VeriSciQA: An Auto-Verified Dataset for Scientific Visual Question Answering

Anonymous ACL submission

Abstract

Large Vision-Language Models (LVLMs) show promise for scientific visual question answering (SVQA), yet open-source models remain limited by the lack of public, large-scale, high-quality training data. Existing LVLM-synthesized SVQA datasets often contain systematic errors due to model limitations and information asymmetry between figures and text. To address this, we propose a cross-modal verification framework that generates QA pairs from figure-citing paragraphs and verifies them against the corresponding figures to filter out erroneous QA pairs. Using this framework, we curate VERISCIQA, a dataset of 20,272 QA pairs across 20 scientific domains and 12 figure types. Human evaluation shows that VERISCIQA achieves higher quality than existing LVLM-synthesized SVQA datasets, particularly in factual correctness. Fine-tuning on VERISCIQA consistently improves SVQA performance on external benchmarks, yielding larger average gains than fine-tuning on existing datasets, with gains scaling with data size. Our results show that cross-modal verification can produce high-quality data even from imperfect generators.

1 Introduction

Recent advances in Large Vision-Language Models (LVLMs) (Alayrac et al., 2022; Liu et al., 2023b; Bai et al., 2023; OpenAI, 2023) have achieved strong performance on general-purpose vision-language tasks. However, scientific-figure understanding remains a major bottleneck, as these figures encode dense quantitative evidence (e.g., trends, comparisons, ablations, and error bars) that requires specialized visual reasoning (Wang et al., 2024). Reliable figure understanding is a prerequisite for building research assistants that can answer figure-grounded questions, cross-check reported results, and support reproducible scientific analysis.

To this end, we focus on the task of *Scientific Visual Question Answering* (SVQA), where an LVLM answers questions grounded in figures from scientific papers. While proprietary LVLMs are reported to approach human performance on SVQA benchmarks (Wang et al., 2024), open-source models still lag substantially behind their proprietary counterparts. A plausible cause is the lack of large-scale, high-precision SVQA training data that is both diverse and visually grounded.

Despite the growing interest in SVQA, existing dataset construction methods face a persistent trade-off among scalability, diversity, and correctness. Template-based efforts such as FigureQA (Kahou et al., 2017) and SciFiBench (Roberts et al., 2024) achieve scalability but remain constrained by fixed question formats, limiting diversity. Manually curated datasets like CharXiv (Wang et al., 2024) ensure correctness through expert annotation, albeit at the cost of scalability and diversity. Recent LVLM-based methods, including ArXivQA (Li et al., 2024), SPIQA (Pramanick et al., 2024), and LiveXiv (Shabtay et al., 2024), improve scalability and diversity, yet LVLM-driven generation usually introduces errors due to hallucination when interpreting visual inputs (Bai et al., 2024). Despite this limitation, LVLM-based methods still offer the best prospect for scaling diverse SVQA data, but their correctness remains an open challenge.

Among LVLM-based methods, ArXivQA typifies this line of work, and its resulting SVQA dataset has been adopted in some follow-up studies (Tong et al., 2024; Dai et al., 2024; Zhang et al., 2024b). However, our systematic analysis of its SVQA dataset reveals four recurring categories of erroneous QA pairs (detailed in § 2). Collectively, these errors raise concerns about the reliability of LVLM-based methods, highlighting potential risks of fine-tuning LVLMs on their synthesized SVQA datasets.

We trace these errors to two primary sources.

(i) *Model-intrinsic hallucination*: LVLMs may fabricate visual evidence or trends that are not present in the figure, consistent with recent findings that even state-of-the-art LVLMs may hallucinate visual content (Bai et al., 2024). (ii) *Information asymmetry*: the QA generation model only sees the figure (or figure+caption), while the figure-citing paragraphs define symbols, experimental settings, and the intended takeaway. This mismatch causes the LVLM to over-rely on its prior knowledge, producing questions whose answers are not entailed by the visual evidence. These observations suggest two potential directions for improvement: (1) introducing verification mechanisms to detect and filter such hallucinations, and (2) incorporating figure-associated context to mitigate information asymmetry.

Several recent methods have made progress along both directions. LiveXiv (Shabtay et al., 2024) uses a second LVLM as a verifier to review each generated answer against the same figure, discarding QA pairs where the verifier disagrees with the generated answer. SPIQA (Pramanick et al., 2024) incorporates figure-citing paragraphs as additional context for QA generation. However, same-figure LVLM agreement is an unreliable proxy for visual grounding: the generator and verifier, both operating on the same visual input, can share correlated hallucinations. Meanwhile, adding figure-citing paragraphs alone reduces information asymmetry but does not prevent visually unsupported QA pairs from entering the dataset. To address both limitations, we decouple generation and verification across modalities: generate QA from figure-citing paragraphs to mitigate information asymmetry, and verify against figures to enforce visual grounding. This cross-modal design reduces correlated errors, since hallucinations from text-based generation and vision-based verification are less likely to coincide.

Concretely, we leverage two key properties of peer-reviewed papers: (1) figure-citing paragraphs typically reflect the authors' intended interpretation of the figure, and (2) many statements in these paragraphs, such as reported trends or numerical results, are verifiable against the figures. Building on these properties, we propose a Cross-Modal Verification framework. In this framework, figure-citing paragraphs are first fed into an LLM to produce questions and answers, mitigating the information asymmetry. An LVLM then generates distractors conditioned on the figure to form multiple-choice QA pairs. Auxiliary filters first target common error patterns, after which cross-modal verification discards QA pairs whose answers are not visually supported by the figure.

Our main contributions are as follows:

- **A Cross-Modal Verification framework for SVQA synthesis.** We propose a scalable framework that generates questions and answers from figure-citing paragraphs, produces plausible distractors via an LVLM to form multiple-choice QA pairs, and verifies visual grounding against figures, leveraging the inherent text-figure alignment in peer-reviewed scientific papers.

- **A diverse, high-quality, and challenging SVQA dataset.** Built with our proposed framework, VERISCIQA comprises 20,272 QA pairs spanning 20 scientific domains, 12 figure types, and 5 question types (§4.1). Human evaluation confirms superior quality compared to existing SVQA datasets. Zero-shot evaluation on our dataset reveals a 15.9 percentage-point accuracy gap between the best proprietary and open-source models.

- **Scaling Data, Scaling Capability.** Models fine-tuned on our dataset achieve consistent improvements across diverse SVQA benchmarks, outperforming those trained on existing SVQA datasets. As our training data scales from 500 to 20,272 examples, fine-tuned models improve monotonically, reaching an average gain of +2.05% over the base model.

We release both the dataset and Data-Juicer (Chen et al., 2025) operators that implement our framework (Appendix I), helping the community reproduce, reuse, and extend the proposed pipeline.

2 Motivation

A prominent line of work in SVQA dataset construction relies on LVLM-based methods that prompt an LVLM with a figure from the paper, its accompanying caption, and optional context parsed from the paper's LaTeX source. ArXivQA (Li et al., 2024), a representative work in this line, synthesizes more than 100 K multiple-choice QA pairs. Despite its large scale, our small audit reveals that it still exhibits a 37% error rate, raising concerns about the data quality of SVQA datasets produced by such LVLM-based methods.

Systematic error analysis of ArXivQA. We randomly sample 100 items from ArXivQA, each released as a five-tuple ⟨`Question`, `Options`, `Label`, `Rationale`, `Figure`⟩, where `Label` refers to the LVLM's prediction rather than ground truth. Our systematic analysis reveals four common categories of erroneous QA pairs, which we refer to as follows:

- **(E1) Incorrectly Visually Grounded** (19/100): questions or answers that exhibit mismatches between textual claims and the visual evidence (Bai et al., 2024), e.g., misreading chart values (Fig. 6a).
- **(E2) Figure–Intent Misaligned** (7/100): questions whose semantics diverge from the figure's purpose, e.g., treating a figure illustrating an embedding space as a natural image and asking for scene classification (Fig. 6b).
- **(E3) Non-Visual** (6/100): answers can be inferred directly from the question and common/domain-specific knowledge, making the figure unnecessary (Fig. 6c).
- **(E4) Outside-Knowledge** (5/100): questions whose visual elements require paper context to interpret, e.g., a heatmap whose cells cannot be identified from the figure alone as co-occurrence, confusion, or correlation (Fig. 6d).

Root causes: hallucination and information asymmetry. LVLMs are known to *hallucinate*, frequently producing responses inconsistent with the visual content provided (i.e., cross-modal inconsistency) (Bai et al., 2024). The first two failure modes, E1 and E2, stem directly from this phenomenon. Methods such as ArXivQA rely solely on LVLM generation without verification, allowing hallucinated responses to pass through. While LiveXiv (Shabtay et al., 2024) uses a second LVLM to verify generated answers, both the generator and the verifier operate on the same visual input and may share similar hallucination patterns. The other root cause is *information asymmetry*. The input to the LVLM often contains only the figure and its caption, omitting essential figure-associated context that authors describe in the main body of the paper. Without this context, the model may over-rely on its broad knowledge and generate QA pairs that extend beyond what the figure alone conveys, leading to E2–E4 errors. SPIQA (Pramanick et al., 2024) incorporates figure-citing paragraphs as additional context for QA generation, but without reliable verification, erroneous QA pairs may persist. Thus, SPIQA and LiveXiv each target one root cause but not the other, motivating a unified approach that combines enriched context with verification, enabled by figure-citing paragraphs.

Key insight: figure-citing paragraphs contain verifiable claims about figures. Figure-citing paragraphs, which can be extracted from the paper's LaTeX source by matching figure reference commands (e.g., `\cref`), often contain verifiable claims such as "As shown in Fig. 2, accuracy rises by 20%." In this example, the claim suggests that the figure likely aims to report model accuracy, and asserts a specific trend that can be checked against the figure. More generally, claims in figure-citing paragraphs exhibit two key properties: they tend to reflect the authors' intended interpretation of the figure, and they often describe observable properties (e.g., trends or numerical results) that are verifiable against the figure. These properties give rise to *Cross-Modal Verification*: questions and answers are first generated from such claims, promoting alignment with the authors' intent; the generated answers can then be verified against the figure, helping ensure visual grounding. Since hallucinations from text-based generation and those from vision-based verification are less likely to be correlated, this cross-modal framework, together with auxiliary filters, helps systematically mitigate both hallucination and information asymmetry errors (E1–E4).

3 Methodology

To mitigate the four error types (E1–E4) identified in § 2, we propose a *Cross-Modal Verification* framework for curating VERISCIQA from arXiv papers. As shown in Fig. 1, the framework has two stages: **Generation**, which derives candidate multiple-choice QA pairs from figure-citing paragraphs, and **Verification**, which filters them through text-based and vision-based checks. In Generation, we decompose figure-citing paragraphs into atomic claims, convert each visually grounded claim into a question with its correct answer, and use an LVLM to generate plausible figure-conditioned distractor options, forming a multiple-choice QA pair. In Verification, text-based checks assess source consistency and visual dependence, while a final vision-based check uses self-consistency to validate the answer against the

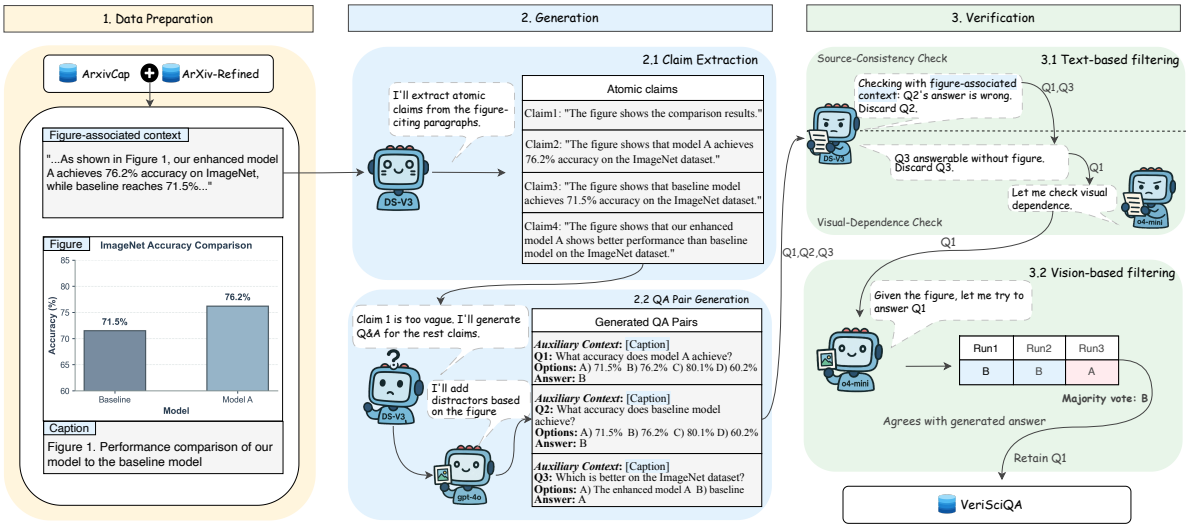


Figure 1: Overview of our Cross-Modal Verification framework for curating VERISCIQA. We first extract figure-citing paragraphs from arXiv papers as figure-associated context. **Generation** stage decomposes each paragraph into atomic claims, generates questions and correct answers via an LLM, and produces visually grounded distractors via an LVLM to form multiple-choice QA pairs; **Verification** stage applies text-based filtering followed by vision-based filtering to remove erroneous QA pairs.

figure. We retain only QA pairs that pass all checks. Because these checks are model-based rather than oracle judgments, we empirically validate them through human evaluation, rejected-sample analysis, ablations, and downstream transfer.

3.1 Framework Formalization

Let a scientific paper be $\mathcal{D} = (\mathcal{T}, \mathcal{F})$, where $\mathcal{T}$ is its text and $\mathcal{F} = \{(F_i, C_i)\}_{i=1}^{N}$ is the set of figure-caption pairs, with F_i denoting a figure and C_i its caption. For each F_i, its *figure-associated context* $\mathcal{P}_i \subset \mathcal{T}$ contains body paragraph(s) explicitly referring to F_i; for readability, we omit the subscript i below. Let M_{text} be the LLM and $M_{\text{vision}}^{\text{gen}}$, $M_{\text{vision}}^{\text{verify}}$ be LVLMs for distractor generation and verification. Generation maps $\mathcal{P}$ to candidate multiple-choice QA pairs $q_j = (Q_j, O_j, A_j^*)$, where Q_j is the question, O_j is the option set, and A_j^* is the correct option. Verification retains only candidates passing three filters:

$$V_{\text{src}} : (Q, O, A^*, \mathcal{P}) \xrightarrow{M_{\text{text}}} \{\text{True}, \text{False}\}$$

$$V_{\text{vis_dep}} : (Q, O, A^*, C) \xrightarrow{M_{\text{vision}}^{\text{verify}}} \{\text{True}, \text{False}\}$$

$$V_{\text{vis_con}} : (Q, O, A^*, F, C) \xrightarrow{M_{\text{vision}}^{\text{verify}}} \{\text{True}, \text{False}\}$$

A candidate q_j is retained in the final dataset when:

$$V_{\text{src}}(Q_j, O_j, A_j^*, \mathcal{P}) \wedge V_{\text{vis_dep}}(Q_j, O_j, A_j^*, C) \wedge V_{\text{vis_con}}(Q_j, O_j, A_j^*, F, C) = \text{True}. \quad (1)$$

3.2 Generation

This stage consists of the following three steps:

Claim Extraction. We instruct M_{text} to extract and reformulate factual statements about the figure F from $\mathcal{P}$ as atomic claims:

$$\mathcal{S} = \{s_j\} = M_{\text{text}}(\mathcal{P}), \quad (2)$$

where each s_j denotes an atomic claim, a concise, self-contained sentence that asserts what F shows, following the pattern "The figure shows..." (see the illustrative examples in Fig. 1).

Question & Answer Generation. For each atomic claim $s_j \in \mathcal{S}$, we instruct M_{text} to generate a question and its correct answer:

$$(Q_j, A_j^*) = M_{\text{text}}(s_j, C, \mathcal{P}), \quad (3)$$

where the model converts s_j into a question Q_j and derives the correct answer A_j^* from the claim. Because each s_j explicitly describes what F shows, generating questions from it promotes figure-intent alignment, mitigating *figure-intent misalignment* errors (E2). M_{text} may decline to produce a question and answer if s_j is too vague or lacks sufficient grounding, e.g., Claim 1 in Fig. 1.

Distractor Generation. To complete the multiple-choice format, we instruct $M_{\text{vision}}^{\text{gen}}$ to generate three plausible distractor options:

$$O_j = M_{\text{vision}}^{\text{gen}}(Q_j, A_j^*, F, C), \quad (4)$$

where O_j includes the correct answer A_j^* along with distractors. By conditioning on both the figure F and the correct answer A_j^*, the LVLM can produce distractors that are visually plausible yet incorrect. Since captions typically contain the information needed to comprehend a figure, each QA pair is accompanied by its corresponding figure's caption C, helping make the pair self-contained and mitigating *outside-knowledge* errors (E4).

3.3 Verification

Each candidate q_j is retained only if it passes all three filters in Eq. 1.

Text-Based Filtering. We first apply two filters without accessing figure F. **(1) Source-Consistency check.** V_{src} prompts M_{text} to answer Q_j from $\mathcal{P}$ and O_j; it returns **True** only if the model uniquely selects A_j^*, checking that the QA pair is supported by its source paragraph. **(2) Visual-Dependence check.** $V_{\text{vis_dep}}$ prompts $M_{\text{vision}}^{\text{verify}}$ to answer from only C, Q_j, and O_j, without F; it returns **True** only if the model cannot identify A_j^*, filtering out questions that are answerable without the figure from the caption, question, and options, thereby mitigating *non-visual* questions (E3).

Vision-Based Filtering. Finally, $V_{\text{vis_con}}$ prompts $M_{\text{vision}}^{\text{verify}}$ to answer from F, C, Q_j, and O_j; it returns **True** only if the prediction matches A_j^*. This checks whether the figure provides sufficient visual evidence to distinguish the annotated answer from the distractors, mitigating *incorrectly visually grounded* (E1) and *outside-knowledge* (E4) errors.

3.4 Instantiation

We instantiate our framework using scientific figures from peer-reviewed papers on arXiv, leveraging existing datasets and language models to implement each stage.

Data Sources. We combine two complementary data sources: ArXivCap (Li et al., 2024), which provides high-quality figure–caption pairs from peer-reviewed papers with heuristic filtering; and RedPajama-ArXiv-Refined (Data-Juicer Team, 2023), which supplies cleaned LaTeX source for extracting figure-associated context. Intersecting these datasets by arXiv ID[1] yields 572K papers. We randomly shuffle this pool to avoid temporal or domain-based ordering bias, then process papers in batches until reaching our target dataset size, ultimately processing 44,345 papers.

[1]The identifier assigned to each arXiv paper (e.g., 2301.12345).

Table 1: Number of QA pairs retained after each step.

Step	Count	Retention
Papers	44,345	—
Atomic claims extracted	680,877	100%
QA pairs generated	261,116	38.4%
After text-based filtering	55,372	8.1%
After vision-based filtering	20,272	3.0%

Context Extraction. For each figure-caption pair (F_i, C_i), we extract its associated context $\mathcal{P}_i$ by identifying paragraphs that cite F_i via LaTeX reference commands (e.g., `\ref`, `\cref`, or `\autoref`). Details are provided in Appendix A.1.

Model Selection. We instantiate M_{text}, $M_{\text{vision}}^{\text{gen}}$, and $M_{\text{vision}}^{\text{verify}}$ with `DeepSeek-v3` (Liu et al., 2024a), `GPT-4o` (OpenAI, 2024), and `o4-mini` (OpenAI, 2025b), respectively. For vision-based filtering, we query $M_{\text{vision}}^{\text{verify}}$ three times and retain only QA pairs where the 2-of-3 majority answer matches A^*, following self-consistency (Wang et al., 2022); model-choice rationale and decoding settings are provided in Appendix A.

Resulting Dataset. Applying our Cross-Modal Verification framework yields VERISCIQA, a dataset of 20,272 quality-controlled QA pairs spanning 20 arXiv categories. Each QA instance consists of a question, multiple-choice options, the correct answer, the associated figure, and its caption. We release the full 20,272 QA pairs as a training resource; downstream effectiveness is evaluated exclusively on external benchmarks (§4.4). For both training and evaluation, models receive the figure, its caption, the question, and the options as input, and are expected to predict the correct answer. This input design mirrors how readers naturally examine a scientific figure alongside its caption (our visual-dependence check helps ensure that the caption alone is insufficient to answer each question). Table 1 shows the filtering statistics at each step; Appendix C analyzes the rejected candidates and shows that most are erroneous.

4 Experiments

We first analyze the diversity of our dataset (§4.1), then evaluate leading LVLMs on it to verify that it is non-trivial for current models and to identify gaps between open-source and proprietary models (§4.2), and finally compare our dataset's quality against existing SVQA datasets (§4.3). We show

that fine-tuning on our dataset yields consistent gains across multiple external benchmarks, with the average gain scaling up with data size (§4.4), and verify the effectiveness of our filtering steps via ablation (§4.5).

4.1 Diversity Assessment

Our dataset comprises 20,272 diverse QA pairs spanning three key dimensions: scientific domains, figure types, and question types. As shown in Fig. 2, our dataset covers 20 scientific domains and 12 figure types. Since our data are sourced from arXiv, CS and Physics are the most frequent domains (Li et al., 2024); Line Plot is the most prevalent figure type, as plots and charts dominate scientific publications (Pramanick et al., 2024). Notably, our dataset is reasoning-heavy: Relational, Comparative, and Compositional questions together comprise 77% of the dataset. This is because our framework generates QA pairs from figure-citing paragraphs (§3.2), which tend to convey key findings derived from the figure rather than low-level perceptual details. Detailed labeling protocol, taxonomy definitions, and complete distribution tables are provided in Appendix B.

4.2 Difficulty Assessment

We evaluate leading proprietary and open-source LVLMs on our dataset in a zero-shot setting.

Setup. We evaluate six models on 2,000 randomly sampled examples: four proprietary models (`OpenAI o3` (OpenAI, 2025a), `GPT-5.2` (OpenAI, 2025c), `o4-mini` (OpenAI, 2025b), `Gemini-3-Flash` (Google DeepMind, 2025)) and two open-source models (`Qwen3-VL-235B-A22B-Thinking` (Team, 2025b), `InternVL3.5-38B` (Chen et al., 2024c)). Detailed configurations are provided in Appendix F.

Results. The best open-source model, `Qwen3-VL`, reaches 64.6% overall accuracy. The best proprietary model, `Gemini-3-Flash`, achieves 80.5%, leaving considerable room for open-source models to improve (detailed results in Appendix G). As shown in Fig. 3, the largest gaps appear in domains such as Physics and Q-Bio, figure types such as Box Plot, Line Plot, and Heatmap, and question types such as Comparative and Relational, which often require precise quantitative reading or multi-step reasoning. Moreover, `Gemini-3-Flash` outperforms our verifier `o4-mini`, suggesting that the retained QA pairs are not trivially biased toward the verifier.

Table 2: Human evaluation ratings on 200 QA pairs per dataset; higher is better.

Dimension	Rating (1–5)			IAA (α)
	VERISCIQA	SPIQA	ArXivQA	
Factual Correctness	**4.49**	4.19	3.96	0.71
Intent Alignment	**4.35**	4.34	4.06	0.62
Visual Dependency	**4.40**	4.35	4.17	0.71
Self-Containment	4.34	**4.52**	4.18	0.60

4.3 Quality Comparison

We compare VERISCIQA with SPIQA (Pramanick et al., 2024) and ArXivQA (Li et al., 2024) through a human evaluation of 600 QA pairs, 200 per dataset, where ten CS graduate students served as annotators and each pair was rated by two annotators on four 5-point quality dimensions aligned with the error types in §2: Factual Correctness (E1), Intent Alignment (E2), Visual Dependency (E3), and Self-Containment (E4). As shown in Table 2, VERISCIQA achieves the highest ratings on E1–E3 while remaining competitive on E4, suggesting that our framework improves factuality, intent alignment, and visual grounding without substantially sacrificing self-containment. The slightly lower self-containment score compared with SPIQA suggests that some questions still inherit contextual dependencies from figure-citing paragraphs, despite our E4 filtering. All four dimensions achieve moderate inter-annotator agreement (α = 0.60–0.71) (Krippendorff, 2018); additional annotation details are provided in Appendix D.

4.4 Effectiveness as Training Data

Setup. We compare the training effectiveness of VERISCIQA against datasets constructed by different methods: template-based SciFiBench (Roberts et al., 2024) (Figure2Caption split, 1,000 examples from General and CS categories) and LVLM-driven ArXivQA and SPIQA (20,272 randomly sampled examples each). For VERISCIQA, we train both the full 20,272-example dataset and a 1,000-example subset matched to SciFiBench's scientific-domain distribution for a same-scale comparison. Using LoRA (Hu et al., 2022), we fine-tune `Qwen2.5-VL-7B-Instruct` (Team, 2025a) and evaluate all fine-tuned variants on three external scientific figure benchmarks: CharXiv (Wang et al., 2024) (4,000 descriptive and 1,000 reasoning examples), MMStar (Chen et al., 2024b) (science & technology subset, 250 examples), and MathVista (Lu et al., 2023) (scientific reasoning subset, 122 examples). Additional training details are in

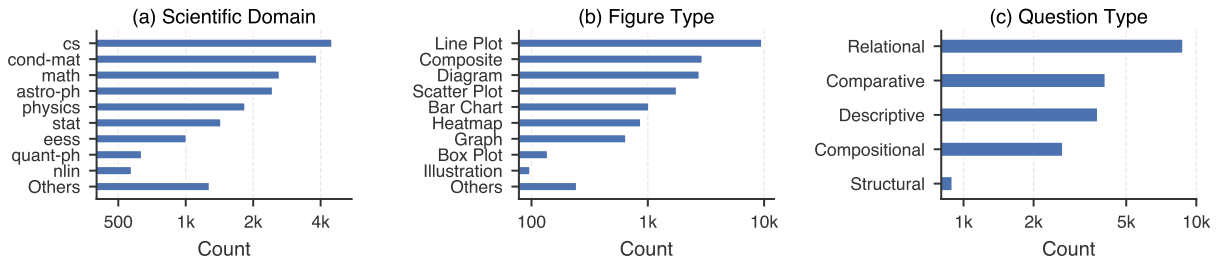


Figure 2: Dataset composition of VERISCIQA across three dimensions: (a) scientific domain distribution showing the top 9 of 20 arXiv categories, (b) figure type coverage showing the top 9 of 12 categories, and (c) distribution across five question types. Complete breakdowns are in Appendix B.

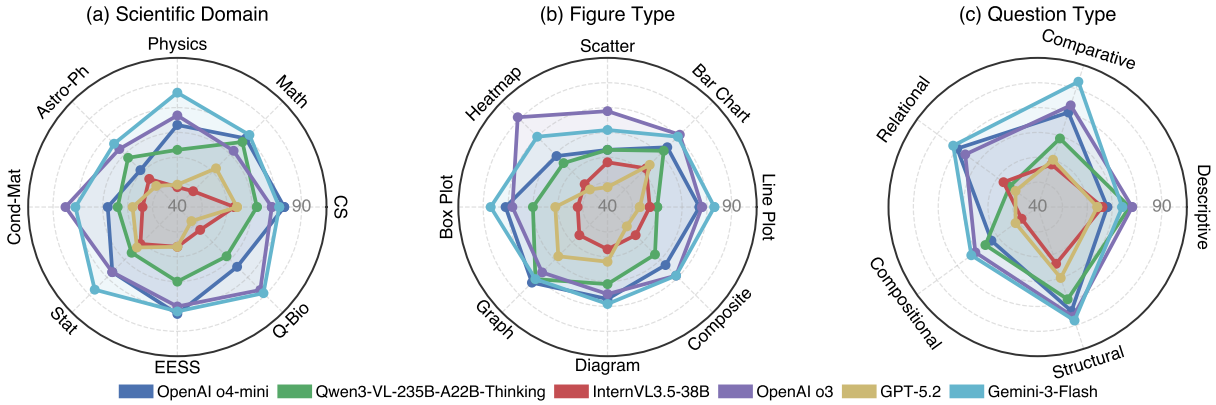


Figure 3: Zero-shot model accuracy on VERISCIQA broken down by (a) scientific domain, (b) figure type, and (c) question type. The largest gaps between proprietary and open-source models appear in domains like Physics, figure types like Box Plot and Heatmap, and question types like Comparative.

Appendix H.

Results. We report performance gains relative to the original `Qwen2.5-VL-7B-Instruct` (baseline). As shown in Table 3, the model trained on our dataset (20,272 examples) achieves the highest average gain (+2.05%) compared to models trained on other datasets. Notably, our dataset is the only one that improves CharXiv Reasoning (+2.10%), whereas other datasets show degradation. We attribute this improvement to our higher data quality (§4.3) and reasoning-heavy question distribution (§4.1), as also reflected in the gains on MMStar Science & Technology and MathVista Scientific Reasoning. On CharXiv Descriptive, our dataset improves by +3.28%, though ArXivQA leads with +5.42%.

Training Data Scaling. To assess the scalability of our dataset, we use the same training setup described above but vary the amount of training data from 500 to 20,272 examples. As shown in Fig. 4, the average gain scales monotonically with training data size, nearly doubling from +1.05% to +2.05% as the data grow from 500 to 20,272 examples. This trend suggests potential for further improvement with more of our training data.

Table 3: Absolute baseline accuracy (%) and fine-tuning gains (Δ%). CX-D/R: CharXiv Descriptive/Reasoning; MM-ST: MMStar Science & Technology; MV-SR: MathVista Scientific Reasoning. Bold indicates best gain per column.

Training Data	CX-D	CX-R	MM-ST	MV-SR	Avg.
Baseline (no fine-tuning)	66.0	42.4	48.0	67.2	55.9
SciFiBench (1K)	+4.28	−1.10	+0.80	0.00	+1.00
VERISCIQA (1K)	+5.50	−1.00	+0.80	−0.82	+1.12
ArXivQA (20K)	**+5.42**	−1.40	0.00	−0.81	+0.80
SPIQA (20K)	+2.45	−2.70	+0.40	−1.63	−0.37
VERISCIQA (20K)	+3.28	**+2.10**	**+2.00**	**+0.82**	**+2.05**

4.5 Ablation Study

Setup. To ablate the effect of each filtering step in our framework, we compare four 20,272-example training sets: no filtering, text-based filtering only, vision-based filtering only, and full verification. Training and evaluation follow §4.4.

Results and Analysis. As shown in Table 4, training on unfiltered data hurts performance (−0.37%), while text-based filtering provides a modest average gain of +0.24%. Although stopping at text-based filtering would retain 2.7× more QA pairs than full verification (see Table 1), this quantity

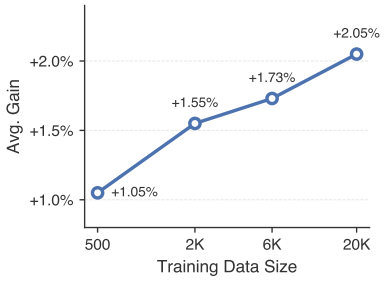


Figure 4: Average fine-tuning gain on external benchmarks, averaged across CharXiv (CX-D/R), MMStar-ST, and MathVista-SR; gains rise as the number of VERISCIQA training examples increases.

Table 4: Average gains over CharXiv, MMStar-ST, and MathVista-SR for 20K-example ablation variants.

Training Data (20K examples)	Avg. Gain (%)
Generation only (no filtering)	−0.37
With text-based filtering only	+0.24
With vision-based filtering only	+1.13
With full verification	**+2.05**

advantage comes at a quality cost: without vision-based filtering, *incorrectly visually grounded* (E1) and *outside-knowledge* (E4) errors remain unfiltered. Vision-based filtering alone is stronger (+1.13%) than text-based filtering alone, but still falls short of full verification (+2.05%). This suggests that the two filters are complementary: they target different error perspectives, and combining them yields better results than either filter alone. Notably, although the core QA generation in VERISCIQA uses `DeepSeek-v3`, the resulting verified data yields larger average training gains than prior datasets synthesized with proprietary LVLMs (see §4.4). This suggests that cross-modal verification can yield high-quality SVQA training data even when the initial candidates come from imperfect generators.

5 Related Work

Scientific Visual Question Answering Datasets. Existing SVQA datasets are built through template-based generation (Kahou et al., 2017; Roberts et al., 2024; Methani et al., 2020), expert annotation (Wang et al., 2024; Masry et al., 2022), or LVLM-based synthesis (Li et al., 2024; Pramanick et al., 2024; Shabtay et al., 2024; Li and Tajbakhsh, 2023). LVLM-based datasets improve scalability and diversity, but LVLMs can hallucinate visual details not supported by the figure; to mitigate such hallucinations, SPIQA (Pramanick et al., 2024) provides figure-citing paragraphs when generating QA pairs, while LiveXiv (Shabtay et al., 2024) verifies generated answers through inter-model agreement on the same visual input. In contrast, VERISCIQA decouples modalities by generating questions and answers from figure-citing paragraphs without the figure, then verifying them against the figure.

Data Synthesis and Verification Techniques. Our framework builds on self-instruction-style data synthesis (Wang et al., 2023) and follows recent synthetic-data surveys (Liu et al., 2024b; Long et al., 2024) in prioritizing verification over sheer scale. Unlike LiveXiv's same-figure agreement and blind-test filtering (Shabtay et al., 2024), our verification combines cascaded text-based checks with vision-based validation strengthened by self-consistency (Wang et al., 2022) via majority voting. It is also distinct from general fact-checking systems that rely on retrieval or external knowledge sources (Lewis et al., 2020; Nakano et al., 2021; Gao et al., 2023): our checks are grounded only in the paper context and the target figure.

Large Vision-Language Models for Science. Recent LVLMs (Alayrac et al., 2022; Liu et al., 2023b; OpenAI, 2023; Chen et al., 2024c) have motivated specialized chart-understanding models such as MatCha (Liu et al., 2023a), ChartLlama (Han et al., 2023), and TinyChart (Zhang et al., 2024a). Their scientific and mathematical reasoning abilities are evaluated by benchmarks including CharXiv (Wang et al., 2024), SciFiBench (Roberts et al., 2024), ChartX (He et al., 2024), and MathVista (Lu et al., 2023), alongside broader LVLM benchmarks such as MMBench (Liu et al., 2024c), MME (Fu et al., 2025), and MMStar (Chen et al., 2024b). These evaluations expose a persistent proprietary–open-source gap, motivating large-scale verified SVQA data for improving scientific visual reasoning.

6 Conclusion

We curate VERISCIQA, a dataset of 20,272 high-quality multiple-choice SVQA pairs spanning 20 scientific domains, created via our Cross-Modal Verification framework that produces reliable training data from unreliable generators by leveraging cross-modal consistency rather than expensive human annotation. Experiments show consistent SVQA benchmark gains from fine-tuning on VERISCIQA, with improvements scaling with data size. More broadly, this framework may extend to other settings where textual descriptions are aligned with visual content.

Limitations

Model-based verification is not an oracle. All filters in our framework are implemented with LLMs or LVLMs and therefore remain imperfect. They can over-filter valid QA pairs or fail to remove subtle errors. We mitigate this through cascaded verification, human evaluation, rejected-sample analysis, and downstream validation on external benchmarks, but VERISCIQA should be viewed as quality-controlled rather than error-free.

Bias toward author-stated takeaways. Because candidate QA pairs are generated from figure-citing paragraphs, VERISCIQA is naturally biased toward the claims that authors choose to verbalize about a figure. This favors relational and comparative questions that reflect the paper's intended message, but may under-represent low-level perceptual questions or visually valid observations that are not explicitly discussed in the surrounding text.

Scope and format. Our data are derived from arXiv papers, and therefore skew toward English, STEM, LaTeX-heavy, and figure-dense research areas. In addition, VERISCIQA uses a multiple-choice format for controlled training and evaluation, which may not fully capture open-ended scientific reasoning.

Use of Generative AI

Following conference disclosure policy, we disclose the use of generative AI in this work.

As a Research Object. Generative AI models are central to our proposed Cross-Modal Verification framework for dataset curation. Specifically, we use DeepSeek-v3 (Liu et al., 2024a) for atomic claim extraction, question–answer generation, and text-based filtering; GPT-4o (OpenAI, 2024) for visually grounded distractor generation; and o4-mini (OpenAI, 2025b) for vision-based verification. These models serve as core components of our methodology and their usage is fully described in §3.

Writing Assistance. We used ChatGPT to assist with grammar checking and polishing of the manuscript text. All generated content was reviewed and revised by the authors, who take full responsibility for the final manuscript.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

arXiv. 2025. arxiv category taxonomy. Accessed: 2025-11-13.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *Preprint*, arXiv:2308.12966.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2024. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*.

Daoyuan Chen, Yilun Huang, Zhijian Ma, Hesen Chen, Xuchen Pan, Ce Ge, Dawei Gao, Yuexiang Xie, Zhaoyang Liu, Jinyang Gao, and 1 others. 2024a. Data-juicer: A one-stop data processing system for large language models. In *Companion of the 2024 International Conference on Management of Data*, pages 120–134.

Daoyuan Chen, Yilun Huang, Xuchen Pan, Nana Jiang, Haibin Wang, Yilei Zhang, Ce Ge, Yushuo Chen, Wenhao Zhang, Zhijian Ma, Jun Huang, Wei Lin, Yaliang Li, Bolin Ding, and Jingren Zhou. 2025. Data-juicer 2.0: Cloud-scale adaptive data processing for and with foundation models. In *NeurIPS*.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024b. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024c. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Wenliang Dai, Nayeon Lee, Boxin Wang, Zhuolin Yang, Zihan Liu, Jon Barker, Tuomas Rintamaki, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nvlm: Open frontier-class multimodal llms. *arXiv preprint arXiv:2409.11402*.

Data-Juicer Team. 2023. Redpajama-arxiv (refined by data-juicer). https://huggingface.co/datasets/datajuicer/redpajama-arxiv-refined-by-data-juicer. A refined version of ArXiv dataset in RedPajama, containing 1,655,259 samples.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and 1 others. 2025. Mme: A comprehensive evaluation benchmark for multimodal large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and 1 others. 2023. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508.

Google DeepMind. 2025. Gemini 3 flash model card. https://deepmind.google/models/model-cards/gemini-3-flash.

Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. 2023. Chartllama: A multimodal llm for chart understanding and generation. *arXiv preprint arXiv:2311.16483*.

Renqiu He, Xiangchao Liu, Rui Zhang, Yichi Han, Chengzhuo Han, Wenjun Zhang, Hang Yu, Zhiyang Zeng, and Shanshan Zhang. 2024. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *arXiv preprint arXiv:2402.12185*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*.

Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. 2024. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models. *arXiv preprint arXiv:2403.00231*.

Shengzhi Li and Nima Tajbakhsh. 2023. Scigraphqa: A large-scale synthetic multi-turn question-answering dataset for scientific graphs. *arXiv preprint arXiv:2308.03349*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Eisenschlos. 2023a. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12756–12770.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and 1 others. 2024b. Best practices and lessons learned on synthetic data for language models. *CoRR*.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024c. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*.

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*.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279.

Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *Proceedings of the ieee/cvf winter conference on applications of computer vision*, pages 1527–1536.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders,

and 1 others. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

OpenAI. 2023. Gpt-4v(ision) system card. Technical report, OpenAI.

OpenAI. 2024. Gpt-4o system card. Technical report, OpenAI.

OpenAI. 2025a. Introducing o3 and o4-mini. Accessed: 2026-02-06.

OpenAI. 2025b. Openai o4-mini model card. Accessed: 2026-02-06.

OpenAI. 2025c. Update to GPT-5 system card: GPT-5.2. Accessed: 2026-02-06.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2024. Spiqa: A dataset for multimodal question answering on scientific papers. *Advances in Neural Information Processing Systems*, 37:118807–118833.

Jonathan Roberts, Kai Han, Neil Houlsby, and Samuel Albanie. 2024. Scifibench: Benchmarking large multimodal models for scientific figure interpretation. *Advances in Neural Information Processing Systems*, 37:18695–18728.

Armin Ronacher. 2008. Jinja2: A modern and designer-friendly templating language for python. `https://jinja.palletsprojects.com/`. Version accessed: 2025.

Nimrod Shabtay, Felipe Maia Polo, Sivan Doveh, Wei Lin, M Jehanzeb Mirza, Leshem Chosen, Mikhail Yurochkin, Yuekai Sun, Assaf Arbelle, Leonid Karlinsky, and 1 others. 2024. Livexiv–a multi-modal live benchmark based on arxiv papers content. *arXiv preprint arXiv:2410.10783*.

Qwen Team. 2025a. Qwen2.5-vl.

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

Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, and 1 others. 2024. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, and 1 others. 2024. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697.

Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. 2024a. Tinychart: Efficient chart understanding with visual token merging and program-of-thoughts learning. *arXiv preprint arXiv:2404.16635*.

Yi-Fan Zhang, Qingsong Wen, Chaoyou Fu, Xue Wang, Zhang Zhang, Liang Wang, and Rong Jin. 2024b. Beyond llava-hd: Diving into high-resolution large multimodal models. *arXiv preprint arXiv:2406.08487*.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2024. Swift:a scalable lightweight infrastructure for fine-tuning. *Preprint*, arXiv:2408.05517.

Appendix Overview.

A. Implementation Details 11
A.1 Data Preparation 12
A.2 Generation 12
A.3 Verification 12
B. Dataset Taxonomies 13
B.1 Figure Type Taxonomy 13
B.2 Question Type Taxonomy 14
B.3 Scientific Domain Taxonomy 14
C. Filtering Analysis 15
C.1 QA Pair Generation Analysis 15
C.2 Text-Based Filtering Analysis 15
C.3 Vision-Based Filtering Analysis 16
D. Human Evaluation Protocol 17
D.1 Annotation Interface 17
D.2 Annotation Guidelines 17
E. Additional Qualitative Examples 18
E.1 QA Pairs in VeriSciQA 18
E.2 Error Cases Captured by Framework 23
F. Model Details 26
G. Per-Category Performance Results 26
G.1 Performance by Scientific Domain 26
G.2 Performance by Figure Type 26
G.3 Performance by Question Type 26
H. Training Details 27
I. Framework Instantiation with Data-Juicer 27
I.1 Custom Operator: LaTeX Figure Context Extractor 27
I.2 Custom Operator: Claim Extractor 28
I.3 Operator Orchestration 28
I.4 Code Release 28

A Implementation Details

This section provides technical details about the instantiation of our Cross-Modal Verification framework for curating VeriSciQA, covering data preparation, generation, and verification stages. A

reusable Data-Juicer implementation is detailed in Appendix I.

A.1 Data Preparation

Our framework requires figure-associated context $\mathcal{P}$ (paragraphs that cite the figure) to generate QA pairs via atomic claim extraction (Eq. 2), but existing datasets like ArXivCap (Li et al., 2024) provide only figure-caption pairs (F, C). We therefore augment ArXivCap to construct triplets $(F, C, \mathcal{P})$, where $\mathcal{P}$ consists of paragraphs that explicitly cite figure F. We detail this process below.

Data Sources. We use two data sources: ArXivCap[2] (Li et al., 2024) for figure-caption pairs (F, C) and RedPajama-ArXiv-Refined[3] (Data-Juicer Team, 2023) for LaTeX source files. Matching by arXiv IDs yields 572K overlapping papers. For each matched paper, we bind its LaTeX source to the figure-caption pairs from ArXivCap. We randomly shuffle this pool and process 44,345 papers in batches to reach our target dataset size.

Licensing and Provenance. We use publicly released data artifacts and cite their creators. The dataset cards list ArXivCap under the CC BY 4.0 license and RedPajama-ArXiv-Refined under the Apache 2.0 license. Because the underlying figures and source texts originate from arXiv papers, we preserve arXiv identifiers during matching so that derived QA pairs remain traceable to their source papers and associated license metadata.

Context Extraction. For each figure-caption pair (F, C) in a matched paper, we extract its associated context $\mathcal{P}$ from the LaTeX source as follows. We first associate figure F with its corresponding figure environment by matching captions. Since ArXivCap normalizes captions using `pylatexenc`,[4] we apply the same normalization to enable efficient matching via Levenshtein similarity (Levenshtein, 1966) (sim ≥ 0.9). We then use regular expressions to extract figure labels from `\label` commands within matched figure environments (including subfigure environments). Figures with empty captions, ambiguous matches (e.g., duplicate captions), or those lacking a valid label are discarded. We then identify all paragraphs that cite figure F by searching for the extracted label in `\ref`, `\cref`, and `\autoref` commands.For figures cited in multiple paragraphs, we retain and concatenate all citing paragraphs as context $\mathcal{P}$.

A.2 Generation

Given triplets $(F, C, \mathcal{P})$ from data preparation, this stage generates candidate QA pairs through three steps: claim extraction (Eq. 2), question & answer generation (Eq. 3), and distractor generation (Eq. 4). We use `deepseek-v3-0324` (Liu et al., 2024a) as M_{text} with temperature $T = 1.0$, chosen for its instruction-following capability and cost-effective large-scale text processing. For distractor generation, we use `gpt-4o-2024-08-06` (OpenAI, 2024) as $M^{\text{gen}}_{\text{vision}}$ because it balances visual reasoning capability and API cost. All prompts are constructed using Jinja2 (Ronacher, 2008) templates; complete templates are provided in Appendix I.4.

Claim Extraction (Eq. 2). We instruct M_{text} to extract atomic claims $\mathcal{S} = \{s_j\}$ from context $\mathcal{P}$, outputting them within XML tags, with each claim s_j following the pattern "The figure shows...". If no valid claim can be extracted, the model outputs "None", and such responses are discarded.

Question & Answer Generation (Eq. 3). For each atomic claim $s_j \in \mathcal{S}$, we prompt M_{text} to generate a question Q_j and its correct answer A^*_j in XML format. The model may output "None" if s_j lacks sufficient detail; such cases are discarded.

Distractor Generation (Eq. 4). To complete the multiple-choice format, we prompt $M^{\text{gen}}_{\text{vision}}$ with figure F, caption C, question Q_j, and correct answer A^*_j to generate three plausible distractor options, yielding the full option set O_j and candidate QA pair $q_j = (Q_j, O_j, A^*_j)$.

A.3 Verification

Given candidate QA pairs from generation, this stage applies cascaded filtering to retain only instances that pass source-consistency, visual-dependence, and visual-consistency checks. Text-based filtering first ensures source-consistency and visual-dependence without accessing the figure; vision-based filtering then validates each remaining QA candidate against the figure. We use `o4-mini-2025-04-16` (OpenAI, 2025b) as $M^{\text{verify}}_{\text{vision}}$ with T=1.0, chosen for its strong scientific figure understanding on CharXiv (Wang et al., 2024).

Text-Based Filtering. The first step applies two figure-free filters without accessing figure F:

Source-Consistency Check (V_{src}). We prompt M_{text} to answer question Q_j from options O_j based solely on source context $\mathcal{P}$. The model outputs its

[2] https://huggingface.co/datasets/MMInstruction/ArxivCap

[3] https://huggingface.co/datasets/datajuicer/redpajama-arxiv-refined-by-data-juicer

[4] https://github.com/phfaist/pylatexenc

selected answer in an <option> tag, or "None" if the information is insufficient to determine a unique answer. The filter $V_{\text{src}}(Q_j, O_j, A_j^*, \mathcal{P})$ returns **True** only if the model's selection exactly matches A_j^*, and **False** otherwise (including "None", multiple answers, or cases where a valid answer cannot be extracted).

Visual-Dependence Check ($V_{\text{vis_dep}}$). To reduce computational cost, we implement this check in two cascaded steps. First, QA pairs are evaluated by `deepseek-v3-0324` given only Q_j, O_j, and caption C; only candidates where it fails to select A_j^* proceed. Second, remaining candidates are validated by $M_{\text{vision}}^{\text{verify}}$. The filter $V_{\text{vis_dep}}(Q_j, O_j, A_j^*, C)$ returns **True** only if $M_{\text{vision}}^{\text{verify}}$ also fails to select A_j^*, indicating that Q_j requires visual information from figure F beyond caption C.

Vision-Based Filtering ($V_{\text{vis_con}}$). We prompt $M_{\text{vision}}^{\text{verify}}$ with figure F, caption C, question Q_j, and options O_j to select an answer with reasoning. We perform 3 independent queries and apply 2-of-3 majority voting; candidates with ties are discarded. The filter $V_{\text{vis_con}}(Q_j, O_j, A_j^*, F, C)$ returns **True** only if the majority-voted option matches A_j^*. For retained QA pairs, we store the reasoning from the first query that agrees with the majority answer.

B Dataset Taxonomies

This section provides detailed definitions and complete distribution tables for the three key dimensions of VERISCIQA: figure types, question types, and scientific domains. Scientific domain labels are assigned according to the arXiv category taxonomy (arXiv, 2025) of the source papers, while figure types and question types are automatically annotated using `GPT-4o` (OpenAI, 2024) with carefully designed prompts (see Appendix I.4). To verify the GPT-4o labels, we randomly sample 50 examples for each figure-type and question-type category for human review; human agreement with GPT-4o labels is at least 96% across all figure-type categories and at least 86% across all question-type categories.

B.1 Figure Type Taxonomy

We categorize figures in our dataset into 12 distinct types based on their visual structure and information encoding. This diversity helps models generalize across different visualization modalities and enables fine-grained performance analysis.

Figure Type Definitions. Our figure taxonomy encompasses both data visualization charts and structural representations:

- **Line Plot** (47.7%): Graphs with continuous lines showing trends, relationships, or functions over continuous variables. Examples include accuracy curves over epochs, loss curves during training, and time series with confidence intervals.
- **Composite** (14.7%): Figures containing multiple subplots/panels where different subplots belong to different visualization types (e.g., combining line plots with bar charts, or mixing scatter plots with heatmaps). Note: Figures with multiple subplots of the same type are classified into that specific type, not as Composite.
- **Diagram** (13.8%): Schematic representations including neural network architectures, system pipelines, algorithm flowcharts, geometric diagrams, and physics/engineering schematics with technical annotations.
- **Scatter Plot** (8.8%): Individual data points plotted without connecting lines, showing correlations, distributions, or clusters. Examples include t-SNE visualizations and feature space distributions.
- **Bar Chart** (5.1%): Rectangular bars comparing discrete values across categories, commonly used for method performance comparison and ablation study results.
- **Heatmap** (4.3%): Color-coded grids or matrices showing values through color intensity, including confusion matrices, attention maps, and correlation matrices.
- **Graph** (3.2%): Graph/network visualizations showing nodes and edges as graph-theoretic structures, including knowledge graphs and dependency graphs.
- **Box Plot** (0.7%): Statistical plots showing data distributions through quartiles, comparing distributions across categories.
- **Other** (0.5%): Unusual chart types not covered by standard categories, such as radar charts, parallel coordinates, and Sankey diagrams.
- **Illustration** (0.5%): Stylized drawings or artistic renderings, including computer-generated scenes and conceptual visualizations.
- **Photo** (0.5%): Real photographs captured by cameras, typically used as dataset samples or

experimental inputs.

- **Pie Chart** (0.2%): Circular charts divided into slices showing proportions or percentages of categorical data.

The complete distribution is shown in Table 5. Line plots dominate the dataset (47.7%), reflecting their prevalence in scientific publications for showing experimental results and trends. Composite figures (14.7%) and diagrams (13.8%) are also common, highlighting the complexity of scientific communication that often requires multiple visualization modalities or structural explanations.

Figure Type	Count	Percentage
Line Plot	9,662	47.7%
Composite	2,971	14.7%
Diagram	2,803	13.8%
Scatter Plot	1,787	8.8%
Bar Chart	1,031	5.1%
Heatmap	879	4.3%
Graph	655	3.2%
Box Plot	139	0.7%
Other	111	0.5%
Illustration	98	0.5%
Photo	95	0.5%
Pie Chart	41	0.2%
Total	**20,272**	**100.0%**

Table 5: Distribution of figure types in our dataset. The taxonomy covers 12 distinct visualization modalities, from standard data plots (Line Plot, Bar Chart) to structural representations (Diagram, Graph) and composite multi-panel figures.

B.2 Question Type Taxonomy

We define 5 question types based on the cognitive operations required to answer them. This diversity helps models generalize across different reasoning operations and enables fine-grained performance analysis.

Question Type Definitions. Our cognitive taxonomy categorizes questions by the primary reasoning operation required:

- **Relational** (43.5%): Identifying trends, patterns, correlations, states, or relationships between variables. Examples: "Does the loss curve show convergence?"; "What is the relationship between learning rate and accuracy?"; "Which curve corresponds to the fluid state?"

- **Comparative** (20.2%): Comparing two or more entities based on quantitative or qualitative measures (magnitude, quality, ranking). Examples: "Which model performs best?"; "Is Model A better than Model B?"; "Which dataset shows the highest accuracy?"

- **Descriptive** (18.7%): Reading or identifying a single value, label, or attribute directly from the figure. Examples: "What is the accuracy of Model A?"; "Which color represents the baseline?"; "What is the value at epoch 10?"

- **Compositional** (13.2%): Aggregation, computation, or multi-step reasoning involving multiple values. Examples: "What is the average of values A, B, and C?"; "What is the sum of all baseline scores?"; "How many models are shown in the legend?"

- **Structural** (4.4%): Understanding figure organization, architecture, flow, or structural relationships. Examples: "What component follows the attention layer?"; "How many stages are in the pipeline?"; "What is the overall architecture?"

The complete distribution is shown in Table 6. Relational questions dominate (43.5%), indicating that scientific figure understanding heavily relies on pattern recognition and relationship analysis rather than simple value extraction. This distribution reflects the analytical nature of scientific inquiry, where understanding trends and correlations is more valuable than reading individual data points.

Question Type	Count	Percentage
Relational	8,811	43.5%
Comparative	4,087	20.2%
Descriptive	3,792	18.7%
Compositional	2,683	13.2%
Structural	899	4.4%
Total	**20,272**	**100.0%**

Table 6: Distribution of question types in our dataset based on cognitive operations. The taxonomy emphasizes higher-order reasoning skills (Relational, Comparative, Compositional) over simple value extraction (Descriptive).

B.3 Scientific Domain Taxonomy

Our dataset spans 20 arXiv primary categories, covering diverse scientific fields from computer science and physics to biology and economics. This broad coverage helps models generalize across scientific domains with varying visual conventions and terminology, and enables fine-grained performance analysis.

Domain Coverage. The distribution across scientific domains is shown in Table 7. Computer Science (cs, 22.3%) and Condensed Matter Physics (cond-mat, 19.1%) are the most represented categories, followed by Mathematics (math, 13.0%) and Astrophysics (astro-ph, 12.1%). This distribution reflects both the prevalence of these fields on arXiv and their heavy reliance on visual figures for communicating results. The dataset also includes substantial representation from applied sciences (Statistics, Electrical Engineering, Quantitative Biology) and physical sciences (high-energy physics, nuclear physics, quantum physics), ensuring broad domain coverage.

ArXiv Category	Count	%
cs (Computer Science)	4,519	22.3
cond-mat (Condensed Matter)	3,862	19.1
math (Mathematics)	2,634	13.0
astro-ph (Astrophysics)	2,455	12.1
physics (Physics)	1,849	9.1
stat (Statistics)	1,443	7.1
eess (Electrical Engineering and Systems Science)	1,012	5.0
quant-ph (Quantum Physics)	639	3.2
nlin (Nonlinear Sciences)	576	2.8
q-bio (Quantitative Biology)	526	2.6
hep-ph (High Energy Physics – Phenomenology)	217	1.1
nucl-th (Nuclear Theory)	124	0.6
gr-qc (General Relativity and Quantum Cosmology)	104	0.5
hep-th (High Energy Physics – Theory)	75	0.4
q-fin (Quantitative Finance)	68	0.3
nucl-ex (Nuclear Experiment)	49	0.2
math-ph (Mathematical Physics)	45	0.2
hep-lat (High Energy Physics – Lattice)	26	0.1
hep-ex (High Energy Physics – Experiment)	26	0.1
econ (Economics)	23	0.1
Total	**20,272**	**100.0**

Table 7: Distribution of scientific domains (arXiv primary categories) in our dataset. The broad coverage across 20 categories ensures domain diversity and generalization capability.

C Filtering Analysis

In §4, we demonstrated that our Cross-Modal Verification framework produces higher-quality QA pairs compared to existing datasets through both human evaluation (§4.3) and training effectiveness (§4.4). To validate the effectiveness of each step, we conduct a small-scale human evaluation on rejected instances. Results suggest the framework effectively identifies and removes the error types (E1–E4) identified in §2.

C.1 QA Pair Generation Analysis

During the QA pair generation step, M_{text} was instructed to decline producing QA pairs from atomic claims that are too vague or lack sufficient grounding (as described in §3.2). To evaluate whether this rejection mechanism works as expected, we randomly sampled 30 rejected claims from the dataset curation process and manually evaluated them.

Evaluation Results. Among the 30 sampled rejected claims:

- **Good (7 claims, 23.3%)**: These atomic claims were actually suitable for QA generation but rejected due to M_{text} limitations. However, given the large pool of atomic claims available, this does not hinder the scalability of the framework.

- **Too Vague (23 claims, 76.7%)**: These atomic claims lacked sufficient grounding to generate valid QA pairs. They either lacked specific visual elements (e.g., data points, labels, structural relationships), or used subjective qualitative descriptors without quantitative grounding (e.g., "good results", "nearly indistinguishable"), making it difficult to construct objective, verifiable questions. See qualitative examples in Appendix E.

Discussion. The analysis reveals that the generation refusal mechanism effectively filters unsuitable claims, with approximately 76.7% of rejections being justified. While 23.3% of rejections are due to M_{text} limitations, the large pool of atomic claims ensures this does not impact the framework's ability to scale. The "too vague" cases can be attributed to authors' writing style in scientific papers, where figure descriptions are often briefly mentioned or glossed over in the main text without providing sufficient detail. When authors reference figures, they frequently use high-level or qualitative descriptions, assuming readers will examine the figure directly for specifics.

C.2 Text-Based Filtering Analysis

As described in §3.3, text-based filtering applies two checks: source-consistency and visual-dependence. To verify these filters work as intended, we randomly sampled 50 QA pairs rejected after each step and manually evaluated them.

Evaluation Results. For the 50 sampled instances rejected by the source-consistency check:

- **Valid rejection (36 instances, 72.0%)**: M_{text} was unable to uniquely identify one correct answer from the provided options based on the source context, indicating genuine inconsistencies or ambiguities in the generated QA pairs.

- **Over-filtering (14 instances, 28.0%)**: Valid QA pairs that were rejected due to M_{text} limitations or overly aggressive prompting, but should have been retained.

For the 50 sampled instances rejected by the visual-dependence check:

- **Valid rejection (33 instances, 66.0%)**: Questions that can be answered or reasonably inferred without viewing the figure, either from the caption alone or from general knowledge, correctly identifying non-visual questions (E3). Notably, some questions can be answered by reasoning through the answer options, which suggests that constructing more challenging distractor options could further improve question difficulty and visual dependence.

- **Over-filtering (17 instances, 34.0%)**: Valid visual questions that were incorrectly rejected. When we instruct M_{text} to attempt answering the question given only the caption, M_{text} sometimes makes lucky guesses on questions that genuinely require visual information, accidentally matching the correct answer A^*. This causes the filter to mistakenly classify these visual-dependent questions as answerable without the figure, leading to false rejections.

Discussion. The text-based filtering achieves 72.0% and 66.0% precision for source-consistency and visual-dependence checks respectively. The over-filtering rates (28.0% and 34.0%) reflect trade-offs in the filtering design: for source-consistency, aggressive prompting helps catch genuine inconsistencies but may be overly strict; for visual-dependence, M_{text} sometimes makes lucky guesses on genuinely visual questions. These results suggest that the text-based filtering works as intended.

C.3 Vision-Based Filtering Analysis

As described in §3.3, vision-based filtering validates that the answer is visually grounded in the figure and does not require outside knowledge, primarily targeting *incorrectly visually grounded* (E1) and *outside-knowledge* (E4) errors. To verify this filter works as intended, we randomly sampled 50 QA pairs rejected by the vision-based filtering and manually evaluated them.

Evaluation Results. The breakdown of the 50 sampled rejected instances is as follows:

- **Incorrectly visually grounded (E1) (25 instances, 50.0%):** The generated questions reference visual elements that are not present or not correctly represented in the figure. In most cases, M_{text} hallucinates details when generating questions from atomic claims. In the remaining cases, the atomic claim itself contains information that does not correspond to the visual content, representing author errors. See Figure 21 in Appendix E for an example.

- **Outside-knowledge required (E4) (16 instances, 32.0%):** The questions require information not visually available in the figure or caption. This error stems from M_{text} generating QA pairs conditioned on figure-associated context $\mathcal{P}$, which may include information not present in the figure (e.g., precise numerical values, experimental settings, or quantitative details only mentioned in surrounding text). See Figure 22 in Appendix E for an example.

- **Over-filtering due to model limitations (9 instances, 18.0%):** Valid QA pairs where the answer is visually grounded and answerable from the figure, but the model failed to provide the correct response due to its capability limitations.

Discussion. The vision-based filtering demonstrates strong precision (82.0%) in identifying genuinely problematic QA pairs, successfully catching E1 and E4 errors that slipped through text-based filtering. The relatively low over-filtering rate (18.0%) suggests that aggressive vision-based filtering is justified, as it removes 82% problematic cases while only sacrificing 18% valid instances due to $M_{\text{vision}}^{\text{verify}}$ limitations. Combined with the ablation study results (§4.5) showing substantially larger gains from vision-based filtering than from text-based filtering alone, these findings suggest that ensuring visual grounding correctness and self-containment (not requiring outside knowledge) plays a critical role in dataset quality.

D Human Evaluation Protocol

We describe the interface and annotation guidelines used in our human evaluation study (§ 4.3). We perform human evaluation on 600 QA pairs (200 each from our dataset, SPIQA (Pramanick et al., 2024), and ArXivQA (Li et al., 2024)), sampled from the training split. Ten CS graduate students with experience reading scientific papers serve as annotators. Before the formal annotation, all annotators studied detailed guidelines with anchor examples at each score level and were calibrated on 20 seed questions, proceeding only after reaching inter-annotator agreement (Krippendorff's $\alpha \geq 0.5$) on every dimension. Each QA pair is independently rated by two annotators.

D.1 Annotation Interface

We developed a web-based annotation interface that presents annotators with a figure F, caption C, question Q, answer options O, and the designated correct answer A^* (following the notation from § 3.1). For QA pairs without captions (e.g., samples from baseline datasets like ArXivQA), the caption field is not displayed. Annotators were not provided with any background information about the datasets or our project to ensure unbiased evaluation. They evaluate each QA pair along multiple dimensions based solely on the presented information. Figure 5 shows a screenshot of the interface.

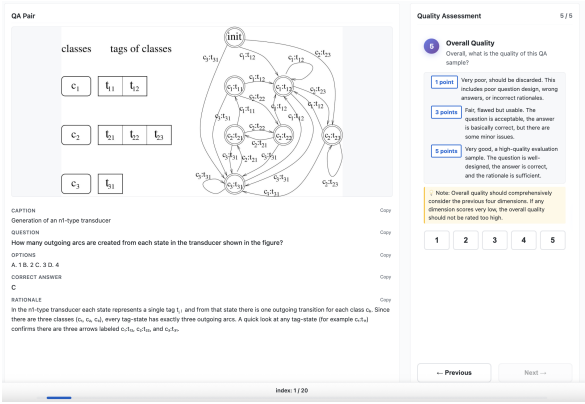


Figure 5: Screenshot of the web-based annotation interface. The left panel displays the question information (figure, caption, question, options, correct answer, and rationale), while the right panel presents a carousel of rating cards for the four quality dimensions and overall quality. Annotators rate each dimension on a 5-point Likert scale before submitting their assessment.

D.2 Annotation Guidelines

We conducted human evaluation to assess the quality of our dataset across four key quality dimensions, together with an overall quality score. This subsection presents the detailed annotation guidelines provided to human annotators.

Overview. For each VQA sample, annotators were shown the figure F, caption C, question Q, answer options O, correct answer A^*, and rationale. They were asked to rate the sample on four quality dimensions plus overall quality using a 5-point Likert scale (1 = lowest quality, 5 = highest quality). For each dimension, we provide anchor descriptions at scores 1, 3, and 5; annotators may assign intermediate scores (2 or 4) for cases falling between these anchors.

Dimension 1: Factual Correctness
Definition. To what extent is the question, answer, and rationale explicitly supported by the visual information in the figure?
Rating Criteria:

- **1 point**: Completely incorrect or contradicts the figure. This includes wrong answers, completely incorrect rationales, or hallucinated question content (asking about elements not present in the figure).
- **3 points**: Ambiguous or partially correct. The answer may be correct but the rationale is insufficient, or the answer might be correct but the evidence is unclear.
- **5 points**: Completely correct. The question is based on actual figure content, the answer is accurate, and the rationale has clear visual evidence support.

Note: Question hallucination (asking about content not present in the figure) counts as factually incorrect and should be rated 1 point.

Dimension 2: Intent Alignment
Definition. To what extent does the question reflect the core scientific information that the figure is intended to convey?
Rating Criteria:

- **1 point**: Completely irrelevant, such as treating a schematic diagram as a landscape photo. The question completely misses the scientific intent.
- **3 points**: Somewhat off-target but still within a reasonable range. The question asks about secondary information in the figure rather than its main message.

- **5 points**: Highly aligned with the scientific intent of the figure. The question asks about the core information the figure is meant to communicate.

Note: Even if the answer is correct, if the question focuses on marginal information rather than core content of the figure, it should not receive a high score.

Dimension 3: Visual Dependency
Definition. If you only see the question and options without viewing the figure, how difficult would it be to answer this question?
Rating Criteria:

- **1 point**: Very easy. The answer is common sense or already given in the question text, indicating the question does not require viewing the figure.
- **3 points**: Possibly guessable but uncertain. The question design is mediocre in requiring visual information.
- **5 points**: Absolutely impossible to answer without the figure. The question has strong dependency on visual information.

Note: High-quality VQA questions should require models to examine the figure to answer, rather than relying on guessing or background knowledge alone.

Dimension 4: Self-Containment
Definition. Does answering this question require knowledge beyond the figure, caption, and question text that is specific to this particular paper (e.g., special symbol definitions, specific experimental settings mentioned elsewhere in the paper)?
Rating Criteria:

- **1 point**: Absolutely requires paper-specific knowledge (such as symbol definitions or experimental settings defined elsewhere in the paper) to answer. The question cannot be answered without reading the full paper (unless this information is provided in the caption).
- **3 points**: Requires some paper-specific background, but can be reasonably inferred from the figure, caption, and general domain knowledge.
- **5 points**: Completely self-contained. All necessary information is provided in the figure, caption, or question text.

Note: Basic academic common sense and general domain knowledge (such as "accuracy", "temperature", and other standard concepts) are acceptable and do not count as "external knowledge". This dimension only evaluates whether paper-specific information beyond what is shown is required.

Dimension 5: Overall Quality
Definition. Overall, what is the quality of this QA sample?
Rating Criteria:

- **1 point**: Very poor, should be discarded. This includes poor question design, wrong answers, or incorrect rationales.
- **3 points**: Fair, flawed but usable. The question is acceptable, the answer is basically correct, but there are some minor issues.
- **5 points**: Very good, a high-quality evaluation sample. The question is well-designed, the answer is correct, and the rationale is sufficient.

Note: Overall quality should comprehensively consider the previous four dimensions. If any dimension scores very low, the overall quality should not be rated too high.

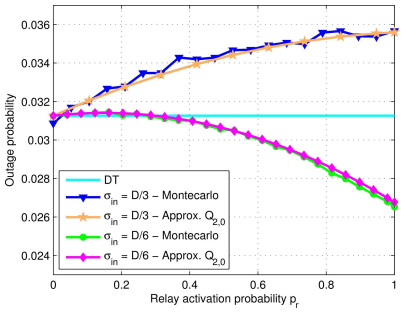
Additional analysis. The higher ratings of VERISCIQA on E1–E3 are consistent with our design choices: vision-based filtering targets factual correctness, figure-citing paragraphs improve intent alignment, and explicit visual-dependence checks improve visual grounding. SPIQA obtains a slightly higher Self-Containment score, likely because some VERISCIQA QA pairs inherit contextual dependencies from figure-citing paragraphs. Our vision-based filtering helps discard such outside-knowledge QA pairs (E4, §2), but a few such pairs persist. Inter-annotator agreement is moderate across dimensions ($\alpha = 0.60$–0.71) (Krippendorff, 2018).

E Additional Qualitative Examples

We provide additional examples to illustrate: (1) QA pairs constructed by our framework and included in VERISCIQA, and (2) error cases captured and filtered out at each step.

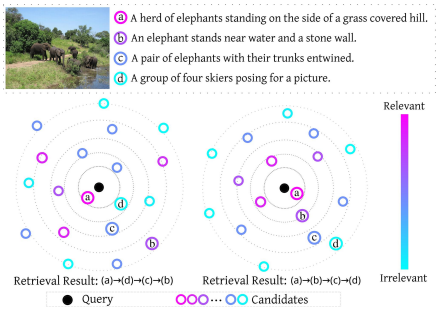
E.1 QA Pairs in VERISCIQA

We present representative examples that demonstrate the diversity and quality of VERISCIQA across different figure types and question complexities.



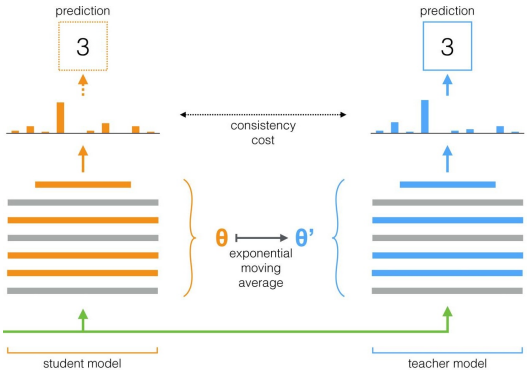
Question: At what relay activation probability does the $\sigma_{in} = D/6$ scenario reach lowest outage?
Options: A. 0.2, B. 0.4, C. 0.6, D. 1.0
Label: C. 0.6
Rationale: The green line shows outage probability decreasing as relay activation increases, reaching the lowest point at around 0.6 before plateauing.

(a) E1: Incorrectly visually grounded



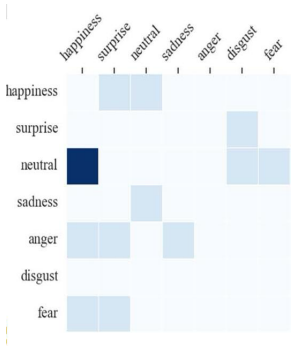
Question: What is the primary subject of the image?
Options: A. Elephants on a hill, B. Elephant near water and wall, C. Elephants with trunks entwined, D. Four skiers posing
Label: B
Rationale: The image shows a single elephant near water with a stone structure. Choices A and C are incorrect because they imply multiple elephants. Choice D is irrelevant as it doesn't feature skiers.

(b) E2: Figure-intent misaligned



Question: What does "exponential moving average" refer to?
Options: A. Smoothing technique for trends, B. Calculating median values, C. Forecasting stock prices, D. Averaging chemical mixtures
Label: A
Rationale: The 'exponential moving average' is a commonly used technique in time series analysis to smooth out data.

(c) E3: Non-visual



(c)

Question: Which emotion pair is least likely to co-occur?
Options: A. Happiness-surprise, B. Anger-disgust, C. Sadness-neutral, D. Disgust-fear
Label: D
Rationale: The lightest square is between disgust and fear, suggesting these emotions are least likely to co-occur.

(d) E4: Outside-knowledge

Figure 6: Illustrative examples of the four common error categories (E1–E4).

Example 1

Figure: 7

Caption: From left to right, the conceptual structures of CALPA-XuNet2, XuNet2, SRNet, and CALPA-SRNet* are illustrated respectively. The number and corresponding shrinking rate of output channels of every convolutional layer is shown alongside the representing bar. For SRNet and CALPA-SRNet, blue "L1" to "L12" represent twelve composition blocks following the notations in cite.

Question: In the SRNet structure shown in the figure, which blocks use transformed shortcut connections?

Options:

- A. L1 and L2
- B. L3 and L4
- C. L9 and L10
- D. L11 and L12

Answer: C

Reasoning: In the SRNet sketch only the residual blocks where the channel-width steps up require a 1×1 convolution in the shortcut (shown as a dashed "transformed" arrow and a striped conv block). These occur at L9 (16→64 channels) and L10 (64→128 channels). The earlier blocks all use

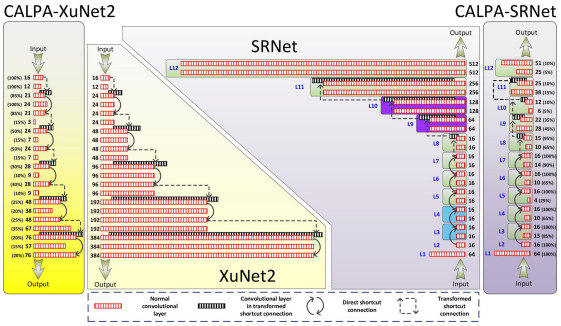


Figure 7: Figure for QA Pair in VERISCIQA

direct (identity) shortcuts, and L11/L12 have no transformed shortcut in SRNet. Hence the transformed shortcuts appear only at L9 and L10.

Example 2

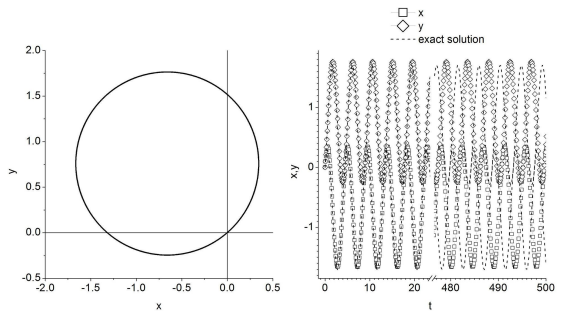


Figure 8: Figure for QA Pair in VERISCIQA

Figure: 8
Caption: Simulated evolution of the particle using our numerical integrator constructed from a DSOCS for $k = 1$, $x(t) = y(t) = 0$ and $\dot{x}(0) = \dot{y}(0) = 1$. Constant time step used: $h = 0.1$. LEFT: trajectory on the plane, RIGHT: comparison between our approximation and the exact solutions of x and y over two time intervals
Question: At what time does the maximum difference between the simulated evolution and the exact solution occur in the figure?
Options:
- A. Near t=100
- B. Near t=300
- C. Near t=500
- D. Near t=700

Answer: C
Reasoning: In the comparison plot on the right, the numerical markers (squares and diamonds) almost exactly follow the dashed exact curves at early times, but by the end of the simulation (around t=500) the dashed and marker curves are visibly out of phase, showing the largest accumulated error. No data are shown near t=700, so the maximum difference clearly occurs near t=500.

Example 3

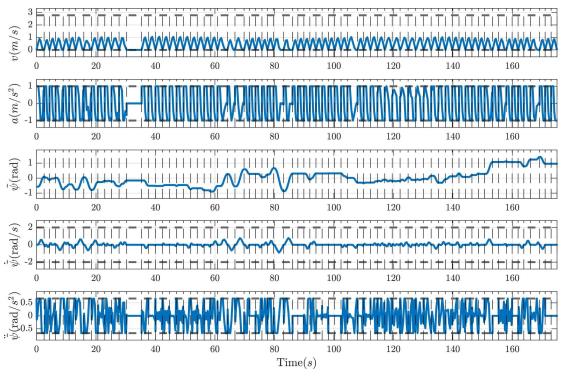


Figure 9: Figure for QA Pair in VERISCIQA

Figure: 9
Caption: Motion trajectory generated by iteratively solving OCP of each segment.
Question: What is the approximate total time taken to complete the motion trajectory shown in the figure?
Options:
- A. ≈ 90 seconds
- B. ≈ 180 seconds
- C. ≈ 270 seconds
- D. ≈ 360 seconds

Answer: B
Reasoning: The horizontal axis of the trajectory plots spans from 0 up to just under 180 s (with the final tick at 160 s and the trace ending around 170 s), so the total maneuver duration is on the order of 180 seconds.

Example 4

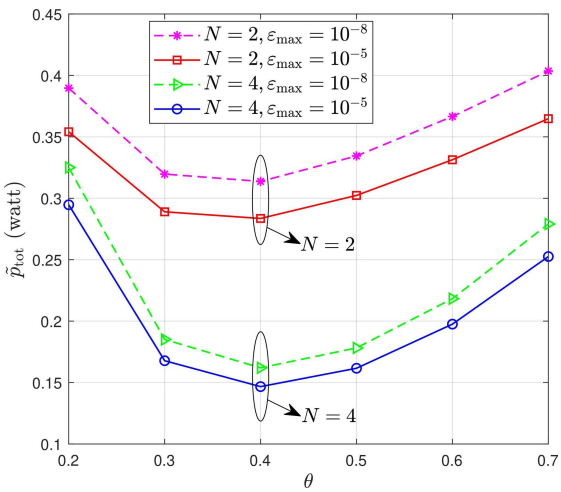


Figure 10: Figure for QA Pair in VERISCIQA

Figure: 10
Caption: $\tilde{P}_{\text{tot}}$ versus θ under different values of N and $\varepsilon_{\max}$ for $\theta = 0.5, K = 5, B = 1000$ bits.
Question: At which values of θ does the figure show that the required power consumption is high?
Options:
- A. $\theta = 0.1$ and $\theta = 0.3$
- B. $\theta = 0.2$ and $\theta = 0.7$
- C. $\theta = 0.4$ and $\theta = 0.6$
- D. $\theta = 0.5$ and $\theta = 0.8$

Answer: B
Reasoning: The plot of total power $\tilde{P}_{\text{tot}}$ versus θ shows a U-shaped curve for all curves: power is high at the two ends of the θ range (around $\theta = 0.2$) and rises again at the upper end ($\theta = 0.7$), with its minimum near $\theta = 0.4$. Hence, the required power is highest at $\theta = 0.2$ and $\theta = 0.7$.

Example 5

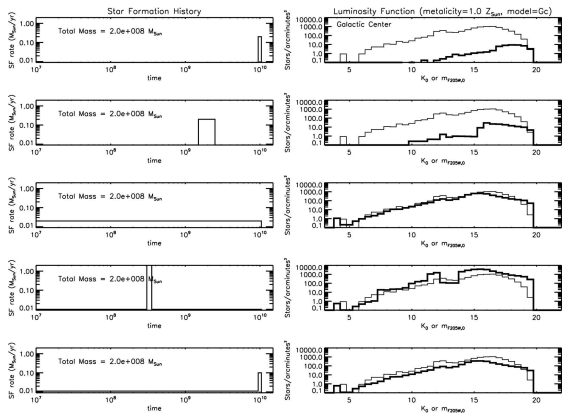


Figure 11: Figure for QA Pair in VERISCIQA

Figure: 11
Caption: A figure adapted from prior work showing various star formation scenarios (left), and resultant model luminosity functions (right, thick) compared to observed luminosity functions (right, thin) in the GC. The models assume a Salpeter IMF slope, an elevated lower-mass turnover of 10, and are additionally constrained to produce 2×10^8 in stars within the region. The observations have been corrected for incompleteness. The third panels from the top, i.e., continuous star formation, best fit the data. The observed turn-down at the faint end appears to be real and is well fit only by assuming a very high lower mass turnover.
Question: At what dereddened K-band magnitude does the observed luminosity function of the field in the GC show a turn-down?
Options:
- A. Greater than 12
- B. Greater than 14
- C. Greater than 16
- D. Greater than 18

Answer: C
Reasoning: In the luminosity-function panels, the observed (thin) counts rise up to $K_0 \approx 15$ and then begin to fall off at fainter magnitudes. The drop-off ("turn-down") occurs just beyond $K_0 \approx 16$, matching the continuous-star-formation model only if one imposes a high low-mass cutoff. Thus the observed LF turns down at dereddened K-band magnitudes greater than 16.

Example 6

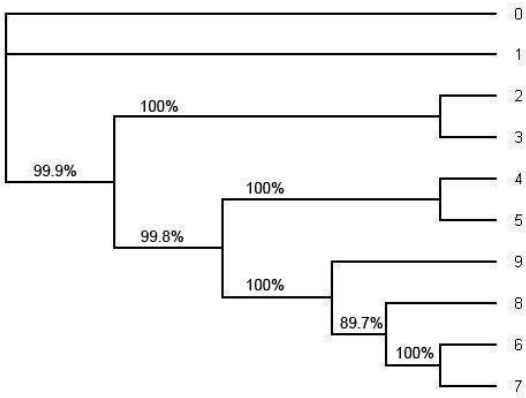


Figure 12: Figure for QA Pair in VERISCIQA

Figure: 12
Caption: The ML tree estimated by the software PHYML under the HKY model from the whole alignment (including positions 1 through 1505). This is an unrooted tree. The number in each split represents the probability of the split estimated by bootstrapping with a bootstrap sample size of 1000.
Question: What is the approximate support value for the split with the lowest probability in the ML tree under the HKY model?
Options:
- A. 50%
- B. 70%
- C. 90%
- D. 100%

Answer: C
Reasoning: The bootstrap support values on the tree are 99.9%, 100%, 99.8%, 100%, 100%, 89.7%, and 100%. The lowest of these, 89.7%, is approximately 90%.

Example 7

Figure: 13

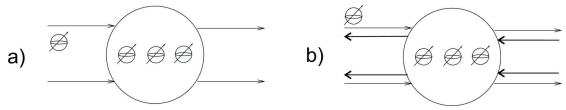


Figure 13: Figure for QA Pair in VERISCIQA

Caption: Two simple DQTA
Question: According to the figure, how many qubits are inside the circle for a segment of TM with n tape cells?
Options:
- A. n
- B. $2n$
- C. $3n$
- D. $4n$

Answer: C
Reasoning: In the diagram the internal "circle" block for a segment shows three qubit-wires per tape cell (symbol, head-position, and state register). Thus for n tape cells one needs $3 \cdot n$ qubits inside the circle.

Example 8

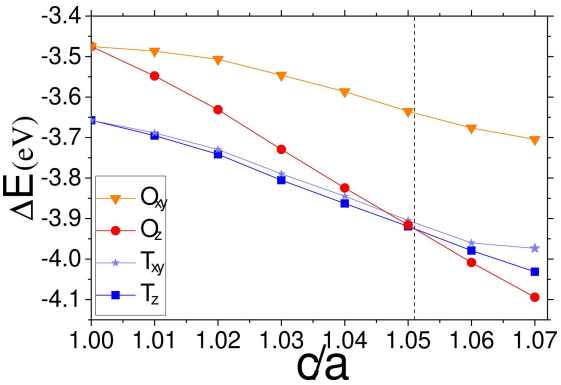


Figure 14: Figure for QA Pair in VERISCIQA

Figure: 14
Caption: (Color online) Energy difference as a function of an externally applied global uniaxial lattice strain c/a where $\Delta E = E(V+H) - E(V)$. The dashed vertical line at $c/a = 1.051$ marks the critical uniaxial lattice strain for which hydrogen occupancy of T_z and O_z sites becomes energetically equivalent.
Question: Under uniaxial strain ($c/a > 1$), which hydrogen site becomes energetically favored according to the figure?
Options:
- A. T_z
- B. O_z
- C. T_{xy}
- D. O_{xy}

Answer: B
Reasoning: From the plotted ΔE curves, as soon as c/a exceeds unity (and in particular beyond the 1.05 crossing), the red-circle curve (the O_z site) drops below all others, indicating the lowest formation energy and hence the most favorable site under uniaxial expansion.

Example 9

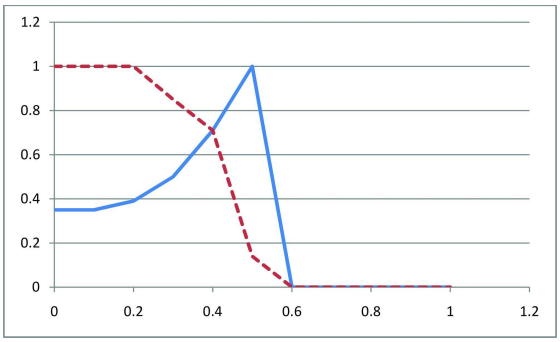


Figure 15: Figure for QA Pair in VERISCIQA

Figure: 15
Caption: How precision (blue/solid) and recall (red/dashed) change with different threshold values.
Question: What is the optimal threshold value (γ) at which the model achieves the highest possible balance between precision and recall?
Options:
- A. 0.2
- B. 0.4
- C. 0.6
- D. 0.8

Answer: B
Reasoning: The precision (solid blue) and recall (dashed red) curves intersect at $\gamma \approx 0.4$, both being around 0.7, which provides the best balance between them.

Example 10

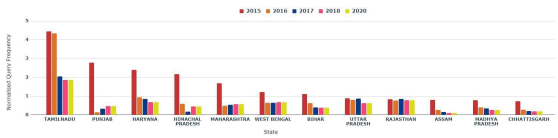


Figure 16: Figure for QA Pair in VERISCIQA

Figure: 16
Caption: Figure showing normalized pest queries received from different states of India, for different years. Normalization is performed by dividing the

frequency of queries from a given district by the gross cropped area in a given year.
Question: Which state receives the highest number of normalized pest queries according to the figure?
Options:
- A. Tamil Nadu
- B. Punjab
- C. Maharashtra
- D. Karnataka

Answer: A
Reasoning: Across all years shown, Tamil Nadu consistently has the tallest bars (peaking around 4.4 in 2015), far exceeding the normalized query frequencies of Punjab, Maharashtra, or any other state. This indicates that Tamil Nadu receives the highest number of normalized pest queries.

E.2 Error Cases Captured by Framework

We present representative examples of error cases captured and rejected at each step of our framework instantiation. These examples illustrate how each step identifies and removes specific error types (E1–E4).

Rejected during Generation: Too Vague Claims

These examples show atomic claims $s_j \in \mathcal{S}$ extracted from figure-associated context $\mathcal{P}$ (Eq. 2) that lack sufficient grounding to formulate valid QA pairs. M_{text} declined to produce q_j for these claims (Eq. 3), preventing low-quality candidates from entering the verification stage.
Claim 1.1 (s_j): "The figure `fig:ssl_workflow` shows the workflow of self-supervision."
Claim 1.2 (s_j): "The figure `fig:toy_example` shows an illustration of the principle with a toy-example."
Rejection Reason: Both claims merely state what type of content the figure depicts (a workflow, an illustration) without mentioning any concrete details that could form the basis of a verifiable question.

Rejected by Source-Consistency Check

These examples show candidate QA pairs $q_j = (Q_j, O_j, A_j^*)$ that were rejected because M_{text} could not uniquely identify the designated correct answer A_j^* when provided with only the source context $\mathcal{P}$ and question Q_j. The filter $V_{\text{src}}(Q_j, O_j, A_j^*, \mathcal{P})$ returned **False**, indicating inconsistencies between the generated QA pair and the source context.
Error Case 2.1
Figure: 17

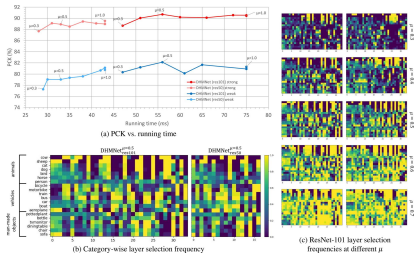


Figure 17: Figure for QA Pair filtered by Source-Consistency filtering

Caption: (color online) (a) Extinction probability P_{ext} as a function of mobility M for different competition rates p. As M increases, there is a transition from stable coexistence ($P_{ext} = 0$) to extinction ($P_{ext} = 1$). Panels (b) and (c) depict typical snapshots after a long relaxation period of the system for $p = 1.0$ and $p = 10.0$, respectively, at $M = 1.0 \times 10^{-4}$. (d) Phase diagram depicting the critical mobility M_c as a function of the competition rate p, separating the absorbing single-species phase (uniformity) and the biodiversity phase. Random initial conditions, $N = 128^2$, and $q = 1.0$ were used for results in all panels.
Source Context: It shows that smaller selection rates in training lead to faster running time in testing, at the cost of some accuracy, by encouraging the model to select a smaller number of layers. The selection rate μ can thus be used for speed-accuracy trade-off.
Question: According to the figure, what is the trade-off between layer selection rate μ and model performance?
Options:
- A. Higher μ increases accuracy but reduces running time
- B. Lower μ increases accuracy but reduces running time
- C. Lower μ reduces accuracy but increases running time
- D. Higher μ reduces accuracy but increases running time

Answer: A
Rejection Reason: The answer contradicts the source context. According to the context, smaller selection rates lead to faster running time. Therefore, higher μ should *increase* running time (slower), not reduce it as stated in option A. **Error Type: E1 (incorrectly visually grounded)**.
Error Case 2.2
Figure: 18
Caption: Proposed parsing network architecture

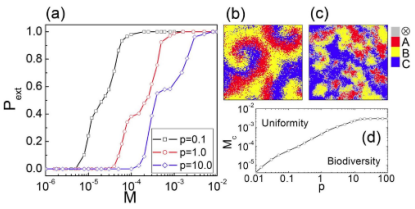


Figure 18: Figure for QA Pair filtered by Source-Consistency filtering

by combining a CNN and a spatial RNN. The CNN generates a coarse label map (b) and a recurrent gate (c), which are fed into 4 RNNs with different directions to generate a more accurate result (d). The network structure is shown where the notation for Conv1 "5×5×16/1" means convolution layer with 5×5 kernel, 16 channels and stride 1. The face image in (d) is further segmented with detailed labels in the second stage.

Source Context: From `Fig. \ref{random}(a)`, one can also observe that the critical value of M_c depends on p; in particular, smaller p yield smaller values of M_c.

Question: According to the figure, what is the relationship between the competition rate p and the critical mobility M_c?

Options:

- A. Larger p yields larger M_c
- B. Smaller p yields larger M_c
- C. M_c is independent of p
- D. p and M_c follow a U-shaped relationship

Answer: B

Rejection Reason: The answer directly contradicts the source context. The context states that "smaller p yield smaller values of M_c", but option B claims that "Smaller p yields larger M_c". This is a clear inconsistency. **Error Type: E1 (incorrectly visually grounded)**.

Rejected by Visual-Dependence Check

These examples show candidate QA pairs that were rejected because $M_{\text{vision}}^{\text{verify}}$ could correctly identify A_j^* using only caption C and question Q_j, without accessing figure F. The filter $V_{\text{vis_dep}}(Q_j, O_j, A_j^*, C)$ returned **False**, indicating **non-visual questions (E3)** that do not require visual information from the figure.

Error Case 3.1

Figure: 19

Caption: Analysis of layer selection on PF-PASCAL dataset (a) PCK vs. running time with varying selection rate μ (b) Category-wise layer selection frequencies (x-axis: candidate layer index, y-axis: category) of the strongly-supervised model with different backbones: ResNet-101 (left) and ResNet-50 (right) (c) ResNet-101 layer selection frequencies of strongly (left) and weakly (right) supervised models at different layer selection rates μ. Best viewed in electronic form.

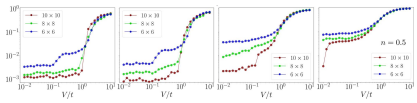


Figure 19: Figure for QA Pair filtered by Visual-Dependence filtering

Question: How does the order parameter behave when the system transitions from the metallic phase to the charge-ordered phase?

Options:

- A. It decreases gradually
- B. It remains constant
- C. It abruptly increases
- D. It oscillates randomly

Answer: C

Rejection Reason: The question can be answered correctly from the caption alone, without visual inspection of the figure. The caption explicitly states that the figure shows the transition from metallic phase to charge-ordered phase, and based on general physics knowledge about phase transitions, the order parameter typically increases abruptly during such transitions. **Error Type: E3 (non-visual question)**.

Error Case 3.2

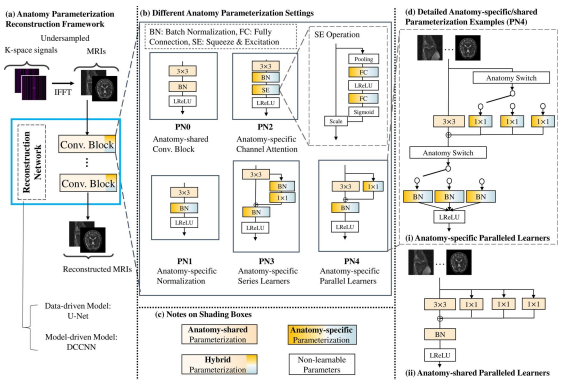


Figure 20: Figure for QA Pair filtered by Visual-Dependence filtering

Figure: 20

Caption: Order parameter as a function of the coupling constant. Each panel shows the order parameter for system sizes $L = 6$, $L = 8$, and $L = 10$, as indicated by different colors in the legend. Lines

connecting dots are for visual guidance. Different panels correspond to different fillings.
Question: In the Anatomy-specific Normalization strategy (PN1), which layers are adapted to the unique intensity distribution of a specific anatomy?
Options:
- A. Convolutional layers
- B. Batch Normalization (BN) layers
- C. LeakyReLU layers
- D. Residual connections

Answer: B
Rejection Reason: The question can be answered from the caption and figure labels alone. The caption explicitly mentions that "Anatomy-specific parameterization is highlighted in yellow", and the PN1 diagram clearly labels the yellow-highlighted layer as "BN". No detailed visual analysis is required. **Error Type: E3 (non-visual question)**.

Rejected by Vision-Based Filtering

These examples show candidate QA pairs that were rejected because $M_{\text{vision}}^{\text{verify}}$ failed to select the designated correct answer A_j^* when provided with figure F, caption C, and question Q_j. The filter $V_{\text{vis_con}}(Q_j, O_j, A_j^*, F, C)$ returned **False**, eliminating **incorrectly visually grounded (E1)** and **outside-knowledge (E4)** errors.
Error Case 4.1 (Incorrectly Visually Grounded)

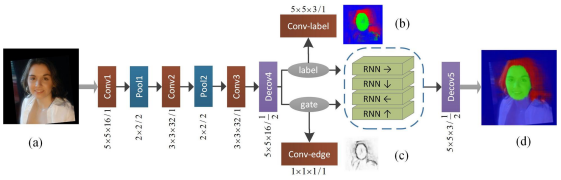


Figure 21: Figure for QA Pair filtered by Vision-Based filtering

Figure: 21
Caption: Illustration of (a) Anatomy parameterization reconstruction framework for multiple anatomy collaborative learning. Our framework can be generalized to both data-driven and model-driven models. (b) Details of four anatomy parameterization settings. (c) Notes on shading boxes. (d) Anatomy-specific parameterization example with parallel learners setting (d-i) and its anatomy-shared variant (d-ii) for the ablation study.
Question: What is the final number of output channels for Deconv6 in the proposed network architecture?
Options:
- A. 8
- B. 16
- C. 32
- D. 3

Answer: D
Rejection Reason: The question asks about "Deconv6", but this layer does not exist in the figure. This error occurred because the atomic claim extracted from the paper's context did not accurately correspond to the visual content of the figure, representing an author error that was successfully captured by the vision-based filter. **Error Type: E1 (incorrectly visually grounded)**.
Error Case 4.2 (Outside Knowledge Required)

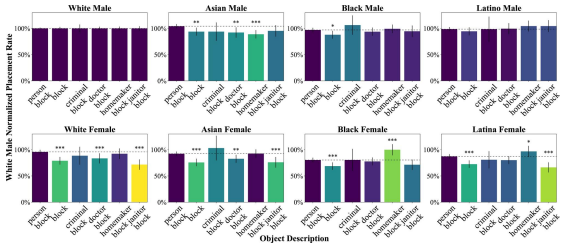


Figure 22: Figure for QA Pair filtered by Vision-Based filtering

Figure: 22
Caption: Per-command White Male normalized object placement rate across combination pairs of ethnicity and gender. The purpose of this chart is to highlight within-identity shifts in placement rate as command strings change. Error bars are corrected 95% confidence intervals. Asterisks indicate the significance of normalized placement rate differences compared to the baseline object descriptor, "person block". One star $*$ means significant, $p < 0.05$; $**$ means very significant, $p < 0.01$; $***$ means extremely significant, $p < 0.001$; and a blank entry means not significant.
Question: When the robot is asked to select a "criminal block", how much more often does it choose the block with the Black man's face compared to selecting a "person block"?
Options:
- A. 5%
- B. 10%
- C. 15%
- D. 20%

Answer: B
Rejection Reason: Although the figure shows bar charts with relative heights, the y-axis only provides coarse gridlines at 0%, 50%, and 100%. To determine the precise 10% difference between two bars, one would need to read exact numerical val-

ues from the paper's data tables or supplementary materials. The visual information alone is insufficient for accurately answering this quantitative question. **Error Type: E4 (outside-knowledge)**.

F Model Details

This section provides details on the models evaluated in our difficulty assessment (§ 4.2). Table 8 summarizes the access methods for each model.

Table 8: Model configurations used in our experiments.

Model	Version	Access
OpenAI o3	`o3-2025-04-16`	OpenAI API
GPT-5.2	`gpt-5.2-2025-12-11`	OpenAI API
o4-mini	`o4-mini-2025-04-16`	OpenAI API
Gemini-3-Flash	`gemini-3-flash-preview`	Google AI Studio
Qwen3-VL-235B	2025-10-04	Bailian API
InternVL3.5-38B	2025-08-26	vLLM Inference

Inference Configuration. We use greedy decoding for all models except `Gemini-3-Flash` and `Qwen3-VL-235B-A22B-Thinking`, whose official documentation warns that greedy decoding may cause unexpected behavior such as endless output loops; for these two models, we adopt their recommended settings. Specifically, for `OpenAI o3`, `GPT-5.2`, `o4-mini`, and `InternVL3.5-38B`, we use greedy decoding (temperature $= 0$). For `Gemini-3-Flash`, we use the recommended temperature of 1.0, as the official documentation states that "setting [temperature] below 1.0 may lead to unexpected behavior, such as looping or degraded performance." For `Qwen3-VL-235B-A22B-Thinking`, we use the model card's recommended parameters for thinking mode: temperature $= 0.6$, top-$p = 0.95$, top-$k = 20$, as the documentation warns that "greedy decoding can lead to performance degradation and endless repetitions."

G Per-Category Performance Results

This section provides per-category breakdowns for the Difficulty Assessment (§ 4.2), where we evaluated leading LVLMs in a zero-shot setting on 2,000 randomly sampled examples. Results are broken down by scientific domains, figure types, and question types. We report accuracy (%) for each category.

Model Abbreviations. In the following tables, we use abbreviated model names for space efficiency: **o3** for OpenAI o3 (OpenAI, 2025a), **GPT-5.2** for GPT-5.2 (OpenAI, 2025c), **o4-mini** for o4-mini (OpenAI, 2025b), **Gem-3F** for Gemini-3-Flash (Google DeepMind, 2025), **Qwen3** for Qwen3-VL-235B-A22B-Thinking (Team, 2025b), and **IntVL3.5** for InternVL3.5-38B (Chen et al., 2024c).

G.1 Performance by Scientific Domain

Table 9 shows model performance across 14 domain categories (13 individual arXiv primary categories plus an "Other-Domain" group covering seven low-frequency categories). We observe significant variation across domains. Proprietary models generally excel in domains such as q-bio and stat, while open-source models face the greatest challenges in Physics (physics) and Mathematics (math).

Table 9: Model accuracy (%) across scientific domains. Seven low-frequency domains (hep-th, q-fin, nucl-ex, math-ph, hep-lat, hep-ex, econ) are grouped as "Other-Domain."

Domain	o3	GPT-5.2	o4-mini	Gem-3F	Qwen3	IntVL3.5
astro-ph	73.00	52.00	61.00	76.00	68.00	56.00
cond-mat	85.00	58.00	68.00	81.00	64.00	54.00
cs	78.00	64.00	83.00	81.00	72.00	63.00
eess	80.00	56.00	83.00	82.00	70.00	56.00
gr-qc	66.67	63.00	79.00	83.00	66.00	55.00
hep-ph	80.95	60.00	73.00	83.00	62.00	52.00
math	72.00	62.00	79.00	81.00	77.00	49.00
nlin	61.29	58.00	78.00	83.00	71.00	55.00
nucl-th	77.78	53.00	78.00	82.00	63.00	50.00
physics	76.83	49.00	73.00	86.00	63.00	48.00
q-bio	87.18	48.00	74.00	89.00	68.00	53.00
quant-ph	90.24	55.00	80.00	75.00	61.00	60.00
stat	77.03	63.00	77.00	87.00	66.00	61.00
Other-Domain	70.67	55.00	79.00	85.00	61.00	54.00
Overall	**78.29**	**54.63**	**74.91**	**80.50**	**64.55**	**57.01**

G.2 Performance by Figure Type

Table 10 presents model performance across 9 figure type categories (three rare types—Illustration, Photo, and Pie Chart—are grouped as "Other"). Graph and Bar Chart figures are generally easier for models, while Scatter Plot and Composite figures pose the greatest challenges.

G.3 Performance by Question Type

Table 11 shows model performance across five question types based on cognitive operations. Structural questions are generally easiest, while Compositional questions (multi-step reasoning) are most challenging. The largest proprietary–open-source gap appears for Comparative questions.

Table 10: Model accuracy (%) across figure types, sorted by frequency in the dataset (most common first). Three rare types (Illustration, Photo, Pie Chart) are grouped as "Other."

Figure Type	o3	GPT-5.2	o4-mini	Gem-3F	Qwen3	IntVL3.5
Line Plot	78.00	53.00	77.00	83.00	60.00	57.00
Composite	79.00	51.00	73.00	79.00	67.00	56.00
Diagram	75.00	62.00	77.00	79.00	71.00	57.00
Scatter Plot	78.57	48.00	63.00	71.00	63.00	58.00
Bar Chart	81.13	64.00	74.00	80.00	72.00	62.00
Heatmap	91.07	50.00	69.00	80.00	65.00	53.00
Graph	77.19	68.00	83.00	81.00	81.00	56.00
Box Plot	78.26	61.00	81.00	87.00	70.00	52.00
Other	73.24	62.00	77.00	84.00	70.00	60.00
Overall	**78.29**	**54.63**	**74.91**	**80.50**	**64.55**	**57.01**

Table 11: Model accuracy (%) across 5 question types. Results reflect the cognitive complexity of different reasoning operations.

Question Type	o3	GPT-5.2	o4-mini	Gem-3F	Qwen3	IntVL3.5
Relational	76.00	51.00	80.00	82.00	54.00	57.00
Comparative	83.00	60.00	80.00	93.00	69.00	58.00
Descriptive	78.00	64.00	68.00	74.00	77.00	66.00
Compositional	71.00	51.00	63.00	73.00	66.00	48.00
Structural	86.36	70.00	84.00	88.00	79.00	64.00
Overall	**78.29**	**54.63**	**74.91**	**80.50**	**64.55**	**57.01**

H Training Details

This section provides additional training details not covered in § 4.4 of the main paper.

Training Data. We compare the training effectiveness of our dataset against datasets representing different construction methods: template-based SciFiBench (Roberts et al., 2024) (Figure2Caption split, 1,000 examples from General and CS categories), and LVLM-driven ArXivQA (Li et al., 2024) and SPIQA (Pramanick et al., 2024) (figure-type questions only), with 20,272 randomly sampled training examples each. For our dataset, we train two variants: one using all 20,272 samples, and one using a 1,000-example subset with the same scientific domain distribution as SciFiBench for a controlled comparison at the same scale.

Evaluation Benchmarks. All fine-tuned variants are evaluated on three external scientific figure benchmarks: CharXiv (Wang et al., 2024) (4,000 descriptive and 1,000 reasoning examples), MM-Star (Chen et al., 2024b) (science & technology subset, 250 examples), and MathVista (Lu et al., 2023) (scientific reasoning subset, 122 examples).

Base Model. We fine-tune `Qwen2.5-VL-7B-Instruct` (Team, 2025a) using LoRA (Hu et al., 2022) adapters (rank=16, α=32) applied to all linear layers, while keeping the vision encoder and alignment module frozen. We use a 9:1 train/validation split and select the best checkpoint based on validation performance.

Training Configuration. The key hyperparameters are:

- Training epochs: 1
- Learning rate: 4e-5 with cosine decay schedule
- Warmup ratio: 0.03
- Batch size: 3 per device with gradient accumulation steps of 5 (effective batch size: 60 across 4 GPUs)
- LoRA rank: 16, LoRA alpha: 32, target modules: all linear layers
- Optimizer: AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay=0.05)
- Mixed precision: bfloat16
- Maximum sequence length: 8,192 tokens
- Attention implementation: FlashAttention
- Gradient checkpointing: enabled
- Random seed: 42

Computing Resources. All experiments were conducted on 4× NVIDIA H100 GPUs using the ms-swift framework (Zhao et al., 2024).

Scalability and Cost. Scaling up our dataset is also practically feasible: on average, our framework requires approximately 59 DeepSeek-v3 calls, 13 GPT-4o calls, and 9 o4-mini calls to produce one verified QA pair; using publicly available APIs, we achieved a throughput of up to 1,200 verified QA pairs per day.

I Framework Instantiation with Data-Juicer

To ensure easy reproducibility, we re-implement our framework instance (§ 3.4) using Data-Juicer (Chen et al., 2024a, 2025), a one-stop system to process multimodal data for foundation models. We customize two new operators for Data-Juicer and orchestrate them with 7 existing built-in operators (detailed later). This Data-Juicer implementation makes our work easier to reproduce and scale to larger data sizes.

I.1 Custom Operator: LaTeX Figure Context Extractor

We customize `latex_figure_context_extractor`, a reusable Data-Juicer operator

that extracts figure-associated context $\mathcal{P}$ from LaTeX documents. Given a figure-caption pair (F, C), the operator identifies and extracts paragraphs from LaTeX source that cite figure F, providing context $\mathcal{P}$ for claim extraction (Eq. 2). The operator can be applied to any LaTeX documents with figure references.

Key Configuration Parameters:

- `caption_matching_threshold`: Caption similarity threshold for figure matching (default: 0.9)
- `citation_commands`: LaTeX reference commands to search (default: `\ref`, `\cref`, `\autoref`)
- `latex_cache_path`: Path to LaTeX source folder
- `paragraph_separator`: Pattern for delineating paragraphs (default: `\n\n`)

This operator implements the same context extraction logic as described in Appendix A.1, but in a reusable Data-Juicer operator format.

I.2 Custom Operator: Claim Extractor

We customize `claim_extractor_mapper`, a Data-Juicer operator that extracts atomic claims about a specific label from text paragraphs. Given a label (e.g., `fig:results`, `tab:comparison`) and paragraph $\mathcal{P}$ that references this label, this operator uses M_{text} to extract a set of atomic claims $\mathcal{S} = \{s_j\}$ about the referenced object (Eq. 2). The operator can be applied to any labeled objects in text (figures, tables, equations, etc.).

Key Configuration Parameters:

- `section_key`: Input field containing text paragraphs
- `label_key`: Input field containing the target label

I.3 Operator Orchestration

We re-implement the data preparation procedure detailed in Appendix A.1 and the Cross-Modal Verification framework (§ 3.4) using Data-Juicer operators. Beyond our two custom operators (§ I.1, § I.2), all other steps leverage built-in Data-Juicer operators:

Stage 1: Data Preparation.

- `remove_comments_mapper`: Removes inline and multiline LaTeX comments
- `expand_macro_mapper`: Expands LaTeX macro definitions (e.g., `\newcommand`, `\def`)
- `remove_bibliography_mapper`: Removes bibliography sections to reduce noise
- `latex_figure_context_extractor`: Extracts figure-associated context $\mathcal{P}$ (§ I.1)

Stage 2: Generation.

- `claim_extractor_mapper`: Extracts atomic claims $\mathcal{S} = \{s_j\}$ from context $\mathcal{P}$ using M_{text} (§ I.2, implements Eq. 2)
- `generate_qa_from_text_mapper`: Generates questions and correct answers (Q_j, A_j^*) from $\mathcal{S}$ using M_{text} (implements Eq. 3)
- `mllm_mapper` (distractor generation): Generates plausible distractor options using $M_{\text{vision}}^{\text{gen}}$ conditioned on (Q_j, A_j^*, F, C), yielding the full candidate QA pair $q_j = (Q_j, O_j, A_j^*)$ (implements Eq. 4)

Stage 3: Verification.

- `llm_analysis_filter` (source-consistency): Validates consistency with context $\mathcal{P}$ via V_{src}
- `llm_analysis_filter` (visual-dependence): Filters non-visual questions via $V_{\text{vis_dep}}$
- `mllm_mapper`: Generates answers using $M_{\text{vision}}^{\text{verify}}$ for each QA pair q_j
- `text_pair_similarity_filter`: Compares model answers with A_j^* via $V_{\text{vis_con}}$

Vision-Based Verification Implementation. Since Data-Juicer does not provide a built-in LVLM filter, we implement $V_{\text{vis_con}}$ through a two-step process: (1) `mllm_mapper` uses $M_{\text{vision}}^{\text{verify}}$ to generate selected options from O_j, and (2) we filter by checking if the selected option matches A_j^*.

I.4 Code Release

We release the complete Data-Juicer configuration in YAML format and the implementations of our new operators. These reusable operators are contributed to the official Data-Juicer repository, helping the community reproduce, reuse, and extend the proposed pipeline.