Beyond Ranking: Fine-Grained Diagnostics and Self-Improvement for MLLMs

Mingze Xu[♢*], **Zijing Zhao**[♢], **Qiming Peng**[♠], **Houwen Peng**[♠], **Han Hu**[♠], **Zhanhui Kang**[♠], **Yuxing Han**[♢†],
[♢]Shenzhen International Graduate School, Tsinghua University [♠]Tencent Hunyuan

Abstract

While Multimodal Large Language Models (MLLMs) are advancing rapidly, accurately evaluating their capabilities remains challenging. Current paradigms primarily rely on holistic scoring and static leaderboards, which fail to disentangle fine-grained competencies. Specifically, they suffer from "Outcome Bias" by validating only final answers and ignoring intermediate reasoning. To address these limitations, we introduce **ATOM** (**A**na**T**omy **O**f **M**LLM), a novel MLLM-as-a-judge framework designed to shift the focus from ranking to fine-grained diagnosis. ATOM decomposes complex reasoning into atomic criteria anchored in visual elements, enforcing verification against explicit visual facts. Validated on a newly constructed benchmark with rigorous human rankings, ATOM[1] achieves state-of-the-art accuracy, surpassing the strongest baseline by up to **7.92%**. Moving beyond ranking, ATOM bridges the gap between assessment and alignment: by pinpointing atomic-level failures, it establishes a closed-loop mechanism for targeted self-correction. This approach enables models to identify and rectify errors autonomously, successfully resolving up to **39.95%** of previously failed queries without human intervention.

1 Introduction

The rapid evolution of Multimodal Large Language Models (MLLMs) has revolutionized tasks ranging from diverse visual understanding to complex reasoning (Kuang et al., 2025; Yang et al., 2025; Qin et al., 2025). As these models become increasingly sophisticated, accurately assessing and understanding their capabilities has become a key challenge (Fu et al., 2023; Ge et al., 2025; Zhang et al., 2025b). Human-based evaluation platforms like Chatbot Arena (Chiang et al., 2024) have long been regarded as the gold standard, providing robust leaderboards based on massive crowdsourced votes. However, this labor-intensive paradigm faces severe scalability bottlenecks, struggling to support the evaluation of thousands of models across massive fine-grained tasks at a manageable cost (Yin et al., 2025; Wang et al., 2025). Consequently, the community has shifted towards the "MLLM-as-a-Judge" paradigm (Chen et al., 2024), leveraging advanced models as surrogates to efficiently approximate rankings.

Nevertheless, current automated benchmarks predominantly adhere to a **paradigm of macro-level ranking via holistic scoring** (Zhao et al., 2025; Yin et al., 2025). They typically aggregate diverse queries into coarse metrics, offering only overall scores or win-rates. This aggregation obscures the correlation between specific task types and model performance, acting as an opaque "black box" that leaves two critical questions unanswered: 1) What specific fine-grained capabilities are actually probed by the evaluation samples and mastered by the model? 2) Is a failure due to a fundamental lack of capability, or can we guide the model to rectify its errors without retraining?

Specifically, current benchmarks fail to address these uncertainties due to three fundamental limitations rooted in their reliance on **coarse-grained approximations**. **First, the "Outcome Bias" in holistic scoring.** Traditional judges treat complex reasoning as a monolithic whole (Chen et al., 2024), validating only final answers. This lack of *process granularity* rewards spurious guesses derived from textual priors despite flawed reasoning (false positives), or penalizes valid reasoning chains that yield rejected answers due to trivial discrepancies (false negatives), failing to distinguish true capability from luck. **Second, the ambiguity of macroscopic taxonomies.** Classifying performance by

[*]Work done during internship at Tencent Hunyuan, Contact: <xumz24@mails.tsinghua.edu.cn>
[†]Yuxing Han is the corresponding author.
[1]Our resources will be available at: https://github.com/MLLMs-project/ATOM.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 20347–20377
July 2-7, 2026

coarse domain labels (Yue et al., 2024; Ying et al., 2024) (e.g., "Math") rather than specific cognitive operations obscures *capability granularity*. This superficial grouping often misattributes failures, confusing atomic visual blindness with high-level reasoning deficits, thereby distorting the capability profile. **Third, the "Black Box" nature of scalar metrics.** Aggregating diverse errors into a single score compresses rich diagnostic signals into a flat number. This loss of *feedback granularity* (Wang et al., 2025) severs the link to optimization; without identifying specific deficits, simple rankings offer no actionable guidance for targeted self-improvement.

To address these limitations, we introduce **ATOM** (**A**na**T**omy **O**f **M**LLM), a novel fine-grained diagnostic framework that shifts the evaluation paradigm from holistic ranking to **structural process verification**. Unlike traditional judges that treat reasoning as a black box, ATOM mitigates "Outcome Bias" through a Cognitive Replay mechanism with Visual Anchoring. By projecting the judge model as an "Ideal Solver" prior to evaluation, ATOM decomposes complex queries into verifiable Atomic Criteria explicitly grounded in visual regions. This ensures that evaluation focuses on the validity of the reasoning process rather than merely the final text match, significantly reducing hallucinations in the judging phase. Furthermore, instead of forcing models into static, coarse-grained domains, we propose a Dynamic Capability Tree construction method. This data-driven approach evolves a hierarchical taxonomy from the atomic criteria themselves, enabling precise attribution of failures to specific cognitive deficits, ranging from low-level perception to high-level logic, thereby resolving the ambiguity of macroscopic taxonomies.

Moving beyond static ranking, ATOM establishes a practical and verifiable workflow for model improvement. Our experiments demonstrate that atom-guided feedback enables **autonomous self-correction** by directing attention to neglected visual evidence, proving that many failures stem from attention drift rather than a fundamental lack of capability. This enables models to achieve performance gains of up to 26.85% without parameter updates. We validated ATOM across 18 open- and closed-source models, achieving state-of-the-art alignment. To this end, we constructed a multimodal benchmark featuring sample-level human ranking annotations for multiple model responses. Applying structural analysis to this collected data,

Figure 1: Distribution of Fine-Grained Capabilities. The taxonomy is dynamically evolved from our collected data, mapping specific atomic skills to high-level dimensions via our structural analysis framework.

we dynamically evolved the fine-grained taxonomy shown in Figure 1. This depth not only aligns evaluation closely with human judgment but also enables "model fingerprinting," distinguishing different model architectures based on their unique capability profiles. Our contributions are summarized as follows:

- **Beyond Ranking: The ATOM Framework.** We propose a paradigm shift from holistic scoring to fine-grained process verification. By decomposing reasoning into visually-anchored atomic criteria, ATOM mitigates "Outcome Bias" and successfully distinguishes true cognitive competencies from spurious guesses.
- **Dynamic Taxonomy & Benchmark.** We construct a rigorous benchmark and a data-driven Dynamic Capability Tree. This bottom-up approach resolves the ambiguity of macroscopic domains by evolving capability definitions directly from atomic errors, enabling precise attribution of deficits.
- **Closed-Loop Self-Improvement.** We resolve the "Black Box" nature of scalar metrics by establishing a verifiable feedback loop. Our diagnostic traces serve as actionable instructions, guiding models to autonomously rectify errors and boosting performance by up to 26.85% without retraining.

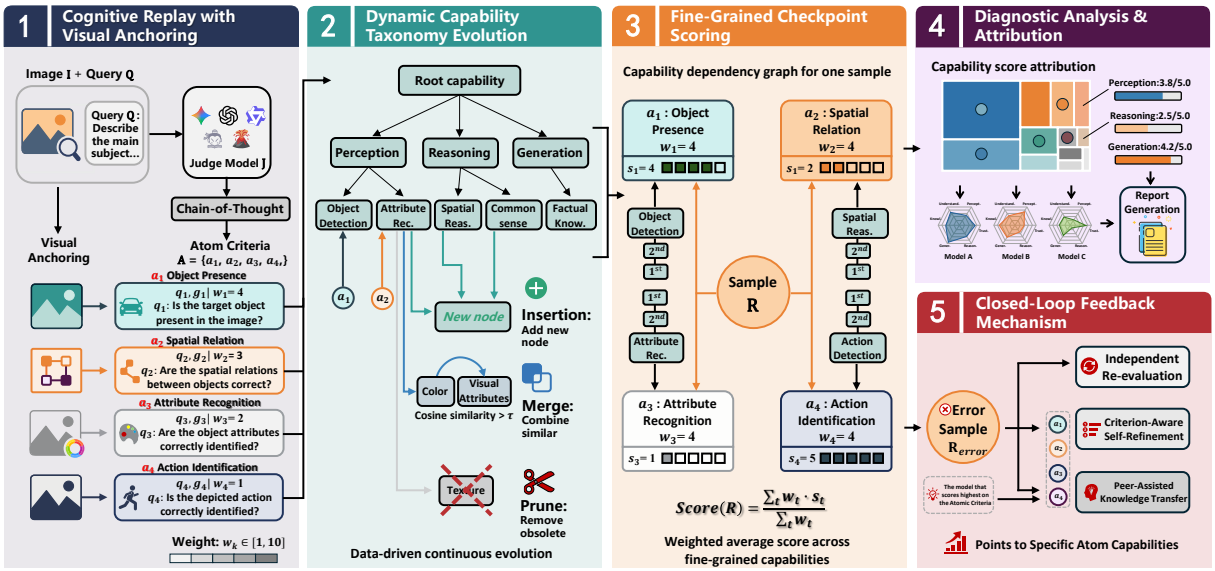


Figure 2: The **ATOM** Evaluation Pipeline. Starting with an image and query, the Judge Model (1) generates visually anchored Atomic Criteria, which are (2) mapped to a dynamically evolved capability tree. Models are then (3) scored at the checkpoint level to (4) produce fine-grained diagnostic reports, enabling (5) targeted self-correction through a closed-loop feedback mechanism.

2 Problem Formulation

Formally, let $\mathcal{D} = \{(I_i, Q_i)\}_{i=1}^{N}$ denote a multimodal dataset with visual input I_i and textual query Q_i. For a candidate MLLM $M_k \in \mathcal{M}$, we evaluate its response $R_{i,k} = M_k(I_i, Q_i)$. Unlike holistic evaluations that map $(I_i, Q_i, R_{i,k})$ to a single scalar, we propose a fine-grained diagnostic framework.

We employ a Judge Model $\mathcal{J}$ to decompose the reasoning process into a set of *Atomic Criteria* $\mathcal{A}_i = \{a_{i,t}\}_{t=1}^{T_i}$. Each criterion is defined as a tuple $a_{i,t} = \langle q_t, g_t, w_t \rangle$, representing a visual-anchored sub-question, local ground truth, and importance weight, respectively. To facilitate structural analysis, we define a Dynamic Capability Tree $\mathcal{T}$ and a mapping function $\phi : a_{i,t} \to l$ that projects each criterion to a specific capability node $l \in \mathcal{T}$.

The evaluation yields a weighted sample-level score $S_{i,k} = \frac{\sum_t w_t \cdot s_{i,k,t}}{\sum_t w_t}$, where $s_{i,k,t} \in \{1, \ldots, 5\}$ is the correctness score for criterion $a_{i,t}$. Finally, we derive diagnostic feedback $\mathcal{F}_{i,k}$ from low-scoring criteria to guide the model in generating a refined response $R'_{i,k} = M_k(I_i, Q_i, \mathcal{F}_{i,k})$.

3 Methodology

We propose a **Fine-Grained Multimodal Diagnostic Framework** that transitions from holistic ranking to structural diagnosis. As shown in Figure 2, the pipeline operates in five stages: (1) **Cognitive Replay** to extract grounded *Atomic Criteria*; (2) **Dynamic Taxonomy Construction** to evolve an adaptive capability tree; (3) **Checkpoint Scoring** to evaluate models against specific nodes; (4) **Diagnostic Attribution** to synthesize qualitative reports; and (5) a **Closed-Loop Feedback** mechanism for targeted improvement.

3.1 Cognitive Replay with Visual Anchoring

We first employ *Cognitive Replay*, where the Judge Model $\mathcal{J}$ generates a high-quality reference chain of thought ($\mathcal{Z}_i^*$) explicitly grounded in visual regions. Crucially, to mitigate "Outcome Bias," we decompose $\mathcal{Z}_i^*$ into discrete *Atomic Criteria* $\mathcal{A}_i$ for fine-grained verification. Each criterion $a_{i,t} = \langle q_t, g_t, w_t \rangle$ comprises a sub-question q_t targeting a specific reasoning step, a local ground truth g_t, and an importance weight $w_t \in [1, 10]$ reflecting the step's logical contribution. The extraction process is defined as:

$$\mathcal{A}_i = \{\langle q_t, g_t, w_t \rangle\}_{t=1}^{T_i} = \mathcal{J}_{\text{extract}}(\mathcal{Z}_i^*, I_i, Q_i) \quad (1)$$

This factorization transforms unstructured grading into a structured protocol, yielding an augmented dataset $\tilde{\mathcal{D}} = \{(I_i, Q_i, \mathcal{A}_i)\}_{i=1}^{N}$ of verifiable checkpoints.

3.2 Dynamic Capability Taxonomy Construction

Unlike static taxonomies, we construct a dataset-specific capability tree, $\mathcal{T}_{cap}$, to ensure evaluation

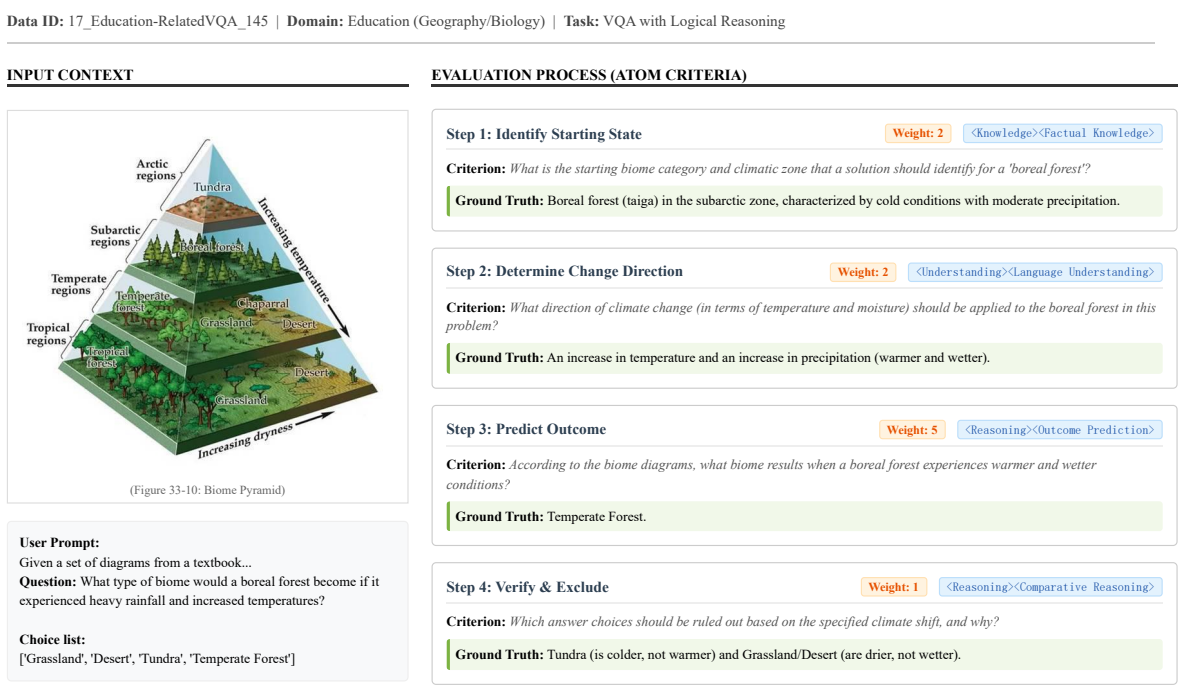


Figure 3: **An Illustrative Example of the ATOM Framework.** Given an input context (left), our framework decomposes the multimodal reasoning process into a structured sequence of **Atom Criteria** (right). Each atom is assigned: (1) a **Capability Tag** (e.g., <Understanding>) for taxonomy mapping, (2) a **Weight** reflecting its contribution to the total score, and (3) a precise **Ground Truth** for step-by-step verification. This granularity allows us to pinpoint exactly *where* and *why* a model fails, unlike traditional evaluations that only reveal if it fails.

metrics align with the input distribution while filtering out irrelevant capabilities. This data-oriented approach captures nuanced capabilities often overlooked by generic frameworks. The construction proceeds in four phases:

Initialization and Candidate Proposal To ensure high-level interpretability and prevent incoherent clustering, we enforce a top-down constraint by initializing $\mathcal{T}_{cap}$ with five fixed anchors: *Perception*, *Reasoning*, *Understanding*, *Trustworthiness*, and *Generation*. Using this skeleton, we employ a bottom-up induction on the augmented dataset $\tilde{\mathcal{D}}$. The Judge Model analyzes each atomic criterion to propose fine-grained capability requirements, generating a pool of raw candidates.

Merging and Pruning We refine the global candidate pool through an iterative process. The Judge Model clusters semantically identical nodes, prunes those with insufficient support or vague definitions, and hierarchically attaches surviving leaves to the appropriate root anchors. This step transforms raw proposals into a compact, optimized tree structure $\mathcal{T}^*_{cap}$.

Atom-to-Capability Mapping With the taxonomy fixed, we perform a definitive mapping to assign every atomic criterion to a specific leaf node in $\mathcal{T}^*_{cap}$. Unlike the proposal phase, this is formulated as a classification task over the pre-defined set. This verification step ensures precision, enabling us to aggregate atom-level results into capability-level scores for subsequent diagnostic analysis. An illustrative example is provided in Figure 3.

3.3 Fine-Grained Checkpoint Scoring

We instantiate the evaluation by directing the Judge model to map each extracted atomic criterion to a specific leaf node in the capability tree $\mathcal{T}$. This transforms the unstructured evaluation into a sequence of measurable capability checkpoints.

Subsequently, we perform a fine-grained verification where the Judge evaluates the candidate response $R_{i,k}$ against each atomic criterion $a_{i,t} = \langle q_t, g_t, w_t \rangle$. Instead of a holistic assessment, the Judge assigns a correctness score $s_{i,k,t} \in \{1, \ldots, 5\}$ by verifying whether the model's reasoning for the specific sub-question q_t aligns with the local ground truth g_t. This isolation prevents the "Outcome Bias," ensuring that a correct final answer does not mask intermediate reasoning errors. Finally, the sample-level quality score $S_{i,k}$ is

Table 1: Pairwise ranking accuracy (%) against human ground truth. We employ six diverse models as backbones for ATOM to evaluate 5 candidate models via ternary comparisons (Win/Tie/Loss).Best results are **bolded**.

Backbone	Method	Dimensions						Average
		Perception	Reasoning	Knowledge	Understand	Trustworthiness	Generation	
Gemini-2.5-Pro	POINTWISE	49.72	50.03	55.14	53.87	53.27	54.35	50.84
	BATCH	54.52	55.28	56.36	57.77	55.96	64.78	55.39
	SCAN	56.51	59.21	53.73	56.02	53.27	65.56	57.17
	ATOM(Ours)	**63.10**	**65.72**	**62.80**	**66.09**	**60.96**	**73.48**	**64.31**
GPT-5	POINTWISE	46.72	46.39	54.54	52.34	51.15	54.78	48.06
	BATCH	47.75	51.68	49.71	50.90	47.31	60.43	49.71
	SCAN	52.55	56.10	48.36	52.32	42.59	61.26	53.25
	ATOM(Ours)	**59.94**	**63.68**	**57.60**	**60.78**	**59.42**	**70.22**	**61.17**
GPT-5-mini	POINTWISE	44.60	44.72	50.40	47.10	49.24	53.15	45.66
	BATCH	46.22	47.25	48.58	47.84	42.93	55.81	47.03
	SCAN	50.23	52.82	47.26	51.42	48.34	59.49	50.94
	ATOM(Ours)	**53.95**	**55.28**	**52.69**	**55.86**	**53.93**	**63.17**	**54.53**
Qwen3-VL 235B-A22B	POINTWISE	42.83	42.94	48.41	45.23	47.29	51.04	43.85
	BATCH	42.99	43.93	47.72	43.75	46.38	52.04	44.05
	SCAN	49.38	50.81	47.85	48.35	**52.16**	53.38	49.73
	ATOM(Ours)	**53.70**	**55.86**	**55.44**	**54.63**	52.04	**60.45**	**54.76**
InternVL3.5 241B-A28B	POINTWISE	45.36	45.27	50.30	47.90	51.84	55.40	46.30
	BATCH	41.29	40.52	44.41	41.04	41.29	49.10	41.45
	SCAN	48.42	49.66	50.49	49.83	42.94	56.17	49.19
	ATOM(Ours)	**53.01**	**52.83**	**55.86**	**55.16**	**56.44**	**59.20**	**53.56**
LLaVA-OneVision 72B	POINTWISE	45.80	45.80	50.28	47.84	52.25	54.89	46.65
	BATCH	38.10	38.72	40.38	39.76	39.05	44.11	38.75
	SCAN	47.48	47.53	43.87	48.36	49.98	**61.00**	47.38
	ATOM(Ours)	**50.38**	**49.16**	**53.55**	**49.83**	**59.19**	58.91	**50.52**

computed as the weighted average of these atomic scores:

$$S_{i,k} = \frac{\sum_{t=1}^{T_i} w_t \cdot s_{i,k,t}}{\sum_{t=1}^{T_i} w_t} \tag{2}$$

By incorporating the importance weight w_t, we ensure that critical decision points contribute more significantly to the final evaluation than trivial steps.

3.4 Diagnostic Analysis & Attribution

To provide actionable insights beyond scalar scores, we employ a "saturate-and-select" paradigm to transform quantitative traces into qualitative reports.

Asset Generation and Synthesis. We first generate a comprehensive library of statistical assets, including capability radar charts and stability tables. The Judge Model M_J performs **Information Filtering**, selecting only salient insights, such as significant anomalies or distinct strengths, to ensure reporting focus (see Appendix H for asset details). Reports are synthesized using a bottom-up strategy: M_J first generates model-specific failure attributions and then aggregates cross-model statistics into a global "model profile." These reports serve as the foundation for the fingerprinting validation in §4.4.

3.5 Closed-Loop Feedback Mechanism

Beyond ranking, ATOM bridges evaluation and alignment by transforming diagnostic traces (§3.1) into actionable optimization signals. We establish a **Closed-Loop Feedback Mechanism** with two strategies targeting specific failure types:

Strategy 1: Criterion-Aware Self-Refinement. We construct feedback prompts $\mathcal{F}_{i,k}^{self}$ using queries q_t from failed atoms $\mathcal{A}_{i,k}^{fail}$ (while masking g_t). This explicitly queries neglected visual details, forcing the model to perform *Targeted Visual Regrounding* without leaking the answer.

Strategy 2: Peer-Assisted Knowledge Transfer. Addressing knowledge gaps, we leverage a *Peer Expert* $M_t^* \in \mathcal{M}$ that successfully solved the specific atom $a_{i,t}$. M_t^* generates a *Guiding Hint* h_t, which points to a plausible reasoning path without revealing the final answer and helps transfer fine-grained expertise from stronger to weaker models.

4 Experiment

Moving beyond standard leaderboard rankings, our empirical evaluation is structured to validate **ATOM**'s capacity to resolve the three fundamental limitations of MLLM evaluation highlighted in Section 1. We examine the framework's effectiveness through four key dimensions: **Evaluation Fidelity (§4.1):** Verify whether atomic verifica-

Table 2: **Ablation Study.** Impact of removing key components from the full model. Removing *Atomic Decomposition* leads to the most severe degradation across all models.

Backbone	*Gemini*	*GPT-5*	*GPT-5-mini*	*Qwen3-VL*	*InternVL3.5*	*LLaVA-OV*
Full ATOM (Ours)	**64.31**	**61.17**	**54.53**	**54.76**	**53.56**	**50.52**
– w/o CoT Reasoning	60.33 (↓4.0)	60.02 (↓1.2)	52.80 (↓1.7)	53.50 (↓1.3)	52.77 (↓0.8)	47.16 (↓3.4)
– w/o Atomic Decomposition	58.14 (↓6.2)	56.87 (↓4.3)	47.89 (↓6.7)	49.97 (↓4.8)	48.99 (↓4.6)	44.95 (↓5.6)
– w/o Adaptive Weighting	63.17 (↓1.1)	60.66 (↓0.5)	53.29 (↓1.3)	54.72 (↓0.1)	51.37 (↓2.2)	50.07 (↓0.5)

tion mitigates "Outcome Bias" to achieve superior alignment with human judgments. **Diagnostic Resolution (§4.2):** Assess if the dynamic taxonomy resolves attribution ambiguity by capturing precise capability boundaries compared to static labels. **Closed-Loop Actionability (§4.3):** Validate whether fine-grained feedback breaks the "Black Box" nature of scalar metrics to enable verifiable, training-free self-correction. **Macro-Insights (§4.4):** Explore the utility of aggregated diagnostic reports in revealing unique "Model Fingerprints" and architectural behaviors.

4.1 Evaluation Fidelity: Mitigating Outcome Bias via Process Verification

We validate evaluation fidelity on our benchmark of over 3,000 samples(see Appendix A), each containing five distinct model responses. The task requires ranking these responses, quantified by the accuracy of pairwise comparisons against human annotations. As shown in Table 1, ATOM consistently achieves state-of-the-art alignment across all six judge backbones. Notably, it surpasses the strongest baseline(see Appendix B for baseline details) by **7.92%** with GPT-5, suggesting that atomic decomposition effectively captures subtle logic flaws often overlooked by holistic scoring.

To isolate the source of these gains, our ablation study (Table 2) reveals that the most significant performance drop occurs in the *w/o Atom* setting. This suggests that holistic scoring is prone to outcome bias, as it may validate correct final answers despite flawed intermediate reasoning, while our atomic verification penalizes such spurious successes. Furthermore, removing visual anchoring (*w/o CoT*) leads to consistent degradation, highlighting the necessity of grounding criteria in visual regions. Finally, human verification(Appendix C) confirms the high quality of these generated criteria (Accuracy $> 2.7/3.0$), ensuring that our performance gains stem from rigorous structural advantages rather than noise.

Taxonomy	**#L**	**Valid**	**MAE_G**	**MAE_I**	Δ
Baselines					
Handcraft	41	4667	20.88	22.00	-1.12
Generalist	120	4282	21.28	21.86	-0.58
Scan (Vis.)	52	3813	21.55	21.54	+0.01
MMBench Orig.	21	4757	20.53	20.26	+0.27
No Data	54	4296	21.19	20.75	+0.44
Ours					
Ours (No Prune)	65	4548	20.08	18.90	+1.18
Ours (Final)	**52**	**4557**	20.40	**18.83**	**+1.57**

Table 3: **Taxonomy Validity Assessment.** Positive Δ indicates effective clustering of homogeneous capabilities. **#L**: Leaf count. **MAE_G/MAE_I**: Global vs. Intra-node prediction error.

4.2 Diagnostic Resolution: From Macroscopic Labels to Dynamic Attribution

We next examine whether our Data-Driven Dynamic Taxonomy resolves the ambiguity of macroscopic taxonomies by capturing precise capability boundaries. We quantify this using a homogeneity hypothesis: a valid capability node should group atomic skills of similar difficulty, thereby minimizing intra-node variance.

Static vs. Dynamic Constraints. Using a Leave-One-Out (LOO) strategy (formal definitions in Appendix F.1), the results in Table 3 reveal a critical insight. Our dynamically evolved tree significantly reduces prediction uncertainty ($\Delta = +\mathbf{1.57}$) compared to the global baseline. In contrast, rigid, handcrafted taxonomies often yield negative gains ($\Delta < 0$). This quantitative gap suggests that static labels may misclassify heterogeneous cognitive tasks into coarse categories, whereas our data-driven approach better captures the underlying capability structure.

Granular Error Attribution. This structural precision enables diagnostic resolution beyond broad domain labels. As illustrated in the case study (Figure 3), instead of labeling a failure simply as "Reasoning," ATOM traces the error to a specific leaf node in the hierarchy (visualized in

Table 4: Correction performance on candidate models' initially failed samples. Upper rows (Black) denote final response Accuracy **(0-100%)**, while lower rows (Green) indicate Atom-Level Quality Scores **(1–5)** evaluated by ATOM. All strategies operate without revealing ground truth; *Re-eval* prompts a retry upon error notification, whereas ATOM methods utilize diagnostic criteria. Gain(Δ) denotes improvements over the *Re-eval*.

Candidate Model	Strategy	Capabilities							Gain (Δ)
		Perception	Reasoning	Knowledge	Understanding	Trustworthiness	Generation	**Average**	
Gemini-2.5-Pro	Ori	0.00 2.94	0.00 2.78	0.00 2.88	0.00 3.29	0.00 2.78	0.00 2.60	**0.00** **2.90**	– –
	Re-eval.	22.25 3.12	24.34 2.91	38.00 3.04	30.23 3.36	46.15 2.73	55.56 2.96	**25.24** **3.04**	+0.00 +0.00
	Atom-Self-corr.	29.67 3.55	38.71 3.58	36.00 3.54	34.88 4.01	53.85 4.19	11.11 3.48	**34.27** **3.62**	**+9.03** +0.58
	Peer-Assisted	32.97 3.80	41.06 4.12	36.00 4.01	51.16 4.32	61.54 4.41	33.33 4.43	**37.93** **4.01**	**+12.69** +0.97
Qwen3-VL 235B-A22B	Ori	0.00 2.21	0.00 2.23	0.00 1.71	0.00 2.46	0.00 1.41	0.00 2.00	**0.00** **2.19**	– –
	Re-eval.	10.41 2.27	10.76 2.32	7.59 1.89	11.11 2.60	3.45 1.42	4.76 2.11	**10.08** **2.28**	+0.00 +0.00
	Atom-Self-corr.	25.85 3.12	26.29 3.15	31.65 3.27	28.40 3.63	79.31 1.69	2.38 3.11	**26.98** **3.11**	**+16.90** +0.83
	Peer-Assisted	32.32 3.65	29.88 3.92	43.04 3.64	32.10 3.76	75.86 4.46	2.38 2.36	**32.02** **3.74**	**+21.94** +1.46
Qwen2-VL 72B	Ori	0.00 1.84	0.00 1.74	0.00 1.82	0.00 2.13	0.00 1.17	0.00 1.56	**0.00** **1.81**	– –
	Re-eval.	7.56 1.92	6.43 1.80	8.60 1.87	7.78 2.22	5.56 1.12	0.00 1.46	**6.96** **1.87**	+0.00 +0.00
	Atom-Self-corr.	26.60 2.81	21.14 2.77	38.71 3.17	32.22 3.07	77.78 3.31	2.33 1.85	**26.24** **2.83**	**+19.28** +0.96
	Peer-Assisted	32.12 3.31	29.60 3.56	50.54 3.84	43.33 3.58	77.78 4.14	4.65 2.72	**33.33** **3.46**	**+26.37** +1.59
InternVL3.5 241B-A28B	Ori	0.00 1.94	0.00 1.68	0.00 1.57	0.00 2.14	0.00 1.11	0.00 1.76	**0.00** **1.82**	– –
	Re-eval.	14.35 2.03	11.48 1.82	10.53 1.57	19.05 2.33	22.86 1.32	0.00 1.76	**13.10** **1.93**	+0.00 +0.00
	Atom-Self-corr.	29.48 2.86	29.78 2.61	36.84 2.93	39.05 3.06	80.00 3.07	0.00 2.04	**31.04** **2.77**	**+17.94** +0.84
	Peer-Assisted	37.50 3.47	38.77 3.60	47.37 3.75	52.38 3.83	88.57 4.14	6.98 2.88	**39.95** **3.58**	**+26.85** +1.65
LLaVA-OneVision 72B	Ori	0.00 1.89	0.00 1.47	0.00 1.67	0.00 2.10	0.00 1.23	0.00 1.38	**0.00** **1.71**	– –
	Re-eval.	5.76 1.93	4.46 1.52	7.44 1.75	6.30 2.09	7.89 1.33	0.00 1.35	**5.32** **1.75**	+0.00 +0.00
	Atom-Self-corr.	22.25 2.64	19.76 2.37	33.06 2.93	24.41 2.85	71.05 2.91	2.27 1.73	**22.75** **2.56**	**+17.43** +0.81
	Peer-Assisted	24.21 3.07	26.45 3.05	38.84 3.29	32.28 3.45	81.58 3.89	4.55 2.70	**27.39** **3.13**	**+22.07** +1.38

Figure 12), such as failing at *verify & exclude* despite succeeding in *visual identification*. This level of resolution effectively disentangles perception blindness from reasoning deficits, providing the necessary prerequisite for targeted improvement.

4.3 Closed-Loop Actionability: Breaking the Black Box for Self-Correction

Moving beyond static diagnosis, We investigate whether diagnostic feedback can effectively guide candidate models to rectify their own errors. To this end, we isolate the subset of samples where each model initially failed (Ori), establishing a 0% baseline.

From Diagnosis to Self-Refinement. Table 4 reveals a stark contrast. Simple *Re-evaluation*, which signals an error and requests a retry, yields negligible gains. In contrast, *Atom-Guided Strategy* uses diagnostic criteria to guide reasoning (while masking specific ground truths), achieving substantial improvements in Accuracy (**+26.85%** on InternVL3.5). Crucially, the concurrent rise in Atom-Level Quality Scores (**1.82** → **3.58**) confirms that the framework rectifies the underlying reasoning process rather than merely refining the output format.

Peer-Assisted Knowledge Transfer. Furthermore, the *Peer-Assisted* setting demonstrates that atomic reasoning paths are transferable. By leveraging diagnostic traces from a stronger peer (e.g., Gemini) to guide a weaker model, we observe even higher accuracy gains, proving that atomic criteria serve as interpretable, model-agnostic instructions.

4.4 Macro-Insights: Model Fingerprinting and Behavioral Profiling

Finally, we explore whether the aggregated diagnostic reports can serve as unique "fingerprints" to reveal architectural behaviors that holistic scores obscure.

Discriminability and Fidelity. We conducted a blinded **Model Fingerprinting Experiment** (protocol in Appendix G.1), where human annotators matched diagnostic reports to raw model responses.

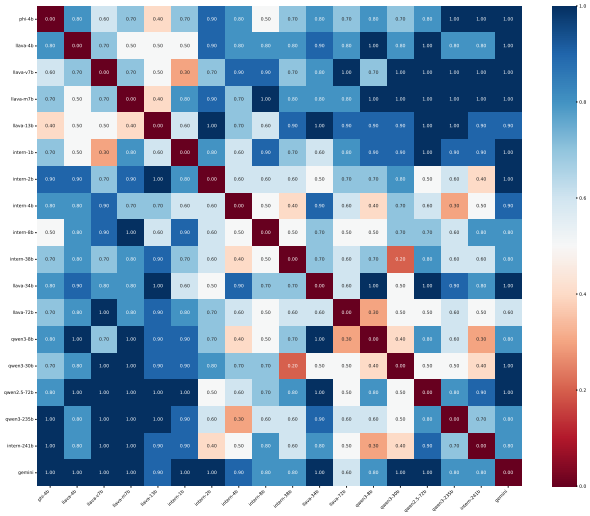


Figure 4: **Model Fingerprinting Accuracy Heatmap.** This heatmap visualizes accuracy in matching reports to models. Higher values (darker blue) indicate strong discriminability; lower values reflect genuine behavioral similarities within model families.

The Similarity Heatmap (Figure 4) demonstrates that ATOM's reports capture distinct behavioral signatures, achieving near-perfect identification accuracy ($\approx$ **1.00**) when distinguishing between different model families (`Intern` vs. `LLaVA`). Conversely, the reduced accuracy among closely related variants (`InternVL3.5-1B` vs. `2B`) actually validates the *fidelity* of our insights: the framework correctly reflects genuine behavioral commonalities rather than fabricating artificial distinctions, ensuring the analysis is both precise and trustworthy.

Granular Capability Landscape. Beyond identification, the generated assets, including capability radar charts and fine-grained skill bars (Figure 16), provide a structured view of model evolution. These visualizations reveal that even models with similar overall rankings often possess drastically different capability profiles, providing actionable roadmaps for architectural optimization.

5 Related Works

Evaluating Multimodal Large Language Models (MLLMs) has evolved from static QA benchmarks (Fu et al., 2023; Liu et al., 2024; Yue et al., 2024; Lu et al., 2023) to dynamic, judge-based assessments, aiming to capture the complexity of open-ended generation.

MLLM-as-a-Judge. The "LLM-as-a-Judge" paradigm (Zheng et al., 2023; Zhu et al., 2023; Kim et al., 2023) reduces reliance on human annotation. Recent multimodal adaptations, such as the MLLM-as-a-Judge benchmark (Chen et al., 2024), UPME (Zhang et al., 2025b), and AUTO-J (Li et al., 2023), have focused on assessing judge consistency and mitigating hallucinations. However, these frameworks predominantly operate on a *holistic* level, assigning coarse-grained scores based on general impressions. Our approach shifts from this "black-box" scoring to *Atom Criteria* verification, requiring judges to validate indivisible evidence points for enhanced interpretability.

Rating & Ranking. While the Elo rating system (Bradley and Terry, 1952; Chiang et al., 2024) remains the industry standard, it is susceptible to biases like verbosity (Saito et al., 2023). Advanced methods such as AlpacaEval 2.0 (Dubois et al., 2024), TrueSkill (Herbrich et al., 2006), and PolyRating (Dekoninck et al., 2024) introduce length-control or Bayesian modeling to refine ranking robustness. Despite these statistical improvements, they primarily yield scalar rankings (e.g., "Model A $>$ Model B"). In contrast, our framework complements these metrics by grounding scores in a structured Capability Tree, generating detailed diagnostic reports and "model fingerprints" rather than simple numerical values.

Structured Evaluation. Recent works emphasize structured capability assessment. (Lin et al., 2024) and (Li et al., 2024) highlight the importance of real-world query distributions. Notably, SCAN (Wang et al., 2025) and EvalTree (Zeng et al., 2025) introduce hierarchical taxonomies to map performance across domains. Crucially, while these works categorize based on *query types* (e.g., coding vs. math tasks), we decompose the *solution process* itself. By constructing a taxonomy based on execution criteria (e.g., spatial perception $\rightarrow$ logic inference), we provide a bottom-up diagnostic view that identifies latent bottlenecks often missed by query-based classifications.

6 Conclusion

In this paper, we introduced **ATOM**, a novel fine-grained diagnostic framework that shifts the MLLM evaluation paradigm from static, holistic ranking to structural, process-oriented diagnosis. Our approach effectively mitigates the "Outcome Bias" and reduces hallucinations inherent in traditional "Black Box" judge models. Beyond evalu-

ation, **ATOM** establishes a critical link between assessment and model alignment. Collectively, ATOM functions as both a high-precision, human-aligned judge and a robust guide for the continuous optimization of next-generation multimodal systems.

Limitations

Despite the promising results achieved by ATOM in fine-grained diagnostics and visual anchoring, we acknowledge several limitations.

First, we acknowledge that ATOM's current implementation is primarily designed for single-turn QA and independent generation tasks. This scoping aligns with recent structured evaluation frameworks, prioritizing the establishment of a robust verification protocol for static visual reasoning before introducing complex conversational state tracking.

Second, our framework relies on the capability of the Judge Model. Although Visual Anchoring significantly reduces hallucinations, the accuracy of the generated "Textbook-Quality Chain of Thought" and subsequent evaluation still depends on the underlying strong MLLM. Extremely subtle visual details or highly specialized domain knowledge might still exceed the judge's perception limits.

Finally, the Dynamic Capability Tree is constructed in a data-driven manner. While this ensures high alignment with the current benchmark, applying the framework to a radically different domain may require re-evolving the taxonomy to capture new capability dimensions effectively.

Ethics Statement

This research investigates the evaluation and self-improvement of Multimodal Large Language Models (MLLMs) and follows standard ethical practices in data collection and annotation.

In our human meta-evaluation and diagnostic validation phases, we recruited 12 annotators with graduate-level training. Prior to the tasks, all participants were fully informed of the research objectives and the intended academic use of the collected data. We obtained explicit informed consent from all individuals. To ensure fair labor practices, annotators were compensated with a competitive hourly wage that aligns with or exceeds the local standard for research assistants. We strictly prohibited the collection of any personally identifiable information (PII) during the annotation process, ensuring complete anonymity.

We carefully curated the benchmark data and multimodal queries used in this study. All samples underwent rigorous pre-screening to filter out offensive, toxic, or sensitive content. This rigorous filtering not only guaranteed a safe working environment for our human annotators but also prevents the propagation of harmful materials in future evaluations.

Our proposed ATOM framework relies on state-of-the-art MLLMs to serve as the Judge Model. While our "Cognitive Replay" and visual anchoring mechanisms significantly mitigate hallucinations and the "Outcome Bias" in evaluation, we acknowledge that the judge models themselves may carry inherent cultural, linguistic, or demographic biases stemming from their pre-training corpora. Consequently, the dynamically evolved Capability Tree and the generated atomic criteria might inadvertently reflect the perspectives of these specific models. We encourage the community to continuously audit strong judge models for underlying societal biases.

The primary objective of the ATOM framework is to promote transparency, reliability, and safety in multimodal AI systems. By transitioning from opaque, holistic scoring to fine-grained, verifiable diagnostics, our work prevents models from masking fundamental reasoning deficits behind seemingly correct final answers. The autonomous self-correction mechanism further provides a verifiable path for aligning AI models with human values without requiring extensive retraining. We foresee no direct risk of malicious dual-use arising from our evaluation framework.

Acknowledgments

This work is supported by research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology. We would like to express our sincere gratitude to Tencent and the Tencent Hunyuan team for providing invaluable support, substantial computational resources, and a highly collaborative research environment during the course of this study. We also extend our special thanks to all the human annotators for their diligent efforts and meticulous work in our evaluation process.

References

Mohammed Talha Alam, Raza Imam, Mohsen Guizani, and Fakhri Karray. 2024. Flare up your data: Diffusion-based augmentation method in astronomical imaging. *arXiv preprint arXiv:2405.13267*.

Lukas Blecher. 2022. Latex-ocr. GitHub. https://github.com/lukas-blecher/LaTeX-OCR.

Rumeysa Bodur, Erhan Gundogdu, Binod Bhattarai, Tae-Kyun Kim, Michael Donoser, and Loris Bazzani. 2024. iedit: Localised text-guided image editing with weak supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7426–7435.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16495–16504.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. 2024. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. In *Forty-first International Conference on Machine Learning*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, and others. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.

Romrawin Chumpu. Multimodal neural translation. https://huggingface.co/datasets/romrawinjp/multi30k.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. 2024. Polyrating: A cost-effective and bias-aware rating system for llm evaluation. *arXiv preprint arXiv:2409.00696*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Hao Fei, Yuan Zhou, Juncheng Li, Xiangtai Li, Qingshan Xu, Bobo Li, Shengqiong Wu, Yaoting Wang, Junbao Zhou, Jiahao Meng, and others. 2025. On path to multimodal generalist: General-level and general-bench. In *Forty-second International Conference on Machine Learning*.

Maxwell Forbes, Christine Kaeser-Chen, Piyush Sharma, and Serge Belongie. 2019. Neural naturalist: Generating fine-grained image comparisons. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 708–717.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and others. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.

Noa Garcia, Chentao Ye, Zihua Liu, Qingtao Hu, Mayu Otani, Chenhui Chu, Yuta Nakashima, and Teruko Mitamura. 2020. A dataset and baselines for visual question answering on art. In *European conference on computer vision*, pages 92–108. Springer.

Wentao Ge, Shunian Chen, Hardy Chen, Nuo Chen, Junying Chen, Zhihong Chen, Wenya Xie, Shuo Yan, ChenghaoZhu ChenghaoZhu, Ziyue Lin, and others. 2025. Mllm-bench: evaluating multimodal llms with per-sample criteria. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4951–4974.

Tanmay Gupta, Dustin Schwenk, Ali Farhadi, Derek Hoiem, and Aniruddha Kembhavi. 2018. Imagine this! scripts to compositions to videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 598–613.

Xintong Han, Zuxuan Wu, Phoenix X Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S Davis. 2017. Automatic spatially-aware fashion concept discovery. In *Proceedings of the IEEE international conference on computer vision*, pages 1463–1471.

Zhen Han, Chaojie Mao, Zeyinzi Jiang, Yulin Pan, and Jingfeng Zhang. 2025. Stylebooth: Image style editing with multimodal instruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1947–1957.

Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. Trueskill™: a bayesian skill rating system. *Advances in neural information processing systems*, 19.

Meng-Ru Hsieh, Yen-Liang Lin, and Winston H Hsu. 2017. Drone-based object counting by spatially regularized regional proposal network. In *Proceedings of the IEEE international conference on computer vision*, pages 4145–4153.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Mohit Iyyer, Varun Manjunatha, Anupam Guha, Yogarshi Vyas, Jordan Boyd-Graber, Hal Daume, and Larry S Davis. 2017. The amazing mysteries of the gutter: Drawing inferences between panels in comic

book narratives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pages 7186–7195.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910.

Kaggle. 2019a. American sign language dataset. https://www.kaggle.com/datasets/ayuraj/asl-dataset.

Kaggle. 2019b. C-nmc 2019 dataset. https://www.kaggle.com/datasets/shafiullahshafin/c-nmc-2019-dataset.

Kaggle. 2020a. https://www.kaggle.com/datasets/andrewmvd/car-plate-detection.

Kaggle. 2020b. Emotion detection (fer). https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer.

Kaggle. 2020c. Instagram images with captions. https://www.kaggle.com/datasets/prithvijaunjale/instagram-images-with-captions.

Kaggle. 2021a. https://www.kaggle.com/datasets/misrakahmed/vegetable-image-dataset.

Kaggle. 2021b. Famous iconic women. https://www.kaggle.com/datasets/fatiimaezzahra/famous-iconic-women.

Kaggle. 2021c. Standard ocr dataset. https://www.kaggle.com/datasets/preatcher/standard-ocr-dataset.

Kaggle. 2022a. Micro-organism image classification. https://www.kaggle.com/datasets/mdwaquarazam/microorganism-image-classification.

Kaggle. 2022b. Sketch2code. https://www.kaggle.com/datasets/vshantam/sketch2code.

Kaggle. 2022c. Yoga pose classification. https://www.kaggle.com/datasets/ujjwalchowdhury/yoga-pose-classification.

Kaggle. 2023a. 100 sports image classification. https://www.kaggle.com/datasets/gpiosenka/sports-classification.

Kaggle. 2023b. Animals-10. https://www.kaggle.com/datasets/alessiocorrado99/animals10.

Kaggle. 2023c. Dog breeds. https://www.kaggle.com/datasets/mohamedchahed/dog-breeds.

Kaggle. 2023d. Facial emotion recognition image dataset. https://www.kaggle.com/datasets/sujaykapadnis/emotion-recognition-dataset.

Kaggle. 2023e. Medical visual question answering. https://www.kaggle.com/datasets/mitanshuchakrawarty/medical-visual-question-answering.

Kaggle. 2023f. Seeds counting. https://www.kaggle.com/datasets/raj123verma/seeds-counting.

Kaggle. 2023g. Strawberry dataset. https://www.kaggle.com/datasets/abdulbasit31/strawberry-dataset.

Kaggle. 2024a. Pumpkin leaf diseases dataset from bangladesh. https://www.kaggle.com/datasets/tahmidmir/pumpkin-leaf-diseases-dataset-from-bangladesh.

Kaggle. 2024b. Radar threat object classification. https://www.kaggle.com/datasets/rauldbcet/radar-threat-object-classification.

Kaggle. 2024c. Terrain recognition dataset. https://www.kaggle.com/datasets/krishuppal/terrain-recognition.

Kaggle. 2025a. Ecg images, national heart foundation of bangladesh. https://www.kaggle.com/datasets/drsaeedmohsen/ecg-images-national-heart-foundation-of-banglades.

Kaggle. 2025b. Guava fruit disease dataset. https://www.kaggle.com/datasets/asadullahgalib/guava-disease-dataset.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pages 4999–5007.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and others. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, and others. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Jiayi Kuang, Ying Shen, Jingyou Xie, Haohao Luo, Zhe Xu, Ronghao Li, Yinghui Li, Xianfeng Cheng, Xika Lin, and Yu Han. 2025. Natural language understanding and inference with mllm in visual question answering: A survey. *ACM Computing Surveys*, 57(8):1–36.

Yann LeCun. Digital consistency comparison (mnist). https://huggingface.co/datasets/ylecun/mnist.

Seongyun Lee, Seungone Kim, Sue Park, Geewook Kim, and Minjoon Seo. 2024. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11286–11315.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. 2014. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 152–159.

Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. 2019. Storygan: A sequential conditional gan for story visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6329–6338.

Yongqi Li, Wenjie Li, and Liqiang Nie. 2022. Mmcoqa: Conversational question answering over text, tables, and images. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4220–4231.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2024. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. *arXiv preprint arXiv:2406.04770*.

Jiazhen Liu, Yuhan Fu, Ruobing Xie, Runquan Xie, Xingwu Sun, Fengzong Lian, Zhanhui Kang, and Xirong Li. 2025. Phd: A chatgpt-prompted visual hallucination evaluation dataset. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19857–19866.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and others. 2024. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

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*.

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

Xiaoqiang Lu, Binqiang Wang, Xiangtao Zheng, and Xuelong Li. 2017. Exploring models and data for remote sensing image caption generation. *IEEE Transactions on Geoscience and Remote Sensing*, 56(4):2183–2195.

Adyasha Maharana, Darryl Hannan, and Mohit Bansal. 2022. Storydall-e: Adapting pretrained text-to-image transformers for story continuation. In *European conference on computer vision*, pages 70–87. Springer.

Shu Pu, Yaochen Wang, Dongping Chen, Yuhang Chen, Guohao Wang, Qi Qin, Zhongyi Zhang, Zhiyuan Zhang, Zetong Zhou, Shuang Gong, and others. 2025. Judge anything: Mllm as a judge across any modality. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 5742–5753.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. 2025. A survey of multilingual large language models. *Patterns*, 6(1).

Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. Verbosity bias in preference labeling by large language models. *arXiv preprint arXiv:2310.10076*.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. 2023. Hierarchical multimodal transformers for multipage docvqa. *Pattern Recognition*, 144:109834.

Junjue Wang, Zhuo Zheng, Zihang Chen, Ailong Ma, and Yanfei Zhong. 2024. Earthvqa: Towards queryable earth via relational reasoning-based remote sensing visual question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 5481–5489.

Zongqi Wang, Tianle Gu, Chen Gong, Xin Tian, Siqi Bao, and Yujiu Yang. 2025. Scan: Structured capability assessment and navigation for llms. *arXiv preprint arXiv:2505.06698*.

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. Recipeqa: A challenge dataset for multimodal comprehension of cooking recipes. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1368.

Zhiyuan Yan, Taiping Yao, Shen Chen, Yandan Zhao, Xinghe Fu, Junwei Zhu, Donghao Luo, Chengjie Wang, Shouhong Ding, Yunsheng Wu, and others. 2024. Df40: Toward next-generation deepfake detection. *Advances in Neural Information Processing Systems*, 37:29387–29434.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yanbin Yin, Kun Zhou, Zhen Wang, Xiangdong Zhang, Yifei Shao, Shibo Hao, Yi Gu, Jieyuan Liu, Somanshu Singla, Tianyang Liu, and others. 2025. Decentralized arena: Towards democratic and scalable automatic evaluation of language models. *arXiv preprint arXiv:2505.12808*.

Kaining Ying, Fanqing Meng, Jin Wang, Zhiqian Li, Han Lin, Yue Yang, Hao Zhang, Wenbo Zhang, Yuqi Lin, Shuo Liu, and others. 2024. Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi. *arXiv preprint arXiv:2404.16006*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 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.

Zhiyuan Zeng, Yizhong Wang, Hannaneh Hajishirzi, and Pang Wei Koh. 2025. Evaltree: Profiling language model weaknesses via hierarchical capability trees. *arXiv preprint arXiv:2503.08893*.

Chenhao Zhang, Xi Feng, Yuelin Bai, Xeron Du, Jinchang Hou, Kaixin Deng, Guangzeng Han, Qinrui Li, Bingli Wang, Jiaheng Liu, and others. 2025a. Can mllms understand the deep implication behind chinese images? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14369–14402.

Qihui Zhang, Munan Ning, Zheyuan Liu, Yue Huang, Shuo Yang, Yanbo Wang, Jiayi Ye, Xiao Chen, Yibing Song, and Li Yuan. 2025b. Upme: An unsupervised peer review framework for multimodal large language model evaluation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9165–9174.

Ruochen Zhao, Wenxuan Zhang, Yew Ken Chia, Weiwen Xu, Deli Zhao, and Lidong Bing. 2025. Autoarena: Automating llm evaluations with agent peer battles and committee discussions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4440–4463.

Yu Zhao, Jianguo Wei, Zhichao Lin, Yueheng Sun, Meishan Zhang, and Min Zhang. 2022. Visual spatial description: Controlled spatial-oriented image-to-text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1437–1449.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.

A Dataset Composition and Statistics

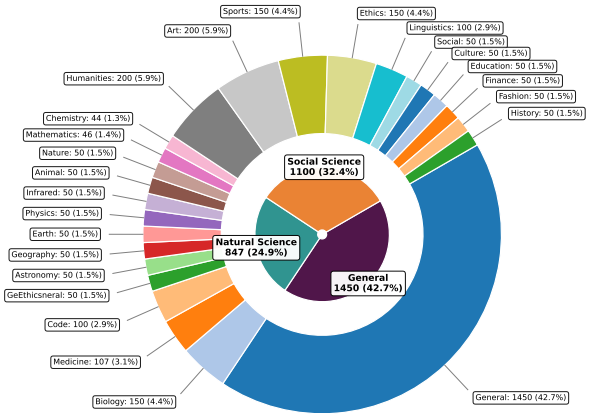


Figure 5: Data distribution across domains and subcategories.

To provide a comprehensive understanding of our evaluation benchmark, we visualize the distribution of the curated samples across different domains. Organized following the methodology of (Fei et al., 2025), the dataset is designed to cover a wide range of multimodal capabilities and knowledge areas. Full data sources and details are provided in Table 6.

Figure 5 illustrates the hierarchical composition of the benchmark. The data is categorized into three primary domains: **General** (42.7%), **Social Science** (32.4%), and **Natural Science** (24.9%). These are further divided into fine-grained subcategories, including but not limited to Medicine, Physics, History, and Culture, ensuring a balanced and rigorous assessment of MLLMs.

Dataset Construction and Annotation Protocol. To construct the candidate response pool, we generated outputs for each sample using five representative MLLMs: **Gemini-2.5-Pro**, **Qwen3-VL-235B-A22B-Instruct**, **Qwen2-VL-72B-Instruct**, **InternVL3.5-241B-A28B-Instruct**, and **LLaVA-OneVision-Qwen2-72B**. The ground truth rankings were established through a rigorous human annotation campaign involving 12 qualified annotators. Each sample was independently evaluated by two annotators. To ensure high-quality labels, we implemented a consensus-based conflict resolution mechanism: in cases of disagreement, a third annotator joined to adjudicate and discuss with the original pair until a final consensus was reached. The specialized interface used for this process is illustrated in Figure 18.

Experimental Stability. Finally, unless otherwise specified, all experimental results reported in this paper represent the average of three independent runs to ensure statistical reliability.

Cap.	Metric	Gemini	GPT-5	GPT-5-mini	Qwen3-VL	InternVL3.5	LLaVA-OV
Perc.	Acc.	**2.73**	2.72	2.61	2.64	2.52	2.69
	Nec.	**2.87**	2.84	2.80	2.78	2.77	2.56
Reas.	Acc.	2.64	**2.65**	2.61	2.62	2.62	2.33
	Nec.	**2.85**	2.81	2.66	2.73	2.70	2.71
Know.	Acc.	2.82	**2.86**	2.84	2.85	2.81	2.58
	Nec.	**2.74**	2.72	2.70	2.70	2.58	2.59
Und.	Acc.	**2.81**	2.69	2.62	2.69	2.64	2.55
	Nec.	**2.83**	2.63	2.62	2.67	2.64	2.51
Trust.	Acc.	2.05	2.58	**2.68**	2.62	2.14	2.26
	Nec.	**3.00**	2.83	2.53	2.75	2.68	2.52
Gen.	Acc.	2.36	**3.00**	2.50	2.76	2.73	2.00
	Nec.	2.45	**2.71**	2.47	2.62	2.62	2.50
Avg.	**Acc.**	**2.72**	**2.73**	**2.62**	**2.66**	**2.61**	**2.58**
	Nec.	**2.84**	**2.80**	**2.72**	**2.75**	**2.73**	**2.60**

Table 5: Combined evaluation. Blue rows : **Accuracy**, Orange rows : **Necessity**.

B Baseline Implementation Details

To rigorously evaluate the ranking effectiveness of ATOM, we compare it against three representative paradigms in automated evaluation. All baselines utilize the same judge backbones as our method to ensure fair comparison.

Pointwise Scoring (Holistic). The judge model evaluates a single candidate response (I, Q, R) independently. It is prompted to assign a scalar score based on general quality dimensions such as helpfulness, relevance, and accuracy. This method represents the most coarse-grained evaluation, susceptible to the generic "helpfulness bias" of LLMs.

Batch Ranking. The judge model is presented with the visual input I, the query Q, and a set of anonymized responses $\{R_1, \ldots, R_k\}$ from different models simultaneously. The judge is instructed to rank these responses based on their overall quality. This approach, widely used in benchmarks like Chatbot Arena (Automated), allows for relative comparison but often struggles with long-context attention and lacks explicit justification for its rankings.

SCAN (Adapted). We adapt the *Pre-Comparison-based Pointwise Evaluation*(Wang et al., 2025) to the multimodal setting. SCAN bridges the gap between pointwise and pairwise evaluation through a three-stage process:

Prompt: Chain of Thought (CoT) Generation

```
#### System Prompt
You are an expert specialized in multimodal question answering, generating a "Textbook-Quality
Chain of Thought". Your goal is to generate a "Textbook-Quality Chain of Thought"(CoT). This CoT
will serve as a high-quality exemplar for other models.

#### Instructions
0. You should record your whole thinking process in the output.
1. Carefully examine the input image, and the question to understand all components of the task.
2. Begin your CoT by providing a concise summary of the image, extracting all useful content from
   the image.
3. Construct a step-by-step reasoning process that logically progresses from the image analysis
   to the final answer. Each step should build upon the previous one, clearly explaining the
   deductions made.
4. Conclude your reasoning process by explicitly stating the final answer.

#### Output Format
Strictly output your analysis within the following XML tags:
<cot>
[Your detailed, complete thinking process.]
</cot>
<answer>
[Your final answer to the user's question.]
</answer>

#### Input Data
<question> {prompt_text} </question>
<reference_answer> {reference_answer} </reference_answer>
```

Figure 6: The system prompt used for generating Textbook-Quality Chain of Thought (CoT).

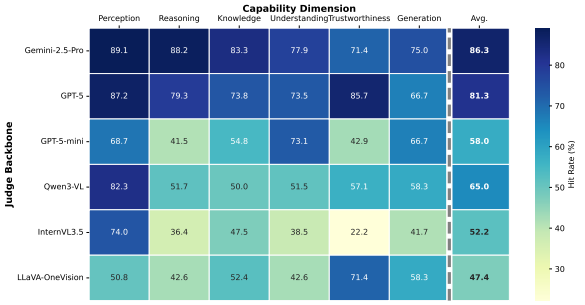


Figure 7: Capability tagging hit rate across different judge backbones and capability dimensions. The last column shows the average hit rate.

1. **Criteria Extraction:** We prompt the judge to compare a diverse set of auxiliary responses $\{y_1, \ldots, y_n\}$ to identify key distinguishing factors.

2. **Weighting:** The judge assigns importance weights W to these extracted criteria based on their relevance to the query.

3. **Scoring:** The target response is evaluated against these weighted criteria to produce a final score.

Unlike ATOM, SCAN derives criteria solely from *textual response differences* without explicit visual grounding, making it a strong baseline for structured evaluation but potentially weaker in verifying visual faithfulness.

Clarification on Pointwise vs. w/o Atomic Decomposition. To clarify the distinction between the baseline methods in Table 1 and the ablation settings in Table 2, we explicitly highlight the difference between the *Pointwise* baseline and the *w/o Atomic Decomposition* ablation:

- **Pointwise (Table 1):** This serves as the standard holistic baseline. The Judge model directly evaluates the candidate response and outputs a single scalar score. It does not generate any intermediate reasoning steps or structured criteria.

- **w/o Atomic Decomposition (Table 2):** In this ablation setting, the Judge retains the "Cognitive Replay" (CoT generation) phase and still attempts to score multiple fine-grained aspects. However, it explicitly skips the "Criteria Extraction" phase. Instead of deconstruct-

Prompt: Atomic Criteria Extraction

```
#### System Prompt
You are an expert specialized in multimodal question answering, extracting **Critical Point**
from a question and its solution. A **Critical Point** is defined as an essential step, a
significant challenge, a key milestone, or a crucial checkpoint within the solution. These points
serve as standardized scoring criteria for evaluating other potential solutions. Your goal is
to deconstruct the provided solution into a set of Critical Points. Each Critical Point must
contain a criterion, a ground truth answer, and an allocated weight.

#### Instructions
1. Carefully examine the input image, and the question, and the reference CoT (i.e. the solution).
2. Identify the essential steps, significant challenges, key milestones, crucial checkpoints of
   the solution. This could include factual or logical or other content.
3. Select 3 to 5 of the most significant points to formulate as Critical Points.
4. For each Critical Point, deconstruct the point into open-ended question (namely criterion)
   and corresponding ground truth. Avoid embedding the ground truth within the criterion itself.
   Make sure that the criterion is about how to evaluate another predefined solution.
5. Allocate a weight to each Critical Point. Assume the total weight for the entire question is
   10. Distribute these weights based on the relative difficulty and importance of each Critical
   Point.

#### Output Format
Strictly output your analysis within the following json format:
[
  {{
    "criterion": "a open-ended question evaluating a predefined solution, start with an
interrogative word like what, etc.",
    "ground_truth": "The corresponding answer to the criterion, derived from the reference
solution",
    "weight": 2
  }},
  ......
]

#### Input Data
<question> {prompt_text} </question>
<cot> {cot} </cot>
```

Figure 8: The system prompt used for extracting fine-grained Atomic Criteria from the solution.

ing the task into independent, verifiable tuples ($\langle q_t, g_t, w_t \rangle$) for step-by-step verification, the Judge implicitly assigns scores based on the general context of the generated CoT.

This comparison underscores that merely prompting the judge to generate a CoT is insufficient; the explicit structural breakdown into atomic criteria is essential for pinpointing errors and mitigating the Outcome Bias.

C Atomic Criteria Quality Verification

The reliability of our framework hinges on the quality of the generated Atomic Criteria. To validate this, we conducted a human meta-evaluation on a stratified sample of generated criteria. Human experts rated each criterion on a 3-point scale across two dimensions:

- **Accuracy:** Is the local ground truth (g_t) factually correct with respect to the image content?
- **Necessity:** Is the sub-question (q_t) necessary for answering the user's original query?

Table 5 presents the results. The high average scores across all judge backbones (Accuracy $> 2.6/3.0$, Necessity $> 2.7/3.0$) confirm that modern MLLMs, particularly Gemini and GPT class models, can generate high-fidelity, visually grounded checkpoints. This validates that the performance gains observed in our main experiments are driven by the structural advantages of our framework rather than noise in the criteria generation process.

Figure 7 reveals a critical insight: while most models can generate valid atomic questions, they exhibit significant disparity in mapping these ques-

Prompt: Atomic Capability Mapping

```
#### Instructions
You are given an input JSON list called "atom", where each element contains a criterion,
ground_truth, and weight. Your task is to map each criterion to exactly one leaf node in
the capability tree provided below. Then output a JSON list in which each original element from
"atom" is copied verbatim, with the only modification being the addition of a new field "cap",
whose value is one leaf-node path in the format:

"<Level1><Level2><Level3>"

For example: "<Knowledge><Knowledge Retrieval><Table Lookup>".

#### Capability Tree
{Tree}

#### Output Format
Strictly output:
[
  {{
    ... original atom element fields ...,
    "cap": "one selected leaf node path"
  }},
  ...
]

#### Input Data
<question> {prompt_text} </question>
<atom> {atom} </atom>
```

Figure 9: The system prompt used for mapping atomic criteria to the dynamic capability tree.

tions to the abstract capability taxonomy. Strong models like `Gemini` and `GPT-5` achieve high alignment with human experts in capability attribution (Hit Rate $> 81\%$), whereas weaker models struggle with abstract dimensions such as *Trustworthiness* and *Reasoning*. This empirical finding provides robust justification for our **Dynamic Capability Tree** construction strategy: employing the "higher-order cognition" of SOTA models as the backbone is indispensable for ensuring the structural precision and human alignment of the diagnostic reports.

D Prompt Templates

To facilitate reproducibility, we provide the full system prompts employed across the different stages of the ATOM framework. The specific prompts for each module are organized as follows:

- **Cognitive Replay (CoT Generation):** The prompt used to generate the textbook-quality chain of thought is shown in Figure 6.
- **Atomic Criteria Extraction:** The prompt for decomposing reasoning into fine-grained checkpoints is presented in Figure 8.
- **Capability Mapping:** The instruction for mapping atomic criteria to the dynamic capability tree is detailed in Figure 9.
- **Fine-Grained Scoring:** The final evaluation prompt used by the judge to score candidate responses against atomic criteria is provided in Figure 10.

E Details of Dynamic Capability Taxonomy Construction

In this section, we provide the implementation details, algorithmic parameters, and the visual structure of the capability taxonomy $\mathcal{T}_{cap}$ discussed in Section 3.2.

E.1 Taxonomy Induction Setup

The construction of $\mathcal{T}_{cap}$ follows a hybrid approach, initializing with top-down architectural constraints and refining via bottom-up data induction.

Data Sampling and Stability To ensure the taxonomy is representative yet computationally efficient to induce, we constructed a balanced subset $\mathcal{D}_{sub} \subset \tilde{\mathcal{D}}$ containing approximately 260 instances. This subset was uniformly sampled across different

Prompt: Fine-Grained Checkpoint Scoring

```
#### System Prompt
Please act as an impartial evaluator. Your task is to score the assistant's answer using the
evaluation criteria provided in the Evaluation System below.

#### Instructions
For each item in the Evaluation System:
1. Read the criterion and the corresponding ground_truth.
2. Compare the assistant's answer with the ground_truth.
3. Give a score from 1 to 5 based on the following rubric:
     5 - Excellent: Fully aligned with ground_truth; accurate, detailed, and well-reasoned.
     4 - Good: Mostly aligned; minor omissions or small inaccuracies.
     3 - Adequate: Partially aligned; shallow coverage or notable gaps.
     2 - Poor: Major missing elements or noticeable misunderstandings.
     1 - Failing: Incorrect, irrelevant, or contradicts the ground_truth.

#### Output Format
<The Start of Evaluation Result>
<Understanding><Scene Understanding><Scene Classification> | score: [2] | Weight 3
<Perception><Fine-grained Perception><Attribute Recognition> | score: [5] | Weight 3
<Perception><Spatial Perception><Object Relationship Recognition> | score: [1] | Weight 4
<The End of Evaluation Result>

Do not add extra categories.
Do not modify or interpret the capability tags.
Evaluate strictly according to the criteria and ground_truth.
Do not explain the scoring reasons; directly output the score in the specified format.
Every score MUST be written strictly as: score: [x]
 - The brackets [ ] are mandatory.
 - No other formats like score: x or score:[x] or score: x] are allowed.

#### Input Data
<question> {prompt_text} </question>
<evaluation_system> {eval_system} </evaluation_system>
<assistant's_answer> {ans} </assistant's_answer>
```

Figure 10: The system prompt used for scoring the model's response against the fine-grained checkpoints.

data sources to maintain diversity. To verify the sufficiency of this sample size, we performed multiple rounds of resampling with consistent sizes; the resulting tree structures demonstrated negligible variance in leaf node distribution, confirming the stability of the induction process.

Judge Model and Parameters We employ a generic SOTA LLM as the Judge Model to drive the proposal and merging phases. The induction process is controlled by the following parameters:

- **Support Threshold**: During the pruning phase, any leaf node supported by fewer than $N = 5$ atomic criteria is considered an outlier or overly task-specific. These nodes are merged into their nearest semantic neighbors or pruned.
- **Structure Hierarchy**: We enforce a strict four-layer hierarchy: *Root* $\rightarrow$ *Category* $\rightarrow$ *Parent* $\rightarrow$ *Capability (Leaf)*.
- **Iterative Refinement**: The taxonomy is refined over fixed iterations. In each step, the Judge Model clusters nodes with semantic redundancy (e.g., merging synonyms) to ensure a compact representation.

E.2 Taxonomy Structure Visualization

Figure 12 illustrates the final induced taxonomy $\mathcal{T}^*_{cap}$, showcasing the hierarchical organization of the six root anchors into fine-grained capability requirements.

E.3 Prompt Templates

The prompt used for the Proposal phase (Cap-Worker), which identifies gaps in the existing tree and proposes new nodes, is detailed in Figure 11.

F Extended Analysis of Taxonomy Validity

Prompt: Capability Proposal (Cap-Worker)

System Prompt

You are an expert in knowledge taxonomy and capability analysis. Your primary function is to analyze reasoning criteria against the **Capability Tree (CapT)** and propose additions where gaps are identified. A **Proposal** is a structured command in the format <ADD,path,name> used to suggest the creation of a new node in the CapT.

Instructions

1. **Reuse Existing Capability (Top Priority)**: Traverse the CapT to find the most relevant existing leaf node that encapsulates the skill described. Try your best to reuse existing nodes.
2. **Proposal Formulation**: If no match exists, name a new capability node representing a general, reusable skill.
3. **Hierarchy Rules**: The CapT has a strict four-layer hierarchy: root > category > parent > capability.
4. **Node Creation**: If no suitable parent exists, propose a new parent first under an appropriate category.

Output Format

Output your proposals in the following XML-style:

<analysis> [Your reasoning process] </analysis>
<proposal> <ADD,[path],[name]> </proposal>

Input Data

<question> {prompt_text} </question>
<criterion> {criterion_list} </criterion>
<CapT> {cap_tree} </CapT>

Figure 11: The system prompt for the Candidate Proposal phase in taxonomy construction.

F.1 Detailed Metric Definitions

To quantify the homogeneity of a capability node, we use the following Leave-One-Out (LOO) metrics:

- **Global MAE (MAE_G):** For a given model M and an atom criterion a, we predict $M(a)$ as the average score of M across all atoms in the dataset excluding a. This represents a structure-agnostic baseline.
- **Intra-Node MAE (MAE_I):** We predict $M(a)$ as the average score of M across only those atoms $a' \in \mathcal{N}(a)$ where $\mathcal{N}(a)$ is the set of atoms sharing the same leaf node as a.

The gain Δ measures the reduction in prediction uncertainty provided by the taxonomy. A negative Δ suggests that the intra-node variance is higher than the global variance, indicating a poor clustering of capabilities.

F.2 Baseline Descriptions

The taxonomies compared in Table 3 are defined as follows:

- **Handcraft:** A manually defined hierarchy based on traditional CV task categories.
- **Generalist:** A broad taxonomy used in general-purpose LLM evaluations, adapted for multi-modal tasks.
- **Scan (Vis.):** A structure derived solely from visual similarity of the input images using clustering algorithms.
- **No Data:** A data-blind tree generated by an LLM using only the text descriptions of criteria, without considering actual model performance.

F.3 Pruning Efficiency and Robustness

Our pruning pipeline serves two purposes: reducing redundancy and improving generalization. As observed in Section 4.2, `Ours (Final)` is more compact than `Ours (No Prune)`. The decrease in MAE_I (from 18.90 to 18.83) after pruning indicates that several small, fragmented nodes in the unpruned version were likely capturing noise. By merging these into statistically significant clusters, we achieve a more robust representation of model capabilities. Detailed leaf-node merging logs and the final tree visualization are available in our repository.

G Human Meta-Evaluation and Improvement Attribution

G.1 Model Fingerprinting Experimental Protocol

To ensure a rigorous validation of report distinctiveness, our blinded matching experiment followed these steps:

- **Anonymized Diagnostic Reports:** We provided qualitative reports generated by `ATOM`, masking all model identities. These reports highlight failure patterns (e.g., "struggles with spatial relations in low-contrast images").
- **Ground-truth Behavioral Evidence:** Annotators were given 30–50 random raw responses from the two models (M_A, M_B) being compared.
- **The Task:** Annotators had to link each report to the model that most likely produced the observed behaviors. Each pair was evaluated by 10 independent annotators.

The high accuracy observed across architectures confirms that the judge model is not generating generic templates but is sensitive to the specific error distributions of the target model.

G.2 Sub-Capability Gain Breakdown

The improvement from 1.82 to 3.58 for `InternVL3.5` (as mentioned in Section 4.3) is not uniform across all tasks. Our analysis reveals that:

- **Reasoning tasks:** Saw the highest gain ($+1.92$), as the atom-level checkpoints forced the model to re-examine visual evidence it previously ignored.
- **Trustworthiness:** Improved because the diagnostic feedback specifically flagged hallucinated entities, causing the model to adopt a more conservative and grounded generation strategy.
- **Understanding scores:** Showed that while the model had the requisite knowledge, it often failed to "activate" it without the fine-grained diagnostic prompts provided by our framework.

G.3 Clarification on Self-Correction Attribution

To address potential concerns regarding whether the performance gains in Section 4.3 stem from genuine autonomous refinement or merely knowledge transfer from a stronger judge, we clarify the mechanics of our *Criterion-Aware Self-Refinement*:

- **Strict Information Masking:** In this strategy, we strictly mask the ground truth answer (g_t) and provide only the atomic question (q_t). This ensures no answer leakage occurs. The prompt acts purely as an *attention-guiding mechanism*, forcing the model to re-evaluate neglected visual evidence.
- **Internal Capability vs. External Help:** The significant performance gains under this strictly masked setting demonstrate that the candidate models often already possess the latent capability to solve the task, but initially fail due to "attention drift." Our diagnostic prompts simply reactivate this internal capability.
- **Validation of Diagnostic Precision:** Ultimately, the primary objective of our self-correction experiments is not merely to claim state-of-the-art refinement scores, but to validate the *effectiveness and precision* of the ATOM framework. The fact that a model can autonomously rectify its error when directed to a specific atomic failure proves that ATOM accurately pinpoints the true root cause of the hallucination.

H Diagnostic Analysis Assets

This section complements the *Diagnostic Analysis* methodology. We present the concrete statistical assets generated during the "saturate-and-select" phase. These visual artifacts serve as the quantitative inputs that enable detailed failure attribution and model fingerprinting.

We present the full outputs of ATOM's automated diagnostic pipeline, illustrating how raw evaluation data is synthesized into actionable qualitative insights. All qualitative diagnostic reports presented in this section were autonomously generated by Gemini-2.5-pro, serving as the Judge Model within our framework. The diagnostic generation operates at two distinct levels of granularity:

Comprehensive Analysis (Macro-Level). This analysis aggregates performance metrics across all 18 evaluated models to identify landscape-level trends. The quantitative foundation for this report(Figure 13 and Figure 14) is the Global Capability Heatmap (Figure 17), which visualizes the proficiency of all models across the full taxonomy constructed in Appendix E.

Model-Specific Profiling (Micro-Level). To demonstrate deep-dive diagnostics, we provide a case study on **Qwen3-VL-30B-A3B**. Unlike the macro analysis, this report is grounded in model-specific visual fingerprints (Figure 16), including radar charts and fine-grained skill bars. The resulting text report, shown in Figure 15, synthesizes these visual signals to critique the model's specific *Strengths/Weaknesses*, offering a detailed explanation for the performance gaps observed in the global ranking.

I Qualitative Analysis and Failure Modes

To further illustrate the practical value of ATOM and address the nuances of our framework, we provide a qualitative analysis focusing on two aspects: demonstrating "capability disentanglement" (mitigating the "right answer, wrong reason" phenomenon) and analyzing the failure modes of the ATOM pipeline itself.

Demonstrating Capability Disentanglement. Traditional holistic evaluations often suffer from "Outcome Bias." For instance, in a complex visual reasoning task, a model might correctly guess the final multiple-choice answer while hallucinating the relative positions of objects in its intermediate steps. A holistic judge typically rewards this spurious guess. In contrast, ATOM explicitly disentangles capabilities by evaluating atomic steps. In such a scenario, ATOM would score the *Visual Identification* atom as correct (5/5) but flag the *Spatial Relationship* atom as failed (1/5). This granularity proves that the model possesses basic perceptual skills but lacks reasoning capability, explicitly exposing the "right answer, wrong reason" flaw.

Failure Modes of the ATOM Framework. Despite its robust verification capabilities, we conducted a manual error analysis on ATOM's generation pipeline and identified two primary failure modes:

- **Granularity Mismatch:** Extremely subtle visual details might occasionally be overlooked by the Judge Model during the initial "Cognitive Replay" phase. Consequently, the framework may fail to extract atomic criteria for these fine-grained features, leading to an incomplete evaluation checklist.
- **Ambiguous Criteria:** In rare instances, the generated atomic criteria may be overly subjective or vaguely phrased, making it difficult for the scoring module to perform a deterministic verification. However, our human meta-evaluation on the "Necessity" dimension (detailed in Appendix C) demonstrates that such ambiguous cases are highly infrequent, occurring in less than 10% of the generated samples.

J Comparison with Other Evaluation Frameworks

While our main text discusses general MLLM-as-a-Judge paradigms, it is also important to contrast ATOM with other evaluation frameworks that employ fine-grained or checklist-based methods:

- **Prometheus-Vision (Lee et al., 2024):** This framework relies on fine-tuning specific open-source models to act as judges. In contrast, ATOM is a *training-free prompting framework* that can leverage any capable off-the-shelf MLLM without additional training costs.
- **MLLM-Bench (Ge et al., 2025) and Judge Anything (Pu et al., 2025):** While these frameworks also utilize per-sample criteria or checklists, they heavily rely on human annotation or human-in-the-loop curation, which limits their scalability. ATOM is fully automated, generating visually-anchored atomic criteria via the "Cognitive Replay" mechanism. Furthermore, rather than relying on fixed taxonomies for static scoring, ATOM builds a data-driven *Dynamic Capability Tree*. This structured diagnosis allows ATOM to move "Beyond Ranking" to support targeted, autonomous model self-correction.

K Computational Efficiency and Scalability

While the ATOM framework introduces a multi-stage pipeline, it is specifically architected to remain highly efficient and scalable for large-scale leaderboard evaluations. We clarify two key design

choices that significantly mitigate computational costs:

- **Dataset-Level Preprocessing ("Generate Once, Reuse Infinitely"):** The most compute-intensive phases (Cognitive Replay and Atomic Criteria Extraction) are executed as dataset-level preprocessing rather than per-model inference. These visually-anchored criteria are generated only *once* per sample and are fixed as a static benchmark reference. When evaluating a new candidate MLLM, these extraction stages are completely bypassed, requiring only the final scoring step.
- **Single-Pass Atomic Scoring:** Contrary to the potential concern that each atomic criterion requires a separate API call, ATOM employs a *batch prompting strategy*. As illustrated in Figure 10, all atomic criteria for a given sample are aggregated into a single system prompt. The Judge Model evaluates the candidate's response against all fine-grained checkpoints simultaneously within a *single inference call*, ensuring that the evaluation latency and cost remain comparable to traditional pointwise judges.

L Ethical Considerations and Human Evaluation

For the human meta-evaluation and diagnostic validation described in this paper, we recruited 12 human annotators. All participants had graduate-level training in Computer Science or related fields, ensuring they possessed the necessary expertise to serve as subject matter experts in MLLMs.

To facilitate rigorous and granular assessment, we employed a specialized annotation interface, as illustrated in Figure 18. This interface presented annotators with a unified view of the visual inputs, generated Chain-of-Thought (CoT) analyses, and atomic criteria, thereby streamlining the verification of ground truth and candidate comparisons.

Prior to the evaluation, all annotators were fully informed of the research's objectives, the specific tasks involved via the aforementioned interface, and the intended academic use of the collected data. They provided explicit informed consent for their contributions to be used in this research. To ensure fair treatment and ethical labor practices, all annotators were compensated with a competitive hourly wage, which is aligned with the local standard for research assistants in their region.

The evaluation process did not involve the collection of any personally identifiable information. Furthermore, we carefully screened the data samples to ensure they did not contain offensive, harmful, or sensitive content, thereby ensuring a safe working environment for the annotators.

Taxonomy Part I: Gen, Perc, Under

- Root
  - Generation
    - Code Generation
    - Content Composition
      - Narrative / Descriptive
      - Stylistic Generation
  - Perception
    - Fine-grained Perception
      - Action Recognition
      - Attribute Recognition
      - Emotion Recognition
      - Object Counting
      - Object Part Recognition
      - Object Recognition
      - Shape Recognition
      - Style Recognition
    - Spatial Perception
      - Depth Perception
      - Obj Relationship Recog
    - Text Perception
      - Text Recognition
  - Understanding
    - Contextual Understanding
      - Convention Understanding
      - Thematic Interpretation
    - Language Understanding
      - Ambiguity Resolution
      - Information Extraction
      - Reference Resolution
      - Semantic Similarity
      - Translation
    - Scene Understanding
      - Scene Classification
    - Semantic Interpretation
      - Symbol Interpretation

Taxonomy Part II: Reason, Know, Trust

- Root
  - Reasoning
    - Comparative Reasoning
      - Alternative Categorization
      - Comparative Analysis
      - Info Consistency Eval
    - Hypothetical Reasoning
      - Hypothesis Formulation
      - Outcome Prediction
      - Plausibility Assessment
    - Inferential Reasoning
      - Causal/Functional Reason
      - Information Synthesis
      - Intent Inference
      - Part-Whole Inference
      - State Inference
      - Taxonomic Classification
      - Temporal Inference
    - Logical Reasoning
      - Deductive Reasoning
    - Mathematical Reasoning
      - Arithmetic Calculation
    - Planning
      - Action Selection
      - Goal Decomposition
      - Prerequisite Analysis
    - Procedural Reasoning
      - Procedural Step Analysis
    - Spatial Reasoning
      - Geometric Property Analysis
  - Knowledge
    - Factual Knowledge
      - Commonsense Knowledge
      - Encyclopedic Knowledge
      - Terminology & Defs
    - Knowledge Retrieval
      - Doc Info Retrieval
      - Table Lookup
  - Trustworthiness
    - Information Integrity
      - Synthetic Content Detect
      - Tampering Detection

Figure 12: Visualization of the Dynamic Capability Taxonomy ($\mathcal{T}^*_{cap}$). To balance the visualization structure, the tree is split into two columns. Left: Generation, Perception, and Understanding. Right: Reasoning, Knowledge, and Trustworthiness.

Table 6: Detailed list of all tasks and skills (meta-tasks) under image and comprehension.

Task Description					Data	
Skill (Meta-Task)	**#**	**Task Short Name**	**Domain**	**Capability**	**Data Source**	**Number**
#1 Behavior Recognition	1	Sports image classification	Sports	Content Recognition, Commonsense Know.	Kaggle, 2023a	50
#2 Code Generation	1	Sketch2html code generation	Code	Creativity, Reasoning Ability	Kaggle, 2022b	50
#3 Disease Recognition	1	Fruit disease recognition	General	Content Recognition	Kaggle, 2025b	50
	2	Abnormal Heartbeat Patients	Medicine	Content Recognition	Kaggle, 2025a	50
	3	Pumpkin Leaf Diseases Recognition	Biology	Content Recognition	Kaggle, 2024a	50
	4	Leukemia Classification	Medicine	Content Recognition	Kaggle, 2019b	7
#4 Document Visual Question Answering	1	Multi-Page Letter Document VQA	General	Content Recognition	Tito et al., 2023	50
	2	Multi-Page Form Document VQA	General	Content Recognition	Tito et al., 2023	50
	3	Multi-Page Poster Document VQA	General	Content Recognition	Tito et al., 2023	50
	4	Multi-Page Infographic Document VQA	General	Content Recognition	Tito et al., 2023	50
#5 Emotion Detection	1	Face emotion recognition	General	Content Recognition, Commonsense Know.	Kaggle, 2023d	50
	2	Emotion detection fer	Humanities	Affective Analysis	Kaggle, 2020b	50
#6 Hallucination Detection	1	Object Hallucination Detection	General, Ethics	Content Recognition	Liu et al., 2025	50
#7 Image Captioning	1	Satellite image caption	General	Content Recog., Reasoning Ability	Lu et al., 2017	50
	2	Instagram image caption	Humanities, Social	Content Recog., Reasoning Ability	Kaggle, 2020c	50
	3	Astronomy image captioning	Astronomy	Content Recog., Reasoning Ability	Alam et al., 2024	50
#8 Image OCR	1	Letter and number ocr	General	Content Recognition	Kaggle, 2021c	50
	2	License plate ocr	General	Content Recognition	Kaggle, 2020a	50
#9 Image Recognition	1	Latex Code Recognition	Code	Content Recognition	Blecher, 2022	50
#10 Image Visual Question Answering	1	Culture Image VQA	Culture	Content Recog., Reasoning Ability	Zhang et al., 2025a	50
	2	Geography VQA	Geography	Reasoning Ability	Lu et al., 2022	50
	3	Chemistry VQA	Chemistry	Reasoning Ability	Lu et al., 2022	44
	4	Mathematical statistical reasoning VQA	Mathematics	Reasoning Ability	Lu et al., 2023	46
	5	Medical image QA	Medicine	Commonsense Know.	Kaggle, 2023e	50
	6	Sporting-related VQA	Sports	Content Recognition	Li et al., 2022	50
	7	Art Image Visual Question Answering	Art	Content Recog., Reasoning Ability	Garcia et al., 2020	50
	8	Webpage Content VQA	General	Content Recognition	Chang et al., 2022	50
	9	Reasoning and compositional image QA	General	Content Recog., Reasoning Ability	Hudson and Manning, 2019	50
	10	Remote sense VQA	Earth	Reasoning Ability	Wang et al., 2024	50
	11	Physics VQA	Physics	Reasoning Ability	Lu et al., 2022	50
	12	Education-related VQA	Education	Content Recognition	Kembhavi et al., 2017	50
	13	Historical MultiSource Dialogue	General	Content Recognition	Li et al., 2022	50
	14	Financial Dialogue VQA	Finance	Content Recognition	Li et al., 2022	50
#11 Multi-image VQA	1	Digital Consistency Comparison	General	Reasoning Ability	LeCun	50
	2	Pose and Activity Consistency	General	Action Affective Analysis	Li et al., 2014	50
#12 Multimodal Dialogue	1	Environment Based Next action Description	General	Interactive Capability	Shridhar et al., 2020	50
	2	Comic Dialogue Completion	Art	Interactive Capability	Iyyer et al., 2017	50
	3	Egocentric Daily Tasks Action Planning	General	Planning Ability, Interactive Capability	Shridhar et al., 2020	50
#13 Multimodal Reasoning	1	Cloth Color Visual Question Answering	Fashion	Reasoning Ability	Han et al., 2017	50
	2	Visual Step Matching Reasoning	General	Reasoning Ability	Yagcioglu et al., 2018	50
	3	ComicPanel VQA	Art	Reasoning Ability	Iyyer et al., 2017	50
#14 Object Counting	1	Seed counting	General	Content Recog., Reasoning Ability	Kaggle, 2023f	50
	2	Vehicle counting	General	Content Recog., Reasoning Ability	Hsieh et al., 2017	50
#15 Multimodal Neural Translation	1	Multimodal Neural Translation	Linguistics	Reasoning Ability	Chumpu	50

... continued from previous page

Skill (Meta-Task)	#	Task Short Name	Domain	Capability	Data Source	Number
#16 Object Recognition	1	Infrared Thermal Image Classification	Infrared	Content Recognition	Kaggle, 2024b	50
	2	Image style classification	Art	Content Recognition	Han et al., 2025	50
	3	Famous iconic women	Humanities, History	Commonsense Know.	Kaggle, 2021b	50
	4	Animal Recognition	Animal	Content Recognition	Kaggle, 2023b	50
	5	Deep fake detection	Ethics	Content Recognition	Yan et al., 2024	50
	6	Vegetable recognition	General	Content Recog., Commonsense Know.	Kaggle, 2021a	50
	7	Microorganism image classification	Biology	Content Recognition	Kaggle, 2022a	50
	8	Dog breeds recognition	General	Content Recognition	Kaggle, 2023c	50
#17 Pose Recognition	1	Yoga pose classification	Sports	Content Recognition	Kaggle, 2022c	50
#18 Relation Reasoning	1	Classification of Visual Spatial Relationship	General	Content Recog., Spatial Perception	Zhao et al., 2022	50
	2	Description of Single Spatial Relationship	General	Content Recog., Spatial Perception	Zhao et al., 2022	50
#19 Ripeness Recognition	1	Strawberry ripeness recognition	General	Content Recognition	Kaggle, 2023g	50
#20 Scene Graph Gen.	1	Image Scene Graph Parsing	General	Reasoning Ability	Krishna et al., 2017	50
#21 Scene Recognition	1	Terrain recognition	Nature	Content Recognition	Kaggle, 2024c	50
#22 Sign Language Recognition	1	American Sign Language Recognition	Linguistics	Content Recognition	Kaggle, 2019a	50
#23 Visual Relation Inference	1	Before After Relationship Caption	General	Content Recognition	Bodur et al., 2024	50
	2	Bird Variation Comparison Description	Biology	Content Recognition	Forbes et al., 2019	50
	3	Multi Image Alteration Description	General	Content Recognition	Johnson et al., 2017	50
#24 Visual Storytelling	1	Cartoon Story Telling	General	Commonsense Underst.	Li et al., 2019	50
	2	Sequential Story Completion	Humanities	Commonsense Underst.	Maharana et al., 2022	50
	3	Multi-image Next frame Description	General	Commonsense Underst.	Gupta et al., 2018	50

ATOM Diagnostic Report: Comprehensive Analysis (Page 1/2)

Overview

Relative Performance

The **Model Rank Distribution** reveals that the **Gemini-2.5-pro** model dominates the highest ranks, achieving first place in **62.52%** of the samples, significantly outperforming all other models. The **Qwen3-VL-235B-A22B** model follows with a strong presence in the top ranks, securing first place in **29.19%** of the samples. In contrast, smaller models like **InternVL3.5-1B** shows lower frequencies in the top ranks, indicating weaker overall performance compared to larger models.

The **Top Tie** highlights that tie patterns are relatively rare, with no single pattern occurring more than 5% of the time. This suggests that most models exhibit distinct performance levels, allowing for clear differentiation in rankings. However, the presence of ties indicates that some models perform similarly in certain contexts, particularly among mid-sized models like **Qwen3-VL-8B** and **Qwen3-VL-30B-A3B**, which occasionally share the top rank.

Overall, the ranking pattern demonstrates that larger models, especially **Gemini-2.5-pro** and **Qwen3-VL-235B-A22B**, consistently achieve higher ranks, indicating superior relative capabilities. Smaller models struggle to compete in the top positions, suggesting a significant performance gap between large and small models.

Absolutely Capability

The **Top-Level Score** provides insights into the absolute capabilities of each model across various top-level skill categories. The **Gemini-2.5-pro** model excels in most categories, achieving the highest average scores across all dimensions, including **Generation, Knowledge, Perception, Reasoning,** and **Trustworthiness**. This indicates that the model performs exceptionally well across a wide range of tasks.

In contrast, the **Qwen3-VL-235B-A22B** model shows strong performance in **Understanding** and **Trustworthiness**, but its scores in other categories are lower compared to **Gemini-2.5-pro**. The **Qwen3-30B-A3B** model also performs well, particularly in **Reasoning** and **Generation**, but it lags behind in **Perception** and **Knowledge**.

Smaller models exhibit lower average scores across all categories, indicating weaker overall capabilities. These models struggle to match the performance of larger models, particularly in complex tasks requiring advanced reasoning and perception skills.

In summary, **Gemini-2.5-pro** demonstrates near-perfect performance across all tasks, while **Qwen3-VL-235B-A22B** excels in specific areas like Understanding and Trustworthiness. Smaller models, however, fall short in most categories, highlighting the importance of model size in achieving high absolute capabilities.

Figure 13: **Comprehensive Diagnostic Report generated by ATOM.** This automatically synthesized report identifies macro-level. (Continued on next page)

ATOM Diagnostic Report: Comprehensive Analysis (Page 2/2)

Model Profiling

Based on the analysis of the provided data, the following key characteristics emerge for each model:

- **Gemini-2.5-pro:** This model stands out as the most capable, excelling in all top-level skill categories. Its defining strength lies in its versatility and robust performance across diverse tasks, particularly in **Generation, Reasoning,** and **Understanding**. The model's ability to maintain high scores in multiple domains makes it a top choice for complex, multi-faceted tasks.
- **Qwen3-VL-235B-A22B:** This model excels in **Understanding** and **Trustworthiness**, making it ideal for tasks requiring nuanced comprehension and secure content detection. However, it lags behind in **Generation** and **Reasoning**, indicating a specialization in understanding-based tasks rather than creative or analytical ones.
- **Qwen3-VL-30B-A3B:** While not as dominant as **Qwen3-VL-235B-A22B**, this model demonstrates strong performance in **Reasoning** and **Generation**, particularly in narrative and deductive tasks. However, it struggles in **Perception** and **Knowledge**, indicating a specialization in language-related tasks rather than spatial or factual recall.
- **Qwen3-VL-8B:** This model shows moderate performance across most categories but lacks standout strengths. It performs reasonably well in **Generation** and **Understanding** but falls short in **Reasoning** and **Perception**, suggesting it is best suited for general-purpose tasks rather than specialized ones.
- **InternVL3.5 Models (1B to 38B):** These models exhibit progressively better performance as their size increases, but they still lag behind the larger models like **Qwen3-VL-235B-A22B** and **gemini**. Their defining characteristic is limited capability in complex reasoning and perception tasks, making them suitable for simpler, less demanding applications.
- **LLaVA Models (4B to 72B):** These models show moderate performance in **Understanding** and **Generation** but struggle in **Reasoning** and **Perception**. Their defining characteristic is a focus on visual understanding tasks, but they lack the depth required for advanced reasoning and spatial perception.

In conclusion, the **Gemini-2.5-pro** model is the most versatile and capable, while other models specialize in specific areas such as **Understanding, Generation,** or **Trustworthiness**. The size of the model plays a crucial role in determining its absolute capabilities, with larger models generally outperforming smaller ones.

Figure 14: **Comprehensive Diagnostic Report generated by ATOM.** This automatically synthesized report identifies macro-level.

ATOM Diagnostic Report: Model Specific Profiling (Case Study: Qwen3-VL-30B-A3B)

Model Specific: Qwen3-VL-30B-A3B

As illustrated in Figure 16, the radar and bar charts for **Qwen3-VL-30B-A3B** provide a comprehensive view of the model's performance across various skills.

Strengths/Weaknesses

The radar charts for **Qwen3-VL-30B-A3B** provide a comprehensive view of the model's performance across various skills, categorized under **Perception & Comprehension** and **Reasoning & Generation**.

In the **Perception & Comprehension** category, the model demonstrates strong capabilities in areas such as **Action Recognition (3.56)**, **Emotion Recognition (2.98)**, and **Object Recognition (2.91)**. However, it shows notable weaknesses in **Depth Perception (1.85)** and **Object Relationship Recognition (2.37)**, indicating potential gaps in understanding spatial relationships and object interactions.

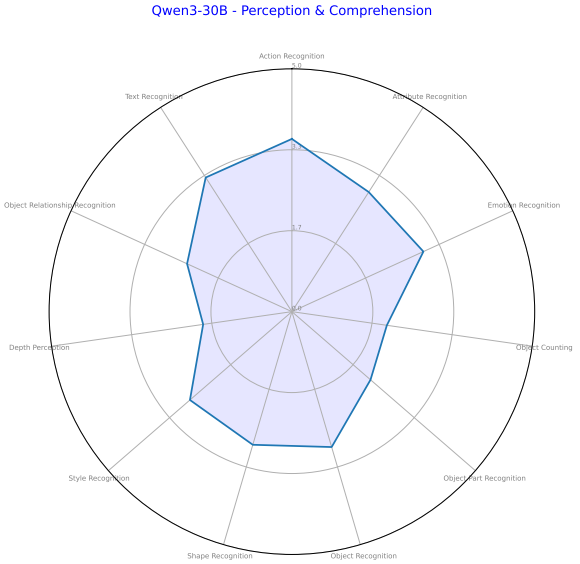
In the **Reasoning & Generation** category, the model excels in tasks like **Information Consistency Evaluation (3.23)** and **Deductive Reasoning (3.58)**, suggesting robust reasoning abilities. Conversely, it struggles with **Alternative Categorization (1.15)** and **Hypothesis Formulation (1.25)**, highlighting areas where the model may need improvement in creative and analytical reasoning.

The bar chart further elaborates on these strengths and weaknesses by presenting average normalized scores for each bottom-level skill. The model performs exceptionally well in **Semantic Similarity (4.67)**, **Information Extraction (4.49)**, and **Translation (4.33)**, which are critical for understanding and generating human-like language.

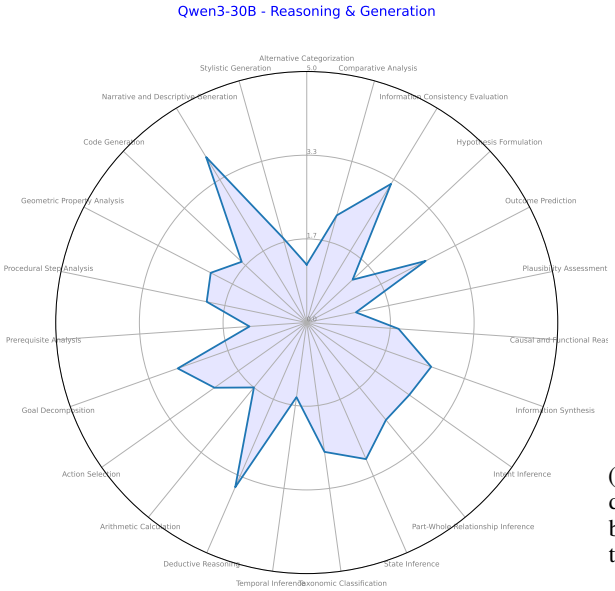
On the other hand, skills like **Plausibility Assessment (1.00)** and **Prerequisite Analysis (1.14)** show significant room for improvement, indicating challenges in assessing logical consistency and breaking down complex tasks into manageable steps.

The best-performing skills include **Semantic Similarity, Information Extraction, and Translation**, reflecting the model's proficiency in understanding and generating nuanced language. In contrast, the worst-performing skills are **Plausibility Assessment, Prerequisite Analysis, and Alternative Categorization**, suggesting that the model may struggle with tasks requiring creative categorization and logical assessment.

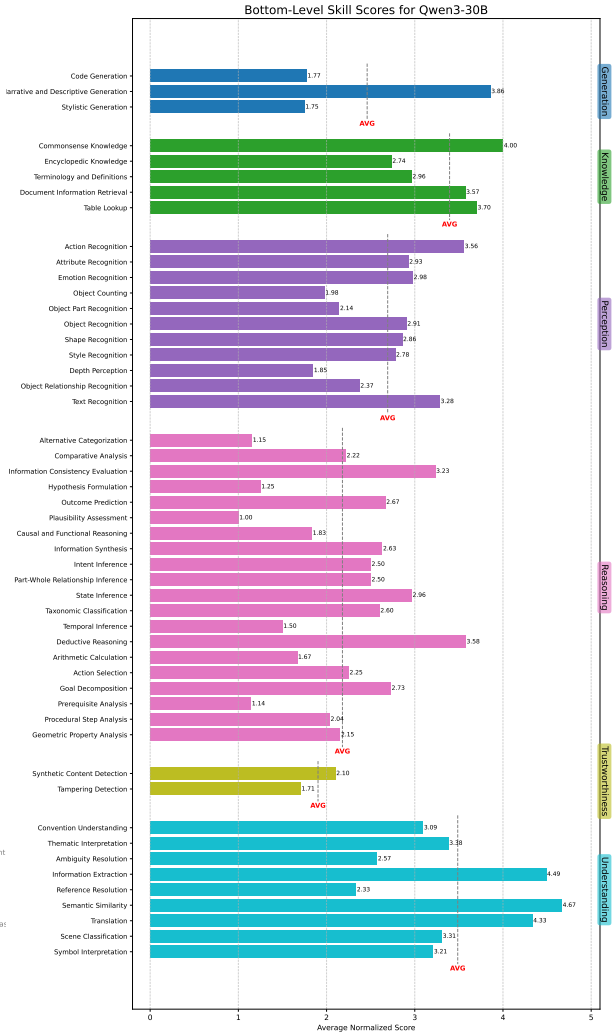
Figure 15: **Comprehensive Diagnostic Report generated by ATOM.** This automatically synthesized report identifies model-specific profiling.



(a) **Radar I: Perception & Comprehension.**



(b) **Radar II: Reasoning & Generation.**



(c) **Fine-Grained Skill Bar (Qwen3-VL-30B-A3B).** A deep-dive diagnostic view isolating atomic skill performance. Red bars indicate significant deviations ($<$ avg) flagged by the system.

Figure 16: **Model-Specific Diagnostic Fingerprints (Case Study: Qwen3-VL-30B-A3B).** (a) & (b) Radar charts visualize the model's structural biases across four major dimensions (Perception, Comprehension, Reasoning, Generation). (c) The granular bar chart serves as the source for specific failure attribution. The Judge Model synthesizes these visual anomalies (e.g., the sharp drop in specific perception sub-tasks shown in the bar chart) to generate the final qualitative report.

Skill Name	InternVL-1B	InternVL-2B	InternVL-4B	InternVL-8B	InternVL-38B	InternVL-241B	Qwen3-8B	Qwen3-30B	Qwen3-235B	Qwen2.5-72B	Phi-3.5-Vis	LLaVA-Phi3	LLaVA-v1.6-7B	LLaVA-v1.6-Mistra	LLaVA-v1.6-13B	LLaVA-v1.6-34B	LLaVA-OV-72B	Gemini-2.5-Pro
● 1. Generation	1.127	1.822	1.867	2.029	2.304	2.016	2.083	2.288	2.173	2.032	1.216	1.04	1.255	1.277	1.107	1.438	1.793	4.08
■ 1.1. Code Generation	1.03	1.875	1.877	1.897	1.616	2.094	2.14	1.773	1.847	2.063	1.03	1.08	1.09	1.0	1.0	1.627	1.827	3.84
○ 1.2. Content Composition	1.223	1.768	1.857	2.161	2.991	1.938	2.027	2.804	2.5	2.0	1.402	1.0	1.42	1.554	1.214	1.25	1.759	4.321
■ 1.2.1. Narrative and Descriptive Generation	1.571	2.286	1.714	2.571	3.857	2.0	2.429	3.857	2.0	2.0	1.429	1.0	1.714	1.857	1.429	1.0	2.143	4.143
■ 1.2.2. Stylistic Generation	0.875	1.25	2.0	1.75	2.125	1.875	1.625	1.75	3.0	2.0	1.375	1.0	1.125	1.25	1.0	1.5	1.375	4.5
● 2. Knowledge	2.256	2.324	2.618	2.861	2.793	3.267	3.179	3.435	3.48	3.449	2.108	1.555	1.585	1.961	1.744	2.188	2.892	4.29
○ 2.1. Factual Knowledge	1.925	1.994	2.348	2.624	2.698	3.063	2.761	3.233	3.401	3.29	1.792	1.638	1.578	2.18	1.511	2.346	2.746	4.085
■ 2.1.1. Commonsense Knowledge	2.0	1.4	2.2	3.0	2.6	3.8	2.8	4.0	4.0	4.0	1.4	1.4	1.4	2.8	1.4	3.0	2.8	4.2
■ 2.1.2. Encyclopedic Knowledge	1.733	2.049	2.211	2.119	2.3	2.349	2.379	2.736	2.852	2.511	1.843	1.64	1.654	1.819	1.621	2.122	2.387	4.015
■ 2.1.3. Terminology and Definitions	2.043	2.533	2.633	2.753	3.193	3.04	3.103	2.963	3.35	3.36	2.133	1.873	1.68	1.92	1.513	1.917	3.05	4.04
○ 2.2. Knowledge Retrieval	2.587	2.654	2.888	3.099	2.887	3.471	3.598	3.637	3.56	3.607	2.423	1.473	1.593	1.743	1.977	2.029	3.039	4.496
■ 2.2.1. Document Information Retrieval	2.731	2.497	2.775	2.828	2.943	2.866	3.195	3.572	3.241	3.155	2.063	1.71	1.539	1.986	1.987	2.056	2.884	4.328
■ 2.2.2. Table Lookup	2.444	2.811	3.002	3.37	2.832	4.076	4.0	3.702	3.878	4.059	2.784	1.235	1.647	1.5	1.966	2.002	3.194	4.664
● 3. Perception	2.135	2.406	2.556	2.49	2.757	2.695	2.834	2.719	3.244	2.689	2.044	1.681	1.763	2.109	2.005	2.227	2.686	3.789
○ 3.1. Fine-grained Perception	1.903	2.013	2.321	2.271	2.451	2.708	2.852	2.765	3.208	2.48	1.758	1.726	1.762	1.822	1.869	1.971	2.597	3.633
■ 3.1.1. Action Recognition	1.928	2.353	2.617	2.585	2.981	3.394	3.34	3.557	3.862	3.194	2.058	1.798	1.853	2.219	2.313	2.326	3.189	3.604
■ 3.1.2. Attribute Recognition	1.88	2.167	2.303	2.374	2.312	2.635	3.037	2.927	3.323	2.633	1.871	1.706	1.661	1.832	1.822	1.919	2.494	3.707
■ 3.1.3. Emotion Recognition	2.417	2.525	3.161	2.795	2.869	3.239	3.564	2.978	3.516	2.09	1.83	2.067	2.304	2.48	2.261	2.43	2.835	4.622
■ 3.1.4. Object Counting	1.424	1.763	1.615	1.885	1.993	2.07	1.96	1.976	2.178	2.022	1.184	1.457	1.066	1.184	1.316	1.158	1.932	2.573
■ 3.1.5. Object Part Recognition	1.615	1.5	1.832	1.596	1.654	2.024	2.269	2.142	2.774	1.865	1.301	1.231	1.417	1.412	1.5	1.655	2.194	3.846
■ 3.1.6. Object Recognition	2.03	2.084	2.328	2.417	2.413	2.63	2.94	2.906	3.233	2.689	1.988	1.929	1.936	1.991	1.947	2.295	2.544	3.836
■ 3.1.7. Shape Recognition	2.0	1.679	2.25	2.179	2.5	2.821	2.929	2.857	3.071	2.607	1.643	1.75	1.75	1.714	1.536	2.018	3.107	3.286
■ 3.1.8. Style Recognition	1.926	2.037	2.463	2.333	2.889	2.852	2.778	2.778	3.704	2.741	2.185	1.87	2.111	1.741	2.259	1.963	2.481	3.593
○ 3.2. Spatial Perception	1.668	1.923	2.361	2.072	2.628	1.961	2.444	2.11	3.064	2.297	2.034	1.695	1.803	2.133	2.229	2.656	2.403	3.513
■ 3.2.1. Depth Perception	1.538	1.846	2.538	1.923	2.769	1.692	2.462	1.846	3.308	2.538	2.385	2.0	1.923	2.154	2.692	3.308	2.308	3.154
■ 3.2.2. Object Relationship Recognition	1.797	2.0	2.183	2.22	2.486	2.23	2.426	2.373	2.821	2.055	1.683	1.391	1.682	2.112	1.765	2.005	2.498	3.873
○ 3.3. Text Perception	2.833	3.283	2.986	3.126	3.192	3.414	3.207	3.283	3.461	3.289	2.341	1.621	1.725	2.374	1.918	2.053	3.057	4.22
■ 3.3.1. Text Recognition	2.833	3.283	2.986	3.126	3.192	3.414	3.207	3.283	3.461	3.289	2.341	1.621	1.725	2.374	1.918	2.053	3.057	4.22
● 4. Reasoning	1.379	1.761	1.999	1.928	2.057	2.263	2.332	2.209	2.757	2.136	1.563	1.353	1.33	1.347	1.372	1.555	2.026	2.919
○ 4.1. Comparative Reasoning	1.403	1.591	2.006	1.939	1.982	2.156	2.554	2.201	2.789	2.302	1.67	1.38	1.17	1.313	1.397	1.563	2.042	2.749
■ 4.1.1. Alternative Categorization	1.0	1.077	1.154	1.385	1.231	1.231	1.769	1.154	1.769	1.308	1.385	1.231	1.0	1.0	1.462	1.385	1.308	2.077
■ 4.1.2. Comparative Analysis	1.496	1.896	1.982	2.094	2.077	2.314	2.543	2.217	2.959	2.464	1.579	1.316	1.173	1.196	1.16	1.425	2.054	2.889
■ 4.1.3. Information Consistency Evaluation	1.712	1.8	2.883	2.337	2.637	2.922	3.349	3.232	3.639	3.135	2.046	1.593	1.337	1.742	1.569	1.878	2.764	3.282
○ 4.2. Hypothetical Reasoning	1.0	1.667	1.694	1.667	1.222	1.528	1.167	1.639	1.917	1.611	1.167	1.25	1.333	1.389	1.167	1.333	1.444	2.361
■ 4.2.1. Hypothesis Formulation	1.0	1.5	1.25	1.5	1.5	1.75	1.5	1.25	1.75	2.0	1.5	1.25	1.5	1.5	1.5	1.0	1.0	2.25
■ 4.2.2. Outcome Prediction	1.0	2.0	2.333	2.0	1.167	1.833	1.0	2.667	3.0	1.833	1.0	1.0	1.5	1.667	1.0	2.0	1.833	2.833
■ 4.2.3. Plausibility Assessment	1.0	1.5	1.5	1.5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.5	1.0	1.0	1.0	1.0	1.5	2.0
○ 4.3. Inferential Reasoning	1.619	1.661	1.72	1.894	2.006	2.064	2.387	2.361	2.556	2.343	1.643	1.432	1.587	1.66	1.795	1.682	2.016	3.587
■ 4.3.1. Causal and Functional Reasoning	1.237	1.226	1.479	1.667	1.767	1.856	2.124	1.833	2.201	1.744	1.37	1.096	1.201	1.134	1.288	1.456	1.931	3.619
■ 4.3.2. Information Synthesis	1.997	2.211	2.333	2.388	2.651	2.68	2.753	2.63	2.99	2.633	1.792	1.375	1.482	1.88	1.562	1.917	2.427	3.622
■ 4.3.3. Intent Inference	2.25	1.5	1.5	2.0	2.25	1.5	2.5	2.5	3.25	2.75	2.0	1.75	2.5	2.75	2.5	2.0	1.75	4.0
■ 4.3.4. Part-Whole Relationship Inference	1.167	1.417	1.417	1.833	1.333	1.667	2.667	2.5	2.083	2.333	1.333	1.25	1.083	1.083	1.833	1.667	2.167	3.0
■ 4.3.5. State Inference	1.928	2.398	2.208	2.455	2.848	2.582	3.128	2.965	3.448	2.908	1.98	1.668	1.94	2.118	2.02	1.97	2.06	3.4
■ 4.3.6. Taxonomic Classification	1.754	1.877	2.1	1.915	2.192	2.162	2.538	2.6	2.923	2.531	2.023	1.385	1.9	1.654	1.862	1.762	2.277	3.969
■ 4.3.7. Temporal Inference	1.0	1.0	1.0	1.0	1.0	2.0	1.0	1.5	1.0	1.5	1.0	1.5	1.0	1.0	1.5	1.0	1.5	3.5
○ 4.4. Logical Reasoning	1.885	3.077	3.538	3.308	3.077	3.308	4.0	3.577	4.308	3.538	2.0	1.462	1.654	1.577	1.462	2.077	3.192	4.154
■ 4.4.1. Deductive Reasoning	1.885	3.077	3.538	3.308	3.077	3.308	4.0	3.577	4.308	3.538	2.0	1.462	1.654	1.577	1.462	2.077	3.192	4.154
○ 4.5. Mathematical Reasoning	1.167	1.278	1.444	1.222	1.222	1.5	1.444	1.667	1.667	1.389	1.167	1.056	1.167	1.111	1.222	1.056	1.389	2.222
■ 4.5.1. Arithmetic Calculation	1.167	1.278	1.444	1.222	1.222	1.5	1.444	1.667	1.667	1.389	1.167	1.056	1.167	1.111	1.222	1.056	1.389	2.222
○ 4.6. Planning	1.295	1.703	2.324	2.218	2.534	2.219	2.053	2.04	2.86	2.463	1.981	1.608	1.3	1.728	1.156	1.712	2.135	2.937
■ 4.6.1. Action Selection	1.1	1.829	2.2	2.35	2.479	2.229	2.236	2.25	2.729	2.35	1.379	1.2	1.4	1.45	1.15	1.5	1.536	3.429
■ 4.6.2. Goal Decomposition	1.5	2.136	2.773	2.591	3.409	3.0	2.636	2.727	3.136	3.182	2.136	1.909	1.5	1.591	1.318	2.636	2.727	2.955
■ 4.6.3. Prerequisite Analysis	1.286	1.143	2.0	1.714	1.714	1.429	1.286	1.143	2.714	1.857	2.429	1.714	1.0	2.143	1.0	1.0	2.143	2.429
○ 4.7. Procedural Reasoning	1.588	2.035	1.882	1.788	2.565	2.788	2.894	2.035	2.882	1.906	1.647	1.176	1.353	1.0	1.471	1.553	2.224	2.647
■ 4.7.1. Procedural Step Analysis	1.588	2.035	1.882	1.788	2.565	2.788	2.894	2.035	2.882	1.906	1.647	1.176	1.353	1.0	1.471	1.553	2.224	2.647
○ 4.8. Spatial Reasoning	1.077	1.077	1.385	1.385	1.846	2.538	2.154	2.154	3.077	1.538	1.231	1.462	1.077	1.0	1.308	1.462	1.769	2.692
■ 4.8.1. Geometric Property Analysis	1.077	1.077	1.385	1.385	1.846	2.538	2.154	2.154	3.077	1.538	1.231	1.462	1.077	1.0	1.308	1.462	1.769	2.692
● 5. Trustworthiness	1.196	1.465	1.499	1.549	1.533	1.58	1.846	1.903	2.282	1.678	1.489	1.35	1.448	1.139	1.178	1.392	1.82	3.713
○ 5.1. Information Integrity	1.196	1.465	1.499	1.549	1.533	1.58	1.846	1.903	2.282	1.678	1.489	1.35	1.448	1.139	1.178	1.392	1.82	3.713
■ 5.1.1. Synthetic Content Detection	1.293	1.436	1.464	1.714	1.879	1.579	2.029	2.1	2.586	1.762	1.857	1.5	1.4	1.071	1.143	1.621	1.861	4.264
■ 5.1.2. Tampering Detection	1.1	1.494	1.533	1.383	1.188	1.581	1.662	1.706	1.979	1.594	1.121	1.2	1.496	1.206	1.212	1.162	1.779	3.162
● 6. Understanding	2.217	2.384	3.062	3.128	3.219	3.271	3.447	3.358	3.792	3.453	2.464	2.226	2.149	2.287	2.044	2.609	3.259	4.255
○ 6.1. Contextual Understanding	1.908	2.244	2.992	2.866	3.208	3.184	3.694	3.237	4.097	3.214	2.095	1.853	2.158	2.074	1.686	2.308	3.126	4.475
■ 6.1.1. Convention Understanding	1.625	2.25	3.031	2.875	2.844	2.844	3.531	3.094	3.812	3.0	2.0	1.469	2.125	1.719	1.562	2.188	3.156	4.188
■ 6.1.2. Thematic Interpretation	2.19	2.238	2.952	2.857	3.571	3.524	3.857	3.381	4.381	3.429	2.19	2.238	2.19	2.429	1.81	2.429	3.095	4.762
○ 6.2. Language Understanding	2.387	2.549	3.285	3.753	3.743	4.053	3.604	3.679	3.61	3.758	3.11	2.888	1.807	2.58	1.99	3.356	3.864	3.875
■ 6.2.1. Ambiguity Resolution	1.571	2.286	2.429	2.286	2.714	2.857	3.429	2.571	3.286	2.714	1.714	2.286	2.143	2.429	1.571	2.571	2.857	4.286
■ 6.2.2. Information Extraction	3.031	3.127	3.884	4.033	4.333	4.298	4.704	4.49	4.433	4.518	3.059	3.155	2.116	2.249	2.158	3.32	4.352	4.423
■ 6.2.3. Reference Resolution	2.333	2.667	3.0	3.667	3.667	3.667	2.333	2.333	1.667	2.667	3.667	3.667	1.333	3.667	1.333	3.0	4.0	4.0
■ 6.2.4. Semantic Similarity	3.0	3.667	3.333	5.0	4.333	5.0	4.0	4.667	4.333	5.0	4.0	3.667	2.333	2.333	2.333	5.0	3.667	4.667
■ 6.2.5. Translation	2.0	1.0	3.778	3.778	3.667	4.444	3.556	4.333	4.333	3.889	3.111	1.667	1.111	2.222	2.556	2.889	4.444	2.0
○ 6.3. Scene Understanding	2.339	2.479	3.264	3.282	3.236	3.086	3.3	3.307	3.864	3.693	2.504	2.482	2.657	2.686	2.464	2.657	3.211	4.071
■ 6.3.1. Scene Classification	2.339	2.479	3.264	3.282	3.236	3.086	3.3	3.307	3.864	3.693	2.504	2.482	2.657	2.686	2.464	2.657	3.211	4.071
○ 6.4. Semantic Interpretation	2.233	2.266	2.709	2.612	2.689	2.761	3.19	3.207	3.596	3.146	2.148	1.68	1.974	1.81	2.036	2.115	2.835	4.598
■ 6.4.1. Symbol Interpretation	2.233	2.266	2.709	2.612	2.689	2.761	3.19	3.207	3.596	3.146	2.148	1.68	1.974	1.81	2.036	2.115	2.835	4.598

Figure 17: **Global Asset: The Comprehensive Capability Heatmap.** This scorecard aggregates the performance of all evaluated models across the full taxonomy constructed in Appendix E. Darker colors indicate higher proficiency. This global view serves as the entry point for diagnostic analysis, allowing for the identification of macro-level trends and capability gaps across the model landscape.

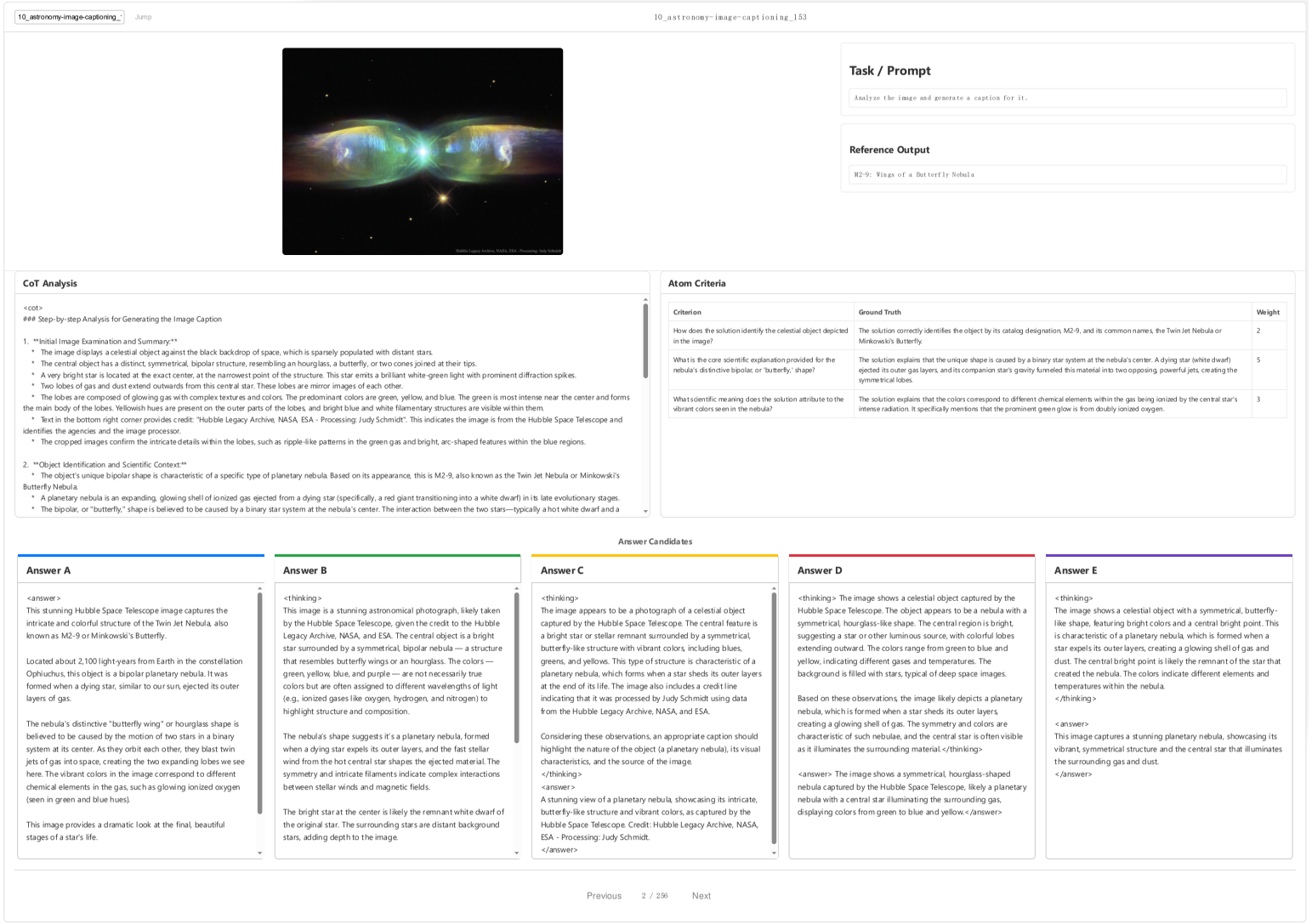


Figure 18: **Qualitative Asset: The Annotation Interface Snapshot.** This figure illustrates the granular data annotation workflow. It displays the key components of the evaluation process: the visual input (e.g., the M2-9 Nebula), the generated Chain-of-Thought (CoT) analysis, the atomic evaluation criteria, and the comparative answer candidates. This interface visualizes the ground truth generation and verification process.