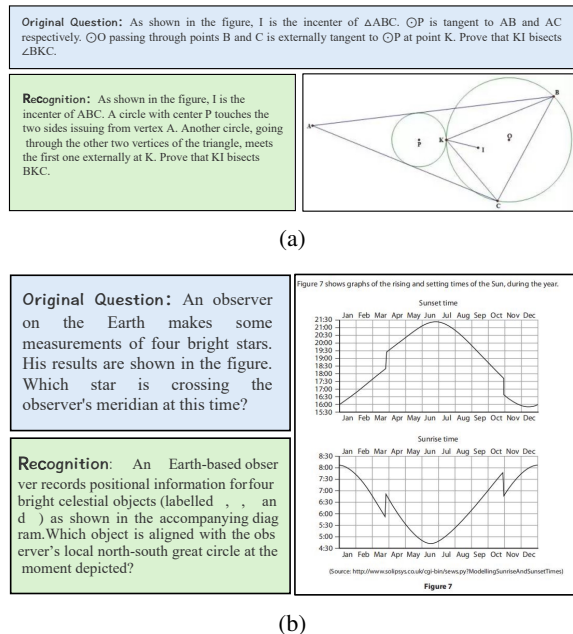


Figure 6: Example for Agent Recognition.

In the process, we use the agent's responses as the source of key visual information rather than relying on pre-defined annotations. The motivation is that allowing the model to indicate its most critical visual cues enables a more direct examination of its internal representations and attention mechanisms. By deliberately attacking specific visual priors that the model relies upon, these agent-driven perturbations are more targeted and challenging than exter-

**BlackBox:** You are presented with a sealed black-box electronic circuit. The internal structure is unknown, but you have access to the following information: 1. The black box has  $n$  input terminals labeled 1, 2, ...,  $n$ , and two output terminals o1 and o2. 2. Inside the box is a 5V DC power supply, along with a complex combination of resistors ( $1\Omega$ ), capacitors ( $1\text{mF}$ ), and inductors ( $10\mu\text{H}$ ). The exact configuration is hidden. 3. Each terminal pair (i-0) receives an applied sinusoidal AC signal. Each signal has a different amplitude and frequency, detailed below. 4. You are provided with the voltage waveform between output terminals o1 and o2 over time. Each input signal follows this format:  $V_{in\_i} = \text{SIN}(0, \text{Amplitude}, \text{Frequency})$  Example input signal table (units: V and Hz):

Terminal	Amplitude	Frequency
1	0.3	2000

Your task: Based on the given input signals and the observed output waveform, infer the likely internal circuit structure — including the types of components (resistors, capacitors, inductors) and their configuration (e.g., series or parallel). Below are several candidate circuit netlists (in SPICE format), please determine which single option best satisfies the specified requirements.

- (A) `*** V1 1 0 DC 5 R0 o1 o2 1 V_GND o2 0 0 R1_1 o1 a1 1 L1_1 a1 b1 10u C1_1 a1 b1 1m R1_2 b1 1 1 R1_3 1 o1 1 R2_1 o1 a2 1 L2_1 a2 b2 10u C2_1 b2 2 1m R2_2 2 o1 1.end***`
- (B) `*** V1 1 0 DC 5 R0 o1 o2 1 V_GND o2 0 0 R1_1 o1 a1 1 C1_1 a1 b1 1m L1_1 b1 c1 10u R1_2 c1 1 1 R1_3 1 o1 1.end***`
- (C) `*** V1 1 0 DC 5 R0 o1 o2 1 V_GND o2 0 0 R1_1 o1 a1 1 C1_1 a1 b1 1m R1_2 b1 1 1 L1_1 1 o1 10u R2_1 o1 a2 1 C2_1 a2 b2 1m L2_1 b2 2 10u R2_2 2 o1 1 R3_1 o1 a3 1 L3_1 a3 b3 10u C3_1 b3 3 1m R3_2 3 o1 1 R1_4 o1 a4 1 C1_4 a4 b4 1m L1_4 b4 4 10u R2_4 4 o1 1 R5_1 o1 a5 1 C5_1 a5 b5 1m L5_1 b5 5 10u R5_2 5 o1 1.end***`
- (D) `*** V1 1 0 DC 5 R0 o1 o2 1 V_GND o2 0 0 R1_1 o1 a1 1 C1_1 a1 b1 1m L1_1 b1 c1 10u R1_2 c1 1 1 C1_2 1 o1 1m R2_1 o1 a2 1 C2_1 a2 b2 1m L2_1 b2 2 10u R2_2 2 o1 1 R3_1 o1 a3 1 C3_1 a3 b3 1m L3_1 b3 3 10u R3_2 3 o1 1 C4_1 o1 a4 1m R4_1 a4 b4 1 L4_1 b4 4 10u R4_2 4 o1 1 R5_1 o1 a5 1 C5_1 a5 b5 1m L5_1 b5 c5 10u R5_2 c5 5 1 C5_2 5 o1 1m R6_1 o1 a6 1 C6_1 a6 b6 1m L6_1 b6 6 10u R6_2 6 o1 1.end***`
- (E) `*** V1 1 0 DC 5 R0 o1 o2 1 V_GND o2 0 0 R1_1 o1 a1 1 C1_1 a1 b1 1m R1_2 b1 c1 1 L1_1 c1 1 10u C1_2 o1 1 1m R2_1 o1 a2 1 C2_1 a2 b2 1m L2_1 b2 2 10u R2_2 2 o1 1 R3_1 o1 a3 1 L3_1 a3 b3 10u C3_1 b3 3 1m R3_2 3 o1 1.end***`
- (F) None of the above

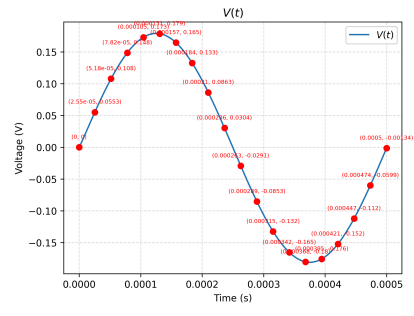


Figure 7: Example for Circuit Black-box Questions

nally imposed random noise, as they directly affect the features most relied upon. If a VLM continues to produce correct answers under such perturbations, it indicates robust fault tolerance and strong generalization. Conversely, a pronounced decline in performance reveals an excessive dependence on localized features and insufficient holistic understanding. Accordingly, this approach provides a more principled criterion for difficulty adjustment by assessing whether the model remains effective when its key visual dependencies are perturbed.

**Automatically Generated Questions** In MORPHOBENCH, automatically generated questions constitute a crucial component of the benchmark. There are two main challenges: how to ensure the logicity, professionalism, and verifiability of the generated questions, and how to automatically adjust their difficulty.

To address the first challenge, we introduce external simulation software to ensure the correctness of the automatically generated questions. For the second challenge, we adjust key parameters

of the automated question generation process to continuously increase both the complexity and the recognition difficulty of the tasks. Concretely, we design circuit black-box problems (Fig. 7) to evaluate reasoning ability and "spot the different one" tasks (Fig. 8) to assess visual recognition capacity. In circuit black-box problems, we leverage circuit simulators to validate outputs, producing waveform diagrams from output terminals to infer the underlying circuit structure. For difficulty adjustment, we control the number of external terminals exposed in the black-box. A larger number of terminals leads to higher difficulty. Although the internal structure is always theoretically solvable with the given component types, the complexity of the problem increases as the terminal count increases, making the reasoning task progressively more challenging.

The "spot the different one" tasks present grids of visually similar characters (for example, Latin letters or Chinese characters), with exactly one character differing from the others, and the model is required to identify the outlier. The difficulty here is modulated either by selecting character pairs

**Spot the different one:** You will be shown an image containing 6 rows and 11 columns of similar characters. Almost all positions contain the same character, except for one position where the character is different. Your task is to identify the location of this different character and provide its row (row) and column (col). Please output your answer strictly in the following format: row={row}, col={col} Notes: 1. Row numbering starts from 1 (the top row is row 1). 2. Column numbering starts from 1 (the leftmost column is col 1). 3. Do not output any additional text or explanation beyond the answer. Please output your answer strictly in the following format: row={row}, col={col}

b	b	b	b	b	b	b	b	b	b	b	b
b	b	b	b	b	b	b	b	b	b	b	b
b	b	b	b	b	b	b	b	b	b	b	b
b	b	b	b	b	b	b	b	d	b	b	b
b	b	b	b	b	b	b	b	b	b	b	b
b	b	b	b	b	b	b	b	b	b	b	b

Figure 8: Example for "Spot the Different One"

with greater visual similarity or by expanding the number of rows and columns. This setting probes the multimodal recognition capacity of VLMs in a controlled manner.

These mechanisms not only ensure the quality of automatically generated questions, but also support the design goal of MORPHOBENCH. The benchmark aims to realize self-evolving difficulty: by expanding terminal counts in circuits or grid size and similarity in visual tasks, the dataset naturally evolves toward harder problems. This allows the benchmark to continually stretch the boundaries of existing models, probing the upper limits of reasoning and multimodal understanding. By embedding evolutionary adjustment of difficulty into the generation pipeline, MORPHOBENCH establishes a dynamic and extensible evaluation platform, maintaining long-term relevance as models advance.

**Category Expansion** To ensure broad coverage across disciplines, we assign structured attributes to problems and organize them into a three-level tree: task type (perception, retrieval, reasoning), knowledge dependence (closed, open, hybrid), and fine-grained skill categories (e.g., arithmetic, geometry, flow). This hierarchical design avoids over-concentration in a single dimension and makes the benchmark more representative.

We iterate by setting per-leaf quotas and targeted collection for sparse leaves. This disciplined assignment and rebalance loop expands breadth while preserving difficulty structure, keeping benchmark diversity controllable over time.

## B More Details of Evaluation Metric

We present a unified evaluation formulation for all metrics:

$$\begin{aligned} \text{Score} &= \text{LLM}(p, a, r) = G_\phi(p, \text{Evidence}), \\ \text{Evidence} &= F(\Phi(p, a)). \end{aligned} \quad (11)$$

We use o3-mini as the judge LLM, and it is not among the evaluated models, to reduce potential evaluation bias (e.g., self-preference) in LLM-as-a-judge scoring.

### B.1 Basic Objects and Notation

**Benchmark dataset.** We evaluate on a benchmark dataset

$$\mathcal{D} = \{d_i\}_{i=1}^N, \quad d_i = (x_i, y_i, h_i), \quad (12)$$

where  $x_i$  denotes the multimodal input,  $y_i$  is the ground-truth answer, and  $h_i$  is an optional hint set. If  $d_i$  belongs to a hint-augmented subset (e.g., R\_Lite, R\_Complex), we define

$$h_i = \{H_{i,k}\}_{k=1}^{K_i}, \quad (13)$$

otherwise  $h_i = \emptyset$ .

**Model output.** Given input  $x_i$ , the evaluated model  $M$  outputs

$$a_i = M(x_i) = (\hat{y}_i, \rho_i), \quad (14)$$

where  $\hat{y}_i$  is the final answer and  $\rho_i$  is an observable reasoning trace (when available).

### B.2 A Unified "Tool Review $\rightarrow$ Evidence $\rightarrow$ LLM Scoring" Framework

For each evaluation aspect  $j \in \{\text{acc}, \text{R}, \text{H}\}$  (Correctness / Reasoning / Hint), we define: (i) a tool-based review operator  $\Phi_j$  producing a structured

review  $r_i^{(j)}$ , (ii) an evidence aggregation operator  $F_j$  mapping  $r_i^{(j)}$  to  $\text{Evidence}_i^{(j)}$ , and (iii) an LLM-based scoring mapping  $G_{j,\phi}$  (parameterized by judge model  $\phi$ ). Formally,

$$\text{Score}_i^{(j)} = \text{LLM}(p_j, a_i, r_i^{(j)}) = G_{j,\phi}\left(p_j, \underbrace{F_j(r_i^{(j)})}_{\text{Evidence}_i^{(j)}}\right),$$

$$r_i^{(j)} = \Phi_j(p_j, a_i, \dots), \quad (15)$$

where  $p_j$  is the expert prompt of aspect  $j$ , and “...” indicates that a stage may additionally use  $y_i$  (for correctness) or  $h_i$  (for hint-following).

### B.3 Correctness (Accuracy)

#### B.3.1 Tool Review (Correctness Judge)

The correctness toolchain outputs a structured review:

$$r_i^{\text{acc}} = \Phi_{\text{acc}}(p_{\text{acc}}, \hat{y}_i, y_i) = (\text{is\_correct}_i), \quad (16)$$

where  $\text{is\_correct}_i \in \{0, 1\}$ .

#### B.3.2 Metric Definition

Single-sample accuracy is defined as

$$\text{Acc}_i = \text{is\_correct}_i. \quad (17)$$

Dataset-level accuracy is

$$\text{Acc}(\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \text{Acc}_i. \quad (18)$$

### B.4 Reasoning Quality (Completeness / Logical Coherence)

Reasoning quality is evaluated by alternating two tools: *step decomposition* and *adjacent-step checking*.

#### B.4.1 Reasoning Step Decomposition (ReasoningBreakdownTool)

We decompose the reasoning trace  $\rho_i$  into an ordered step sequence:

$$s_{1:T_i}^{(i)} = \Phi_{\text{bd}}(p_{\text{bd}}, \rho_i) = (s_1^{(i)}, \dots, s_{T_i}^{(i)}). \quad (19)$$

#### B.4.2 Adjacent Step Review (Step-by-Step Check Tool)

For each adjacent pair  $(s_{t-1}^{(i)}, s_t^{(i)})$ , the step checker produces a local review:

$$e_t^{(i)} = \Phi_{\text{step}}(p_{\text{step}}, s_{t-1}^{(i)}, s_t^{(i)}), \quad t = 2, \dots, T_i. \quad (20)$$

The structured output is  $e_t^{(i)} = (a_t^{(i)}, g_t^{(i)}, c_t^{(i)}, m_t^{(i)}, r_t^{(i)}, \text{issues}_t^{(i)}, \text{comment}_t^{(i)})$ , where

- $a_t \in \{0, 1\}$ : is\_sound,
- $g_t \in \{0, 1\}$ : has\_logical\_gap,
- $c_t \in \{0, 1\}$ : has\_contradiction,
- $m_t \in \{0, 1\}$ : missing\_justification,
- $r_t \in \{0, 1, 2\}$ : local discrete rating,

and  $\text{issues}_t$  /  $\text{comment}_t$  summarize the detected problems and explanations.

We define the overall reasoning-stage review as

$$r_i^{\text{R}} = \Phi_{\text{R}}(p_{\text{R}}, a_i) = \{e_t^{(i)}\}_{t=2}^{T_i}, \quad (21)$$

$$\Phi_{\text{R}} := \Phi_{\text{step}} \circ \Phi_{\text{bd}}.$$

#### B.4.3 Evidence Aggregation and Final Scores

We aggregate stepwise reviews into evidence:

$$\text{Evidence}_i^{\text{R}} = F_{\text{R}}(r_i^{\text{R}}) = F_{\text{R}}(\{e_t^{(i)}\}_{t=2}^{T_i}). \quad (22)$$

Then the reasoning judge expert (summary LLM) outputs two scores:

$$(\text{Comp}_i, \text{Coh}_i) = G_{\text{R},\phi}(p_{\text{R}}, \text{Evidence}_i^{\text{R}}), \quad (23)$$

$$\text{Comp}_i, \text{Coh}_i \in [0, 100].$$

Here  $\text{Comp}_i$  measures whether the reasoning chain covers key steps required for solving the problem, while  $\text{Coh}_i$  measures logical self-consistency, supported by evidence such as contradiction detection, logical-gap statistics, and aggregated issue summaries.

Dataset-level aggregation is

$$\text{Comp}(\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \text{Comp}_i, \quad (24)$$

$$\text{Coh}(\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \text{Coh}_i.$$

#### B.4.4 Case Study: Diagnosing Reasoning Trajectories

To illustrate this diagnostic granularity, consider a complex problem requiring models to deduce a black-box circuit’s internal structure based on its output waveform.

##### Model A: Claude-Opus-4.6 (Question ID: 503)

- **Step 1:** Characterizes output waveform (peak, frequency). **Audit: 0/2 (Gap).** Claims lack derivation from visual input.

- **Step 2:** Identifies four frequency components. **Audit: 0/2 (Gap).** Introduces new quantities without justification.
- **Step 3:** Uses impedance analysis on Option D. **Audit: 2/2 (Sound).** Rigorous calculation based on established premises.
- **Step 4:** Verifies predicted values against data points. **Audit: 2/2 (Sound).**
- **Step 5:** Gives final conclusion. **Audit: 0/2 (Logical Leap).** Lacks a logical bridge to the final answer.

**Final Score:** Completeness 60/100, Coherence 60/100.

**Model B: Qwen3-VL-32B (Question ID: 462)**

- **Step 1:** Claims waveform main frequency is  $\sim 2000\text{Hz}$ . **Audit: 0/2 (Contradiction).** Incorrect terminal identification conflicts with premise.
- **Step 2:** Claims Option B values suit the audio band. **Audit: 0/2 (Gap).** Disconnected from Step 1; lacks justification.
- **Step 3:** Calculates 2000Hz gain. **Audit: 1/2 (Minor Issue).** Contains calculations but misses intermediate derivations.
- **Step 4:** Selects both B and D. **Audit: 0/2 (Contradiction).** Explicitly contradicts its own analysis in Step 3.

**Final Score:** Completeness 30/100, Coherence 40/100.

**Insight Revealed:** Our mechanism diagnostically isolates *where and how* the reasoning breaks down. Instead of merely assigning a binary score, it successfully recognized valid intermediate mathematical derivations (Steps 3–4) in Opus’s response, while accurately pinpointing the exact steps where the internal logic became inconsistent for Qwen. This demonstrates our framework’s capability to provide fine-grained, objective evaluation of intermediate reasoning trajectories.

## B.5 Hint Follow (Hint Alignment; Hint Subsets Only)

Let the hint-augmented subset be

$$\mathcal{D}^{\text{hint}} = \{d_i \in \mathcal{D} \mid h_i \neq \emptyset\}. \quad (25)$$

The hint stage reuses the step sequence  $\mathbf{s}_{1:T_i}^{(i)}$ .

## B.5.1 Stepwise Hint Alignment Review (HintCheckTool)

For each step  $s_t^{(i)}$ , the hint checker produces

$$u_t^{(i)} = \Phi_{\text{hint}}(p_{\text{hint}}; x_i, h_i, \mathbf{s}_{<t}^{(i)}, s_t^{(i)}). \quad (26)$$

We represent the structured output as  $u_t^{(i)} = (\alpha_t^{(i)}, \mathcal{K}_t^{(i)}, J_t^{(i)}, Q_t^{(i)}, \text{comment}_t^{(i)})$ , where

- $\alpha_t \in \{\text{aligned, deviated, neutral}\}$ : alignment status,
- $\mathcal{K}_t \subseteq \{1, \dots, K_i\}$ : relevant hints (1-based indices),
- $J_t \in \{\text{true, false, null}\}$ : whether a deviation is justified,
- $Q_t \in \{\text{strong, weak, none, null}\}$ : justification quality.

The overall hint-stage review is

$$r_i^{\text{H}} = \Phi_{\text{H}}(p_{\text{H}}, a_i, h_i) = \{u_t^{(i)}\}_{t=1}^{T_i}. \quad (27)$$

Notably,  $\Phi_{\text{hint}}$  evaluates *alignment with the provided hints* rather than the factual correctness or helpfulness of the hints.

## B.5.2 Evidence Aggregation and Hint Alignment Score

We aggregate hint reviews into evidence:

$$\text{Evidence}_i^{\text{H}} = F_{\text{H}}(r_i^{\text{H}}). \quad (28)$$

The hint judge (summary LLM) outputs the final alignment score:

$$\begin{aligned} \text{HAlign}_i &= G_{\text{H},\phi}(p_{\text{H}}, \text{Evidence}_i^{\text{H}}), \\ \text{HAlign}_i &\in [0, 100]. \end{aligned} \quad (29)$$

Dataset-level aggregation (over  $\mathcal{D}^{\text{hint}}$ ) is

$$\text{HAlign}(\mathcal{D}^{\text{hint}}) = \frac{1}{|\mathcal{D}^{\text{hint}}|} \sum_{d_i \in \mathcal{D}^{\text{hint}}} \text{HAlign}_i. \quad (30)$$

## B.6 Final Outputs (Four Metrics)

For each sample  $d_i$ , the system outputs

$$\mathbf{m}_i = (\text{Acc}_i, \text{Comp}_i, \text{Coh}_i, \text{HAlign}_i), \quad (31)$$

where  $\text{HAlign}_i$  is defined only when  $h_i \neq \emptyset$ ; otherwise it is marked as NA and excluded from the dataset-level mean.

Model	Version
Claude-Opus-4.6	claude-opus-4.6
Gemini-2.5-flash	gemini-2.5-flash-thinking
Gemini-2.5-pro	gemini-2.5-pro-thinking
Gemini-3-pro	gemini-3-pro-preview-thinking
GPT-5.1	gpt-5.1-2025-11-13
Grok-4	grok-4-0709
o3	o3
o4-mini	o4-mini
Qwen3-VL-32B	qwen3-vl-32b

Table 6: **Model versions.** We report the exact API model identifiers used for inference. For all models, we enable the highest available thinking/reasoning budget in our inference configuration.

## C More Evaluation Results

### C.1 Evaluations on Additional Models

To further validate the generalizability of MORPHOBENCH across diverse model architectures and capacities, we conduct additional evaluations on the leading open-source multimodal model, Qwen3-VL-32B, and the frontier closed-source model, Claude-Opus-4.6.

As shown in Table 7, both models strictly follow the expected difficulty gradients on MORPHOR (Lite > v0 > Complex) and exhibit consistent accuracy drops under multimodal perturbations (MORPHO-P). Notably, Claude-Opus-4.6 achieves highly competitive results, performing on par with GPT-5.1.

Furthermore, Table 8 reports their diagnostic metrics. The significant drop in Claude-Opus-4.6’s Hint Alignment under misleading hints explicitly demonstrates the framework’s effectiveness in diagnosing over-reliance on erroneous external guidance.

### C.2 Diversity Analysis

The classification results in Fig. 9 show that reasoning tasks are predominant, while all three top-level categories remain well represented. This ensures the benchmark includes both problems solvable through prompt-only evidence and those requiring external knowledge. At the leaf level, the dataset spans a diverse spectrum—from combinatorics and geometry to timeline reasoning and logical entailment.

Following our expansion and rebalancing operations, both hierarchical evenness and entropy show notable improvement, with leaf coverage reaching approximately 60% of possible taxonomy paths. This validates both the taxonomy’s

expressiveness and the effectiveness of our balancing policy. For future iterations, we will prioritize problems with Open/Hybrid knowledge closure, retrieval-anchored items, and perception tasks requiring open knowledge. This strategy will help smooth the long-tail distribution while maintaining strong reasoning requirements.

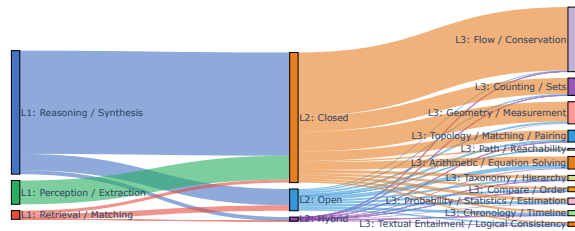


Figure 9: Diversity analysis of MORPHOBENCH.

### C.3 Visualized Examples of Difficulty Adjustments

To further demonstrate the adaptability and generality of our benchmark, we present representative examples under the two proposed difficulty adjustment paradigms: agent recognition and agent reasoning.

In the agent recognition adjustment, difficulty is modulated through textual fuzzification guided by visual grounding. Specifically, the model first identifies the key visual elements that support the correct answer, such as symbols, numbers, geometric labels, or local regions, and then weakens or replaces the corresponding textual expressions in the question with qualitative descriptions. This process preserves solvability while increasing ambiguity, compelling models to rely more on visual perception rather than direct text–answer mapping.

In contrast, the agent reasoning adjustment focuses on the cognitive chain of inference. By analyzing the essential theorems and intermediate steps within the reasoning process, we strategically

Model	MORPHO-R (Lite)	MORPHO-R (v0)	MORPHO-R (Complex)	MORPHO-P (v0)	MORPHO-P (Perturbed)
Qwen3-VL-32B	26.61 (+2.80)	23.81	19.18 (-4.63)	25.89	22.27 (-3.62)
Claude-Opus-4.6	55.84 (+3.97)	51.87	45.83 (-6.04)	48.32	41.60 (-6.72)

Table 7: **Answer correctness of additional models on MORPHOBENCH.** We report accuracy (%); parentheses denote absolute changes relative to the corresponding v0 baseline.

Model	Reasoning Quality (v0) (Comp. / Coh.)	Hint Alignment (Lite → Complex)
Qwen3-VL-32B	36.36 / 37.51	60.99 → 50.85 (-10.14)
Claude-Opus-4.6	71.86 / 74.72	89.20 → 69.11 (-20.09)

Table 8: **Reasoning Quality and Hint Alignment for additional models.** Reasoning Quality is reported on MORPHO-R(v0) as Completeness and Logical Coherence. Hint Alignment shows the performance degradation when shifting from supportive hints (Lite) to misleading perturbations (Complex).

introduce irrelevant or partially related hints to interfere with the model’s logical flow. These additions encourage the model to distinguish between critical and misleading information, thereby evaluating its structured reasoning ability under uncertainty.

In the following examples (Figs. 10 to 13), we visualize several representative instances to illustrate these two adjustment modes. These multi-disciplinary examples collectively demonstrate how our benchmark dynamically reconfigures question difficulty through two complementary mechanisms, enabling more fine-grained and interpretable evaluation of multimodal reasoning capabilities.

## D Broader Impact and Ethical Considerations

### D.1 Societal Impact

We propose MORPHOBENCH, a high-quality multidisciplinary reasoning benchmark that provides a robust standard for evaluating the reasoning capabilities of state-of-the-art models. It has no direct negative societal impacts. However, we must be cautious about the potential misuse of MORPHOBENCH and the models it evaluates by malicious actors.

### D.2 Data Sourcing, Privacy, and Quality Assurance

We collected data from two main sources: the Art of Problem Solving (AoPS) website, and Chinese Mathematics/Physics Olympiad Training Problems. Both sources already provide complete solutions or official answers. We obtained permission from

the respective data providers before using their materials for research purposes. These resources were chosen because they are authoritative, widely used in scientific training, and highly relevant to the complex problem domain addressed in our study.

To guarantee determinism and verifiability within our difficulty adaptation pipeline, we integrated a rigorous expert review phase. Domain experts meticulously reviewed the automatically generated question variants, reaching a strong consensus that the perturbed questions are logically sound, unambiguous, and perfectly compatible with the original ground-truth answers. The human checkers responsible for this verification were compensated at approximately USD 1200 per month, which is well above the local minimum wage and aligns with the standard compensation for domain experts in their region. The experts were recruited through university networks. Detailed instructions were provided to all domain experts to standardize the verification of problem validity and answer compatibility. Given that the human involvement was strictly limited to verifying the mathematical and logical correctness of standard problem-solving datasets, this data quality assurance process was deemed exempt from formal Ethics Review Board approval.

We relied on widely used benchmark datasets that have long served as standard resources in the research community. These datasets are curated by reputable organizations. To the best of our knowledge, they contain no personally identifiable information or inappropriate material; any elements with potential sensitivity have already been anonymized or excluded.

Original Question: Which element has these spectral lines?
Recognition : Which element corresponds to the emission pattern shown?
Reasoning: Which element has these spectral lines? <ul style="list-style-type: none"> <li>• The spectral lines of an element are directly related to its atomic number. Try to match the spectral lines with the atomic number of the elements.</li> <li>• Elements in the same group of the periodic table have similar spectral lines. Look for an element in the same group as Calcium but not Calcium itself."</li> </ul>

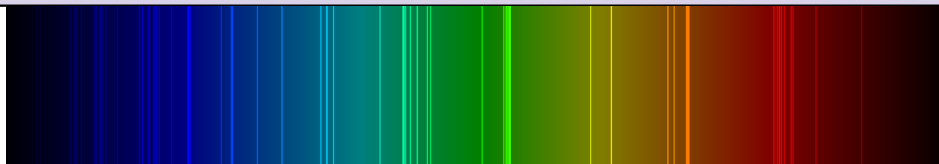


Figure 10: Multi-disciplinary examples under agent recognition and reasoning.

### D.3 Potential Risks and Disclosure

- **Exam/contest leakage risk:** Some problems originate from past or mock exams and could potentially be misused for unauthorized “drill” purposes rather than their intended scientific evaluation.
- **AI-assisted writing disclosure:** An AI assistant was employed solely for grammar correction and minor stylistic improvements of the manuscript. It was not involved in the scientific design, analysis, answer annotation, or technical development of the research. All data generation and labeling were strictly verified by human experts.

**Original Question:** "For the three heritable features, Alfa, Baker, and Charlie, pedigree analysis was performed on pedigree A, pedigree B, and pedigree C, respectively, and the results in Figure 1 were obtained. Indicate whether each of the following statements is true or false.

\begin{description}  
 \item[1.] An analysis of pedigree A suggests that the inheritance pattern of characteristic Alfa could be due to a dominant allele.  
 \item[2.] An analysis of pedigree C suggests that the inheritance of the characteristic Charlie could be due to a dominant allele.  
 \end{description}  
 A subsequent detailed analysis revealed that all of the inheritance patterns of Alfa, Baker, and Charlie were due to recessive alleles on the autosome.  
 \begin{description}  
 \item[3.] B1 and B3 of family B are definitely carriers.  
 \item[4.] C1 and C3 of family C are definitely carriers.  
 \end{description}"

**Recognition :** "For three observable characters—Alfa, Baker, and Charlie—family trees were compiled in the first, second, and third diagrams of Figure 1.

Indicate whether each of the following statements is true or false.

1. Examination of the first diagram suggests that the form taken by Alfa might stem from an allele that shows its effect with a single copy.
  2. Examination of the third diagram suggests that the form taken by Charlie might stem from an allele that shows its effect with a single copy.
- Subsequent study showed that all three traits are in fact produced by recessive autosomal alleles.
3. Individuals B1 and B3 in the second diagram are definitely heterozygous for the trait allele.
  4. Individuals C1 and C3 in the third diagram are definitely heterozygous for the trait allele."

**Reasoning:** "For the three heritable features, Alfa, Baker, and Charlie, pedigree analysis was performed on pedigree A, pedigree B, and pedigree C, respectively, and the results in Figure 1 were obtained. Indicate whether each of the following statements is true or false.

\begin{description}  
 \item[1.] An analysis of pedigree A suggests that the inheritance pattern of characteristic Alfa could be due to a dominant allele.  
 \item[2.] An analysis of pedigree C suggests that the inheritance of the characteristic Charlie could be due to a dominant allele.  
 \end{description}  
 A subsequent detailed analysis revealed that all of the inheritance patterns of Alfa, Baker, and Charlie were due to recessive alleles on the autosome.  
 \begin{description}  
 \item[3.] B1 and B3 of family B are definitely carriers.  
 \item[4.] C1 and C3 of family C are definitely carriers.  
 \end{description}"

- When analyzing pedigrees, a dominant allele is often indicated when the trait appears in every generation. Therefore, for pedigree A, the characteristic Alfa is likely due to a dominant allele.
- In pedigree C, if the characteristic Charlie skips a generation, it is likely due to a recessive allele. Hence, the statement is false.

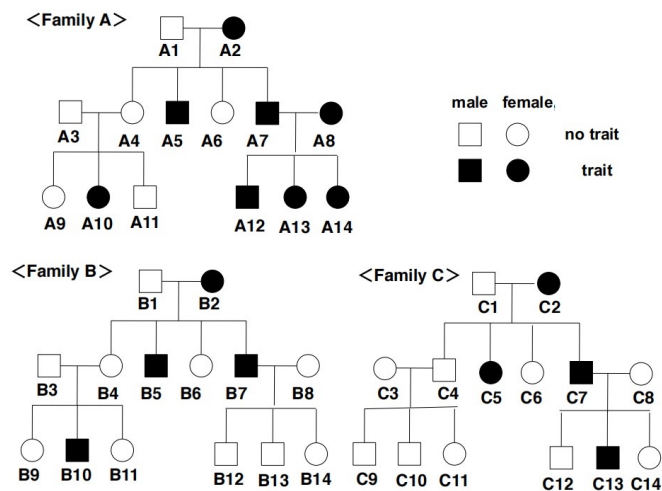


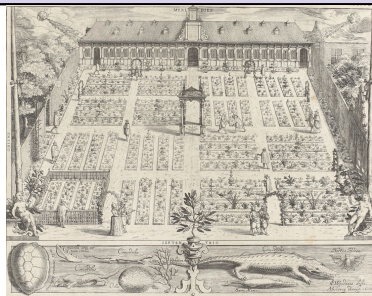
Figure 11: Multi-disciplinary examples under agent recognition and reasoning.

**Original Question:** Where is this botanical garden located?

**Recognition:** Where is this university-affiliated green space located?

**Reasoning:** Where is this botanical garden located?

- The botanical garden you're looking for is located in a city known for its iconic Eiffel Tower.
- This city is also famous for its Louvre Museum, which houses the Mona Lisa.



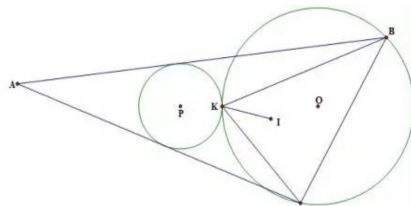
(a)

**Original Question:** As shown in the figure,  $I$  is the incenter of  $\triangle ABC$ .  $\odot P$  is tangent to  $AB$  and  $AC$  respectively.  $\odot O$  passing through points  $B$  and  $C$  is externally tangent to  $\odot P$  at point  $K$ . Prove that  $KI$  bisects  $\angle BKC$ .

**Recognition:** As shown in the figure,  $I$  is the incenter of  $\triangle ABC$ . A circle with center  $P$  touches the two sides issuing from vertex  $A$ . Another circle, going through the other two vertices of the triangle, meets the first one externally at  $K$ . Prove that  $KI$  bisects  $\angle BKC$ .

**Reasoning:** As shown in the figure,  $I$  is the incenter of  $\triangle ABC$ .  $\odot P$  is tangent to  $AB$  and  $AC$  respectively.  $\odot O$  passing through points  $B$  and  $C$  is externally tangent to  $\odot P$  at point  $K$ . Prove that  $KI$  bisects  $\angle BKC$ .

- Consider using the Pythagorean theorem to find the lengths of the sides of the triangle. This will help you determine the angles.
- Assume that the circles are not homothetic about point  $K$ . This will simplify the problem and lead you to the correct answer.



(b)

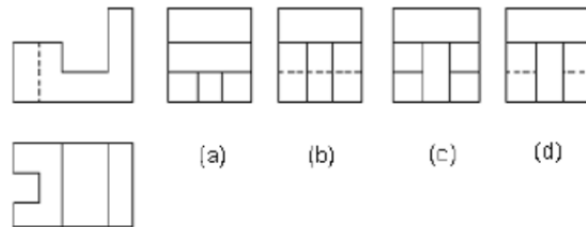
Figure 12: Multi-disciplinary examples under agent recognition and reasoning.

**Original Question:** Given the front view and the top view of an object, please choose the correct left view from options A, B, C, and D. Answer with the minimal form only. No other content.

**Recognition :** Given the two given orthographic projections of a solid, determine which of the labelled sketches (A, B, C, D) depicts the view from the remaining orthogonal direction. Answer with the minimal form only. No other content.

**Reasoning:** Given the front view and the top view of an object, please choose the correct left view from options A, B, C, and D. Answer with the minimal form only. No other content.

- Remember that the left view is always a mirror image of the front view. So, look for the option that is a mirror image of the front view.
- The top view and the left view are usually identical. So, the correct left view should look exactly like the top view.



(a)

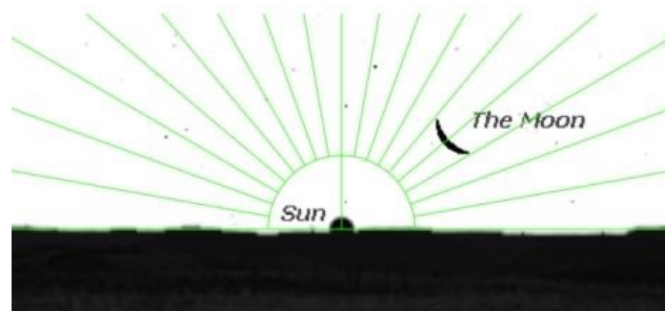
**Original Question:** "An astro-photographer has taken the photo of the moon close to a new moon day shown below right before the sunset on December 21 (Winter Solstice) in a wide open area. (a) In which hemisphere (Northern or Southern) is the photographer located?"

**Recognition :** An astro-photographer captured an image of a razor-thin lunar crescent just as the day's light fades around the December solstice in a wide, unobstructed location.

(a) In which hemisphere (Northern or Southern) is the photographer located?

**Reasoning:** In which hemisphere (Northern or Southern) is the photographer located?

- Consider the date of the photograph. December 21 is the Winter Solstice, which is the shortest day of the year in the Northern Hemisphere. Therefore, the photographer must be in the Northern Hemisphere.
- Remember that the moon is closer to the horizon during a new moon. This would suggest that the photographer is in the Northern Hemisphere, where the moon appears lower in the sky during winter."



(b)

Figure 13: Multi-disciplinary examples under agent recognition and reasoning.