SciVSG: Unified Visual-Semantic Graph with Traceable Evidence for Scientific Diagram Understanding

Anonymous ACL submission

Abstract

Scientific diagrams capture complex mechanisms, experimental evidence, and structural relationships, yet remain challenging to interpret and reason over due to their heterogeneous layouts and lack of traceable representations. Scientific Visual-Semantic Graph (SciVSG) addresses this challenge by providing a unified visual-semantic representation that integrates layout and semantic information. It first constructs a Visual Layout Hierarchy (VLH) from layout cues and reading conventions, establishing a structured foundation for diagram understanding. Node-level verifiable evidence, including localized OCR and aligned paper snippets, grounds predictions to explicit text spans and regions. On this basis, a Semantic Scene Graph (SSG) is built by linking typed entities and normalized relations to nodes under strict evidence constraints, enabling module-aware reasoning and fine-grained traceability. A benchmark of diverse scientific diagrams is also provided, annotated with VLH, node evidence, entities, relations, and expert-authored QA pairs across multiple categories. Experiments demonstrate that SciVSG substantially enhances knowledge extraction and produces more reliable, evidence-attributable answers for diagram-based question answering.

Keywords: Scientific Diagram Understanding; Visual-Semantic Representation; Visual Layout Hierarchy; Semantic Scene Graph; Knowledge Extraction

1 Introduction

Scientific diagrams are essential for conveying mechanisms, experimental procedures, and structural relationships in research papers(Pan et al., 2024). They often encode actionable knowledge that researchers seek to retrieve, compare, and reuse, including workflows, component arrangements, and interaction logic (Zala et al., 2023; Pramanick et al., 2024). Despite their importance, accessing diagram-centric knowledge remains challenging due to visual heterogeneity, structural density, and modular organization with connectors and localized text (Zala et al., 2023; Kembhavi et al., 2016; Cui et al., 2025).

Scientific diagrams can be broadly categorized into four layout-driven types based on visual composition and relation expression(Mondal et al., 2024): **Process Diagrams** such as experimental pipelines, **Structural Diagrams** like part-whole schematics, **Interaction Diagrams** including regulatory networks, and **Composite Diagrams** that combine multiple modules (Zala et al., 2023). This categorization illustrates the diversity of diagram structures, highlighting the need for representations capable of capturing modularity, terminal visual units, and explicit linking cues. These diagram types serve as background context rather than the focus of the present work.

Existing methods address diagrammatic challenges unevenly (Suri et al., 2025). While regular process diagrams benefit from specialized benchmarks and QA methods (Pramanick et al., 2024), and educational datasets like AI2D (Kembhavi et al., 2016) and AI2D-RST (Hiippala et al., 2021) offer insights into element-level and discourse-inspired organization, they are not designed as general-purpose interfaces for scientific diagrams. Specifically, they lack support for module-aware extraction within nested hierarchies and fail to enforce verifiable evidence attribution linking predictions to localized regions. Similarly, conventional scene graph methods (Kembhavi et al., 2016; Hiippala et al., 2021), although effective for natural images, cannot handle the multi-module structures and dense connectors inherent in scientific contexts, thus limiting their capacity for traceable reasoning.

To bridge this gap, this work introduces **Scientific Visual-Semantic Graph (SciVSG)**, a unified

representation that combines layout and semantic information into a coherent, evidence-grounded structure. SciVSG first constructs a **Visual Layout Hierarchy (VLH)** using visual cues and reading conventions, decomposing diagrams into modules and terminal visual units. Node-level verifiable evidence, including localized OCR and aligned paper snippets, is then attached to VLH nodes, grounding predictions in explicit text and regions. Building on this foundation, SciVSG generates a **Semantic Scene Graph (SSG)** by linking typed entities and normalized relations to nodes under strict evidence constraints, producing a traceable semantic graph that enables module-aware reasoning and evidence-attributable predictions. Together, VLH and SSG form an integrated framework for structured diagram understanding.

The main contributions are as follows: (1) SciVSG, a unified visual-semantic representation combining a layout-induced hierarchy (VLH), node-attached verifiable evidence, and a node-grounded semantic graph (SSG) for scientific diagrams. (2) Traceability-oriented constraints that ensure all extracted entities, relations, and QA answers can be checked against localized regions and evidence spans. (3) A benchmark of scientific diagrams with expert-authored QA for rigorous evaluation of unified, evidence-grounded diagram understanding.

2 Related Work

Hierarchical Layout and Visual Structure. Understanding diagrams requires modeling hierarchical and compositional visual structures that reflect how elements are organized and connected (Shen et al., 2021; Zhong et al., 2019). Early resources such as AI2D provide annotated elements and parse graphs for educational diagrams, capturing relations among textual and visual components (Kembhavi et al., 2016). AI2D-RST extends AI2D with multi-layer annotations of grouping, connectivity, and discourse structure, supporting analysis of compositional layouts and visual discourse (Hiippala et al., 2021). Recent diagram parsing work emphasizes modular decomposition and structural grouping, but largely targets dataset construction or element-level parsing rather than unified, evidence-grounded semantic representations.

Graph-Based Semantic Modeling. Graph-structured representations (e.g., scene graphs) encode objects and relations for visual reasoning (Krishna et al., 2017; Zellers et al., 2018). However, they are less suited to multi-module diagrams with dense connectors and localized labels (Li and Tajbakhsh, 2023). Graph-based multimodal corpora also model discourse and connectivity among diagram elements (Hiippala et al., 2021), but typically lack typed entities and normalized relations under strict evidence constraints. SciVSG advances this line by integrating layout hierarchies, typed entities, and semantic relations into a unified, traceable graph representation.

Diagram Question Answering. Diagram QA benchmarks have grown from educational and flowchart datasets (e.g., TQA, FlowchartQA) (Kembhavi et al., 2016; Kafle et al., 2018; Tannert et al., 2023) to scientific resources. SPIQA targets figures and tables in scientific papers, evaluating interpretation of complex visual structures in research contexts (Pramanick et al., 2024). MathVerse benchmarks visual math understanding, testing whether models interpret diagrams rather than rely on textual shortcuts (Zhang et al., 2024). MISS-QA evaluates multimodal foundation models on schematic diagrams in scientific papers (Zhao et al., 2025). While these datasets support evaluation, none provide an interpretable computational framework for explicit reasoning over evidence-grounded semantic relations.

3 Layout-induced Hierarchical Diagram Representation

We present *SciVSG*, our *Layout-induced Hierarchical Diagram Representation* shown in Fig. 1. SciVSG builds a *Visual Layout Hierarchy* (VLH) to organize a diagram into localized regions with explicit containment and a reader-oriented traversal order, providing a layout-grounded backbone for subsequent understanding. To ensure traceability, each node is augmented with verifiable evidence, and we further attach typed entities and normalized relations to form a node-grounded *Semantic Scene Graph* (SSG) under strict evidence constraints. We also release a diagram benchmark annotated with VLH, node evidence, SSG, and expert-authored QA for evaluation.

3.1 Visual Layout Hierarchy

We represent a scientific *diagram* as a *Visual Layout Hierarchy* (VLH): a rooted tree that decom-

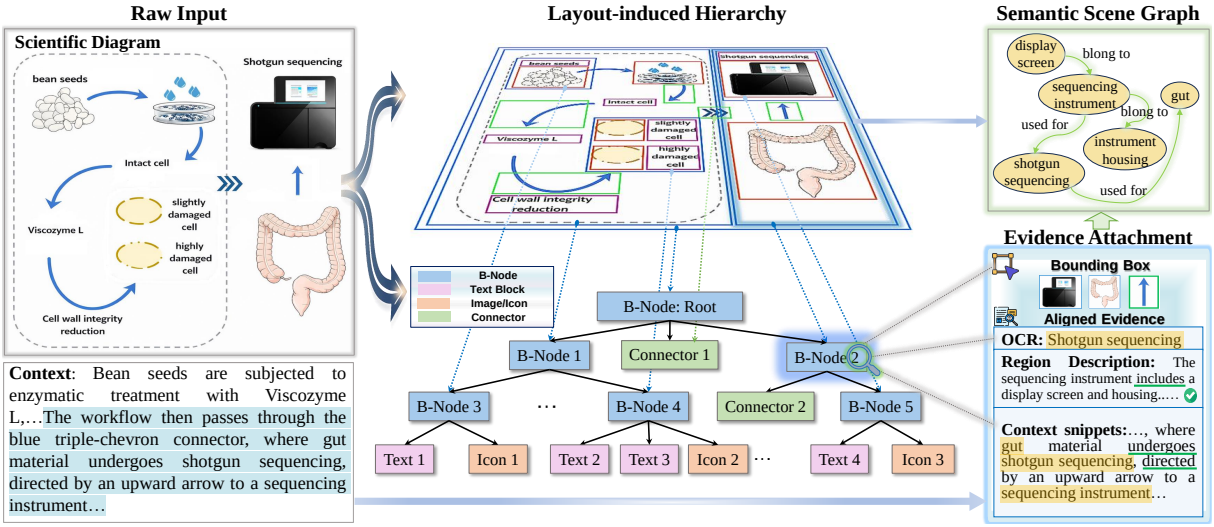

Figure 1: Layout-induced Hierarchical Diagram Representation.

poses the diagram into localized regions with explicit containment and a reader-oriented traversal order. The VLH encodes *layout* only; semantic objects (entities and relations) are attached later as node-level annotations (Section 3.2).

3.1.1 Representation Schema

A central design choice in VLH is separating *decomposable layout regions* from *terminal visual units*. `Branch` nodes represent composite regions that can be further partitioned. `TextBlock`, `Icon`, `Image`, and `Connector` are treated as terminal primitives (leaf nodes) during induction. This is a structural rather than semantic distinction. Terminal nodes are not split even when they contain multiple mentions or relations.

Each node belongs to one of five categories covering dominant diagram building blocks: **Branch node (B-Node)**: a composite module (e.g., panel/subfigure, bounded schematic module) that reduces cross-module ambiguity and supports module-level aggregation and traceable referencing. **TextBlock**: a contiguous text region (e.g., title, label cluster) serving as the primary OCR carrier. **Icon**: a non-photographic drawn region conveying schematic content, providing stable anchors for descriptions and aligned textual evidence. **Image**: a photographic/experimental region (e.g., microscopy, gels) that links visually grounded observations to aligned textual explanations. **Connector**: an explicit linking cue (e.g., arrows, lines, braces) modeled as a first-class node due to its directionality and structured associations.

For grounding and traceability, each node stores: (i) a parent–child link (containment), (ii) a bounding box, (iii) a short region description, and (iv) node-referenced text sources (localized OCR and selected caption/context snippets). How these fields are populated and aligned is described in Section 3.1.3.

VLH nodes need not be semantically atomic. We store entities as node-local mention strings (optionally typed) under the hosting node; each mention must be supported by at least one node-referenced text source, preserving traceability while keeping the layout backbone independent of the semantic layer.

3.1.2 Layout-induced Hierarchy Induction

We induce the Visual Layout Hierarchy (VLH) in a top-down manner using fixed annotation rules and reading conventions. The process is *annotation-driven*: the hierarchy is deterministically constructed from manually annotated node types and bounding boxes, without learned detectors. The VLH has no fixed depth.

Step 1: determine decomposability. Starting from the root, we assess whether the diagram contains separable modules under the schema. Visually atomic diagrams are represented as single-node trees.

Step 2: propose modules. For decomposable diagrams, we annotate perceptually coherent regions as `Branch` nodes when at least one cue is present, including boundaries, background differences, distinctive alignments, or local headers such as panel tags.

Step 3: build the branch backbone. We establish parent–child relations among `Branch` nodes by minimal geometric containment and order sib-

lings by reading conventions to obtain a deterministic traversal.

Step 4: attach leaf nodes. Terminal primitives are attached to the smallest containing `Branch` node and are not further decomposed, serving as atomic units for evidence attachment and grounding.

Overall, given annotations and fixed rules, VLH induction is deterministic: branch consolidation follows overlap resolution, parent assignment follows minimal containment, and expansion terminates at terminal primitives, yielding a stable structure for subsequent evidence alignment and semantic graph construction.

3.1.3 Node Evidence Attachment

To support extraction and reasoning, each VLH node is grounded to a region and augmented with compact, verifiable evidence fields, while preserving the VLH topology.

Bounding box. Each node stores a bounding box for localization and stable attribution.

Region description vs. context snippets. A *region description* summarizes observable visual cues (e.g., position, boundaries) and is independent of paper text, serving as an interpretable anchor and auxiliary alignment signal. A *context snippet* is a relevant caption/context span aligned to the node as textual evidence.

Localized OCR. OCR is primarily attached to `TextBlock` nodes; for `Connector` nodes we record nearby OCR fragments when available. For `Image/Icon` nodes, OCR may be sparse and aligned snippets serve as the main textual anchors.

Aligned paper snippets. We align caption/context spans using reproducible heuristics based on OCR overlap, connector-adjacent trigger words, and panel/local identifiers (e.g., "(a)", "Panel B"). `Branch` nodes may store module-level snippets, while fine-grained evidence is anchored at terminal primitives.

All evidence is stored with minimal provenance (*source*, *node_id*, exact `span`); each `span` must be an exact substring of its source (no paraphrasing). Region descriptions cannot introduce entity mentions; entities must be supported by OCR or aligned snippets. We enforce: **tree validity**, **containment consistency**, **type constraints** (no decomposition of primitives), and **evidence integrity** (exactly matchable spans), flagging violations automatically.

This step outputs a grounded VLH tree $\mathcal{T}$ with verifiable node evidence. In §3.2, we attach typed entities and normalized relations to $\mathcal{T}$ to enable module-aware reasoning with region-level attribution.

3.2 Semantic Scene Graph (SSG)

We build a *Semantic Scene Graph* (SSG) on top of the visual layout hierarchy (VLH) by attaching typed entities and normalized relations to VLH nodes. The result is a traceable semantic graph that supports module-aware reasoning and *verifiable* evidence attribution at the node (region) level.

3.2.1 Entity and Relation Inventory

An entity is a *nameable* scientific object mention. Each entity must be grounded in a mention string that can be *verbatim located* in at least one **textual** node-attached evidence field (e.g., localized OCR strings or region captions). This constraint ensures automatic verifiability and discourages hallucinated entities.

We adopt a seven-way entity type system (Hu et al., 2024): **Tool** (named instrument, device, platform, or software; e.g., *mass spectrometer*); **Material** (physical, biological, or data-carrying substance; e.g., *serum sample*); **Indicator** (measurable metric or property; e.g., *accuracy*); **Method** (approach, algorithm, or model; e.g., *Bayesian inference*); **Problem** (task, defect, disease, or phenomenon; e.g., *noise removal*); **Dataset** (uniquely identifiable named dataset; e.g., *MNIST*); **Value** (numeric or comparative numeric expression; e.g., *95%*).

A relation is a directed semantic edge between two grounded entities, drawn from a *fixed* label set. Relation labels are normalized semantic categories and need not appear verbatim in evidence; instead, evidence must support the underlying claim and be attributable to a specific node.

We define seven relation labels: **Belong To** (part–whole or membership); **Used For** (functional usage); **Measure** (measurement or evaluation); **Produce** (generation or output); **Precede** (temporal or procedural ordering); **Equal To** (aliasing or semantic equivalence); **Affect** (directed influence such as increase or suppression).

If a downstream benchmark adopts a different relation inventory, only label substitution is required; grounding and evidence rules remain unchanged.

3.2.2 Node-attached Entity Construction

SSG does not assume that VLH nodes are semantically atomic. A single node (including `TextBlock`, and nodes containing embedded labels near `Icon` or `Image` regions) may host zero, one, or multiple entity mentions. In our current data format, a node stores entity mentions as strings (with types) rather than forcing entities to be leaf nodes in the hierarchy.

Entity mentions are extracted and attached using **textual** node-attached sources only, namely localized OCR strings and aligned caption, context snippets. We enforce a *verbatim mention constraint*: every stored entity string must be exactly matchable as a substring in one of the allowed textual evidence fields of its hosting node. Notably, region descriptions are **not** used as an entity source, and cannot introduce new entity mentions.

We apply lightweight normalization (whitespace trimming and canonical punctuation where appropriate) while preserving scientific symbols and units. If the same mention appears in multiple nodes, we keep multiple grounded instances because they carry distinct spatial provenance; an optional canonical id can be added for downstream consolidation.

3.2.3 Relation Construction with Evidence Attribution

For each node v, we form candidate entity pairs within a *module-constrained* scope enabled by the VLH: (i) **connector-centric** pairs suggested by connector geometry and nearby labels, (ii) **within-node** co-occurrence pairs from entities attached to the same node, and (iii) **within-subtree** pairs whose endpoints fall under the same `Branch node`. This avoids global enumeration and reduces cross-panel ambiguity.

For each candidate (h, t), we select a relation label r from the fixed inventory using node-scoped cues (connector direction/morphology, nearby triggers, aligned caption/context, and region descriptions). Each relation must include at least one evidence record $\langle source, span, v \rangle$, where *span* is copied verbatim from an allowed node-attached source (OCR, connector-local text, region description, or aligned snippets); we enforce **exact span matching** (no paraphrasing). Evidence need not contain the label name, but must support the normalized claim. Attaching relations to VLH nodes with boxes yields a traceability chain: *relation* $\rightarrow$ *evidence span* $\rightarrow$ *node id* $\rightarrow$ *node box*.

If endpoints are grounded to different nodes, we attach the relation to the *lowest common ancestor* `Branch node` covering both, while preserving each endpoint's original node id in the evidence metadata.

3.3 Benchmark Construction (Data and QA)

We construct a benchmark to evaluate unified and traceable diagram representations across four genres: **Process**, **Structural**, **Interaction**, and **Composite** diagrams. Each sample is annotated with (i) a Visual Layout Hierarchy (VLH), (ii) node evidence fields enabling span-verifiable attribution, and (iii) a Semantic Scene Graph (SSG) with typed entities and normalized relations attached to VLH nodes. Domain experts additionally author QA pairs, which we categorize into four types with strict answer matching for automatic evaluation.

3.3.1 Diagram Collection and Coverage

We collect candidate papers from IEEE venues and arXiv, starting from approximately 30000 papers with clear diagrams and sufficient context. Manual screening of panels and captions yields 18760 diagram candidates that are readable, self-contained, and interpretable with surrounding text.

We then curate a high-quality subset suitable for annotation, retaining diagrams that meet: (i) adequate resolution for text/connectors, (ii) clear module boundaries when applicable, (iii) unambiguous reading order or structural organization, and (iv) sufficient context for evidence alignment. After filtering, we retain **1200** diagrams, including **600** English and **600** Chinese. Coverage across the four genres is summarized in Table B.1 (Appendix).

We balance the dataset across genres. Process diagrams show stepwise transitions. Structural diagrams highlight partwhole organization. Interaction diagrams feature dense, typed relations. Composite diagrams combine multiple local organizations in a single layout.

3.3.2 Annotation Protocol

Annotation layers. Each diagram is annotated in three layers.

(1) VLH structure and node boxes. Annotators construct VLH following the layout-induced rules in Section 3.1.2, assigning each node a type and bounding box. Branch nodes are decomposable modules, while terminal primitives (TextBlock, Icon, Image, Connector) are not subdivided.

(2) Node evidence fields. We attach three textual sources per node: (i) *region description* written purely from visual content, (ii) *context snippets* aligned from paper text, and (iii) *OCR text* for TextBlock nodes (and optional nearby OCR cues for Connector nodes). These fields support verifiable attribution in extraction and QA.

(3) SSG entities and relations. Annotators attach typed entities and normalized relations to VLH nodes under evidence constraints: entity mentions are copied verbatim from OCR or context snippets, and each relation is supported by at least one evidence span.

Each evidence record is stored as a four-tuple

$$\langle source, node\_id, span, char\_offset \rangle,$$

where *span* is an exact substring of the declared source (OCR/context/region description), and *char_offset* is verified by deterministic lookup.

3.3.3 Consistency and Quality Control

Each diagram is annotated by two annotators and adjudicated by a senior annotator to resolve disagreements in VLH boundaries/types, entity typing, and relation labels. We apply checks for (i) tree validity (single root, acyclic, unique parent), (ii) geometric consistency (child boxes within parent up to tolerance), (iii) connector integrity (terminal connectors with supported endpoints when applicable), and (iv) evidence integrity (exact substring spans with valid offsets).

Agreement is reported at multiple layers: mean IoU and node-type agreement for VLH/boxes; set-based Precision/Recall/F1 over unique (mention, type) pairs for entities; and set-based Precision/Recall/F1 over relation triples for relations. Representative agreement numbers are: mean box IoU **0.86**, node-type agreement **0.95**, entity F1 **0.84**, and relation F1 **0.79**.

3.3.4 Expert-authored QA and Taxonomy

Domain experts author questions using the diagram and aligned context; questions are not templated from labels. We group questions into four categories and constrain answers to be short and matchable for automatic scoring.

Q1 Entity: identify a named entity of a specified type. **Q2 Relation:** identify the normalized relation label between two named entities with traceable evidence. **Q3 Hierarchical trace:** identify the module where a claim is expressed and, if needed, return the module path from the root using matchable identifiers (e.g., panel tags or OCR fragments). **Q4 Hierarchy-helpful reasoning:** answer questions that benefit from restricting or aggregating evidence within the hierarchy.

We collect **8** questions per diagram on average, yielding **7200** QA pairs, balanced across four categories. Detailed distributions and verification constraints are summarized in Table B.2 (Appendix).

4 Experiments

4.1 Overview and Experimental Setup

We evaluate SciVSG on four targets: VLH structure extraction, SG construction, SSG construction, and diagram question answering (QA). Our benchmark covers four diagram genres: *Process*, *Structural*, *Interaction*, and *Composite*. These genres differ in module organization and in connector-mediated relations.

Why off-the-shelf foundation models without training? We validate SciVSG as a *model-agnostic intermediate representation* and test how robustly different multimodal foundation models instantiate it under the same stage-wise protocol. This setting (i) isolates the contribution of the representation and supervision interface (schema, evidence constraints, normalization) from task-specific fine-tuning, and (ii) reflects a plug-and-play deployment without additional training.

Stage-wise construction and evaluation targets. Given a diagram image and associated paper text, we run a fixed pipeline to obtain: (i) a VLH with node types and parent–child edges, (ii) an SG of semantic triplets, and (iii) an SSG that attaches typed entities and normalized relations to VLH nodes with evidence. We evaluate VLH node typing and edges, SG/SSG triplets, and QA under controlled ablations.

Cross-model evaluation (Table 1). We compare multiple multimodal foundation models with identical inputs and prompts to quantify consistency in hierarchy induction and semantic graph construction under strict matching.

Ablations (Table 2). Using a single model, we vary inputs (image; caption+context; image+caption+context) and provided representations (VLH, SG, SSG) to isolate component effects.

Traceability analysis. A visualized QA case study in Appendix Figure B.1 shows that VLH enables precise back-tracing from answers to supporting nodes and evidence.

4.2 Metrics

We evaluate (i) **VLH structure extraction** with Node-type macro-F1 and Edge-F1, (ii) **SG/SSG construction** with set-based triplet Precision/Recall/F1, and (iii) **diagram QA** with BLEU and ROUGE-L.

Notation. Let a diagram have a gold VLH with node set $\mathcal{N}$ and directed parent–child edge set $\mathcal{E}$. Let $\hat{\mathcal{N}}$ and $\hat{\mathcal{E}}$ denote predicted nodes and edges. Let $\mathcal{T}^{SG}$ and $\hat{\mathcal{T}}^{SG}$ be gold and predicted SG triplet sets, and $\mathcal{T}^{SSG}$ and $\hat{\mathcal{T}}^{SSG}$ be gold and predicted SSG item sets.

Exact matching and normalization. For graph items containing text fields (heads/tails in SG/SSG), we normalize mentions by (1) trimming leading/trailing whitespace and (2) collapsing multiple spaces into one, while keeping case and punctuation unchanged. All set intersections below use *exact* equality after this normalization.

Unified set-based Precision/Recall/F1. For any gold set A and predicted set $\hat{A}$, we compute

$$P(A,\hat{A}) \coloneqq \frac{|\hat{A}\cap A|}{|\hat{A}|}, \tag{1a}$$

$$R(A,\hat{A}) \coloneqq \frac{|\hat{A}\cap A|}{|A|}, \tag{1b}$$

$$F1(A,\hat{A}) \coloneqq \frac{2\,P(A,\hat{A})\,R(A,\hat{A})}{P(A,\hat{A})+R(A,\hat{A})}. \tag{1c}$$

To provide a more comprehensive evaluation of the diagram parsing performance, we apply this unified metric to evaluate the various structural dimensions of the diagrams. Specifically, we report Node-F1 and Edge-F1 to assess node typing and edge recovery within the Visual Layout Hierarchy (VLH). For semantic triplet extraction in both Scene Graphs (SG) and Situated Scene Graphs (SSG), we report the full suite of Precision, Recall, and F1-score. Detailed formal definitions for each task-specific instantiation are provided in Appendix A.

4.3 Cross-model Evaluation on Structure, SG, and SSG Construction

Table 1 summarizes module-wise evaluation across foundation models for VLH structure extraction, SG construction, and SSG construction.

VLH is relatively stable across models. Structure extraction yields comparatively high Node-type F1 and Edge-F1 across models, indicating that layout-induced hierarchy can be reliably induced from visual layout cues and reading conventions.

Graph construction remains the bottleneck. SG F1 is consistently lower than VLH scores, reflecting the difficulty of extracting correct semantic relations among heterogeneous terminal units. SSG is further harder than SG because it requires both semantic correctness and correct node attachment under a normalized inventory.

SSG exposes grounding failures beyond semantic errors. The gap between SG and SSG indicates additional error modes, including incorrect assignment of entity mentions to nodes, attachment inconsistency across modules, and relation normalization mistakes. These results motivate a controlled study of how different inputs and intermediate representations affect downstream QA.

4.4 Ablations over Raw Inputs and Representation Variants

Table 2 reports ablations over raw inputs and representation variants using a single fixed model.

Effect of raw inputs. Caption plus context improves language-heavy questions because many answer strings are explicitly stated in the paper text. Image-only inputs remain competitive when questions depend on visual layout cues, module boundaries, and connector-mediated transitions. Combining image with caption plus context yields the best overall performance, showing that visual structure and textual evidence are complementary.

Effect of VLH. Providing VLH improves Q3 and Q4 because the hierarchy constrains the candidate space to the relevant module and reduces cross-module confusion when similar tokens recur in multiple regions.

SG versus SSG. SG helps relation-centric questions by explicitly exposing candidate triplets, but it can still mix instances across modules when the structure is implicit. SSG strengthens consistency by enforcing node attachment and normalized relations, which is especially beneficial for questions that require traceability and module-consistent grounding.

4.5 Summary

Overall, the experiments show that a reliable layout hierarchy forms a strong backbone, while semantic graph construction remains the primary challenge. The ablations confirm that VLH contributes by enforcing hierarchical constraints

Table 1: **Module-wise evaluation across foundation models.** We report Node-type F1 and Edge-F1 for VLH structure extraction, and Precision / Recall / F1 for SG construction and SSG construction.

Model \ Metrics	Structure extraction (VLH)		SG construction (SG)			SSG construction (SSG)		
	Node-F1↑	Edge-F1↑	P↑	R↑	F1↑	P↑	R↑	F1↑
GPT-5-mini	0.861	0.413	0.546	0.482	0.514	0.699	0.636	0.668
GPT-5-nano	0.712	0.345	0.538	0.474	0.506	0.685	0.622	0.654
GPT-5.1	**0.936**	**0.619**	**0.586**	**0.518**	**0.552**	**0.748**	**0.676**	**0.712**
Gemini-3	0.921	0.618	0.532	0.468	0.499	0.674	0.611	0.643
Qwen3-VL-235B	0.420	0.310	0.540	0.476	0.508	0.681	0.618	0.649
Qwen2.5-VL-32B	0.743	0.367	0.387	0.310	0.349	0.415	0.333	0.374
DeepSeek-VL2	0.280	0.220	0.333	0.258	0.296	0.402	0.317	0.360
Claude Haiku 4.5	0.858	0.407	0.555	0.489	0.522	0.701	0.636	0.669

Table 2: **Ablations over raw inputs and representation variants.** We use a single model and report **BLEU** and **ROUGE-L** by question type (Q1–Q4) and the macro average (AVG).

	Raw inputs		Representation			BLEU by question type					ROUGE-L by question type				
Setting	**Image**	**Cap+Ctx**	**VLH**	**SG**	**SemSG**	**Q1**	**Q2**	**Q3**	**Q4**	**AVG**	**Q1**	**Q2**	**Q3**	**Q4**	**AVG**
S1	✓	–	–	–	–	0.58	0.79	0.32	0.67	0.59	0.76	0.83	0.53	0.81	0.73
S2	✓	–	✓	–	–	0.89	0.91	0.14	0.62	0.64	0.92	0.89	0.38	0.75	0.73
S3	✓	–	–	✓	–	0.90	0.62	0.12	0.61	0.56	0.92	0.96	0.38	0.76	0.75
S4	✓	–	✓	✓	–	0.90	0.28	0.14	0.67	0.50	0.91	0.97	0.39	0.78	0.76
S5	✓	–	✓	–	✓	**0.92**	**0.94**	**0.35**	**0.66**	**0.72**	**0.89**	**0.89**	**0.56**	**0.81**	**0.79**
S6	–	✓	–	–	–	0.28	0.87	0.18	0.41	0.44	0.46	0.91	0.39	0.60	0.59
S7	–	✓	✓	–	–	0.79	0.91	0.09	0.59	0.60	0.85	0.87	0.37	0.69	0.70
S8	–	✓	–	✓	–	0.77	0.96	0.13	0.60	0.62	0.79	0.93	0.38	0.73	0.71
S9	–	✓	✓	✓	–	0.84	0.99	0.12	0.56	0.62	0.87	0.97	0.37	0.72	0.73
S10	–	✓	✓	–	✓	**0.85**	**0.98**	**0.39**	**0.65**	**0.72**	**0.87**	**0.97**	**0.55**	**0.75**	**0.79**
S11	✓	✓	–	–	–	0.61	0.92	0.37	0.66	0.64	0.79	0.89	0.55	0.82	0.76
S12	✓	✓	✓	–	–	0.86	0.94	0.14	0.59	0.63	0.90	0.91	0.38	0.73	0.73
S13	✓	✓	–	✓	–	0.86	0.96	0.16	0.62	0.65	0.89	0.91	0.39	0.76	0.74
S14	✓	✓	✓	✓	–	0.90	0.97	0.17	0.61	0.66	0.93	0.94	0.39	0.75	0.75
S15	✓	✓	✓	–	✓	**0.91**	**0.98**	**0.38**	**0.72**	**0.75**	**0.93**	**0.96**	**0.57**	**0.80**	**0.82**

and module locality, and SSG further improves faithfulness through node-attached semantics and evidence-consistent grounding. These findings support SciVSG as a practical intermediate representation for diagram-centric knowledge extraction and question answering.

5 Conclusion

We tackle scientific diagram understanding, where complex layouts and weak grounding make extraction and reasoning hard to verify. We propose SciVSG, a unified visual-semantic representation that combines a layout backbone with node-grounded semantics: it first builds a Visual Layout Hierarchy (VLH) from layout cues and reading order, then attaches verifiable node evidence (localized OCR and aligned caption/context snippets), and finally constructs a node-grounded Semantic Scene Graph (SSG) with typed entities and normalized relations under strict evidence constraints. Experiments on our benchmark show that SciVSG improves diagram knowledge extraction and produces more reliable, evidence-attributable answers for diagram QA.

Limitations

SciVSG still has limitations, notably it does not directly handle charts, which require specialized parsing of axes, scales, and data series. Looking forward, two practical directions are as follows.

(i) building robust detectors for the basic visual units defined by SciVSG to enable scalable, automatic construction;

(ii) strengthening cross-source grounding by tighter alignment between figure regions and paper text/tables, supporting broader coverage and more complete scientific interpretation.

Acknowledgments

This paper used large language models (e.g., ChatGPT) to assist data annotation during the image understanding pipeline, and we sincerely appreciate their support.

References

Zhiqing Cui, Jiahao Yuan, Hanqing Wang, Yanshu Li, Chenxu Du, and Zhenglong Ding. 2025. Draw with thought: unleashing multimodal reasoning for scientific diagram generation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM 25, pages 5050–5059. Association for Computing Machinery.

Tuomo Hiippala, Malihe Alikhani, Katri Haverinen, Arne Köhn, Stina Ojala, Achim Stein, and Sanni Uusi-Mäkelä. 2021. Ai2d-rst: A multimodal corpus of 1,000 primary school science diagrams. *Language Resources and Evaluation*, 55(3):661–688.

Maodi Hu, Li Qian, Zhijun Chang, and Zhixiong Zhang. 2024. Kdpg-enhanced mrc framework for scientific entity recognition in survey papers. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 32:25322543.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. DVQA: Understanding data visualizations via question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5648–5656.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *Computer Vision – ECCV 2016*, pages 235–251, Cham. Springer International Publishing.

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

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

Ishani Mondal, Zongxia Li, Yufang Hou, Anandhavelu Natarajan, Aparna Garimella, and Jordan Lee Boyd-Graber. 2024. Scidoc2diagrammer-maf: towards generation of scientific diagrams from documents guided by multi-aspect feedback refinement. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13342–13375. Association for Computational Linguistics.

Huitong Pan, Qi Zhang, Cornelia Caragea, Eduard Dragut, and Longin Jan Latecki. 2024. Flowlearn: evaluating large vision-language models on flowchart understanding. *arXiv preprint arXiv:2407.05183*.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2024. SPIQA: A dataset for multimodal question answering on scientific papers. In *Advances in Neural Information Processing Systems*, volume 37, pages 118807–118833. Curran Associates, Inc.

Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. 2021. LayoutParser: A unified toolkit for deep learning based document image analysis. In *Document Analysis and Recognition – ICDAR 2021*, pages 131–146, Cham. Springer International Publishing.

Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A. Rossi, Vivek Gupta, and Dinesh Manocha. 2025. Follow the flow: fine-grained flowchart attribution with neurosymbolic agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22485–22508. Association for Computational Linguistics.

Simon Tannert, Amir Sadafi, and Klemens Böhm. 2023. Flowchartqa: The first large-scale benchmark for reasoning over flowcharts. In *Proceedings of the 1st Workshop on Linguistic Insights from and for Multimodal Language Processing*.

Abhay Zala, Han Lin, Jaemin Cho, and Mohit Bansal. 2023. Diagrammergpt: Generating open domain diagrams via diagram understanding and text-to-diagram generation. arXiv preprint arXiv:2310.12128. ArXiv:2310.12128.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

R. Zhang, D. Jiang, Y. Zhang, and 1 others. 2024. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, ECCV, pages 169–186, Cham. Springer Nature Switzerland.

Yilun Zhao, Chengye Wang, Chuhan Li, and Arman Cohan. 2025. Can multimodal foundation models understand schematic diagrams? an empirical study on information-seeking qa over scientific papers. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18598–18631, Vienna, Austria. Association for Computational Linguistics.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. PubLayNet: Largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022.

A Appendix A: Evaluation Metrics

In this section, we provide the formal mathematical definitions for the metrics discussed in Section 4.2. Following the unified Precision/Recall/F1 framework defined in Eq. 1, we specify the truth sets (A) and predicted sets ($\hat{A}$) for each task-specific instantiation to ensure a comprehensive evaluation of both structural and semantic parsing.

VLH Node-type macro-F1. We measure node typing as a per-class set matching problem. For each node type $c \in \mathcal{C}$, define

$$A_c \coloneqq \{n \in \mathcal{N} : y(n) = c\}, \quad \hat{A}_c \coloneqq \{n \in \mathcal{N} : \hat{y}(n) = c\}. \tag{2}$$

We compute $F1_c = F1(A_c, \hat{A}_c)$ by Eq. (1c) and report the macro average:

$$\text{Node-F1} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} F1_c. \tag{3}$$

VLH Edge-F1. Edge-F1 evaluates recovery of directed parent–child relations. Let $A = \mathcal{E}$ and $\hat{A} = \hat{\mathcal{E}}$; Edge-F1 is $F1(A, \hat{A})$ by Eq. (1c).

SG Triplet-F1. Each SG item is a triplet (h, r, t). Let $A = \mathcal{T}^{SG}$ and $\hat{A} = \hat{\mathcal{T}}^{SG}$; SG Precision/Recall/F1 are computed by Eq. (1) under exact triplet matching after mention normalization.

SSG Triplet-F1 with node attachment. Each SSG item is (h, r, t, u_h, u_t), where u_h and u_t are the VLH node ids to which the head and tail mentions are attached. Let $A = \mathcal{T}^{SSG}$ and $\hat{A} = \hat{\mathcal{T}}^{SSG}$; SSG Precision/Recall/F1 are computed by Eq. (1). This metric is stricter than SG because a prediction is correct only when both the semantic triplet (h, r, t) and the node attachments (u_h, u_t) match exactly.

B Appendix B: Visual Layout Hierarchy Induction Algorithm

Algorithm 1 Layout-induced VLH construction (rule-based, annotation-driven)

Require: Diagram image I, annotated regions $\mathcal{R}$ with types and boxes, fixed decision rules $\mathcal{D}$
Ensure: VLH tree $\mathcal{T}$

$\mathcal{P} \leftarrow \{r \in \mathcal{R} \mid \text{type}(r) \in \{\texttt{TextBlock}, \texttt{Icon}, \texttt{Image}, \texttt{Connector}\}\}$
$\mathcal{G} \leftarrow \{r \in \mathcal{R} \mid \text{type}(r) = \texttt{Branch node}\}$
if IsAtomicDiagram($\mathcal{P}, \mathcal{D}$) **then**
    Create a single root node v with $\text{type}(v) \in \{\texttt{Image}, \texttt{Icon}\}$ covering the diagram
    return $\mathcal{T} \leftarrow \{v\}$
end if
Create root branch node G_0 covering the full diagram; initialize $\mathcal{T} \leftarrow \{G_0\}$
$\mathcal{G} \leftarrow$ MergeAndResolveConflicts($\mathcal{G}, \mathcal{D}$) ▷ merge near-duplicates, resolve heavy overlaps by fixed rules
BuildBranchBackbone($\mathcal{T}, G_0, \mathcal{G}, \mathcal{D}$) ▷ assign parent by minimal geometric containment
for all $p \in \mathcal{P}$ **do**
    $a \leftarrow$ SmallestContainingBranch($p, \mathcal{G} \cup \{G_0\}$)
    Add p as a child of a in $\mathcal{T}$
end for
if NoReliableBranchNodes($\mathcal{G}, \mathcal{D}$) **then**
    Remove internal branch node structure; attach all $p \in \mathcal{P}$ directly under G_0
end if
Stop rule: never expand terminal primitives ▷ VLH depth is not fixed; expansion ends at primitives
return $\mathcal{T}$

Table B.1: Benchmark coverage across four diagram genres. Counts are curated from an initial pool of 18760 candidates extracted from approximately 30000 papers.

Diagram genre	English	Chinese	Total	Typical cues
Process Diagrams	170	170	340	stage blocks, ordered arrows, step titles
Structural Diagrams	155	155	310	part to whole nesting, component labels, callouts
Interaction Diagrams	135	135	270	relation-dense arrows, typed links, direction cues
Composite Diagrams	140	140	280	multi-module layout, panel boundaries, mixed substructures
Total	**600**	**600**	**1200**	

Table B.2: Question answering statistics and verification constraints. The reported counts are placeholders and will be updated with the final benchmark release.

QA category	Per-diagram avg.	Total	Answer form	Verification rule
Q1 Entity	2.0	2400	entity mention string	exact match to entity and evidence span
Q2 Relation	2.0	2400	relation label	exact match to relation and evidence span
Q3 Hierarchical trace	1.0	1200	module identifier plus path	exact substring match in allowed sources, full path completeness
Q4 Hierarchy helpful	1.0	1200	short answer or matchable span	exact substring match in allowed sources, module-consistent evidence
Total	**8.0**	**7200**		

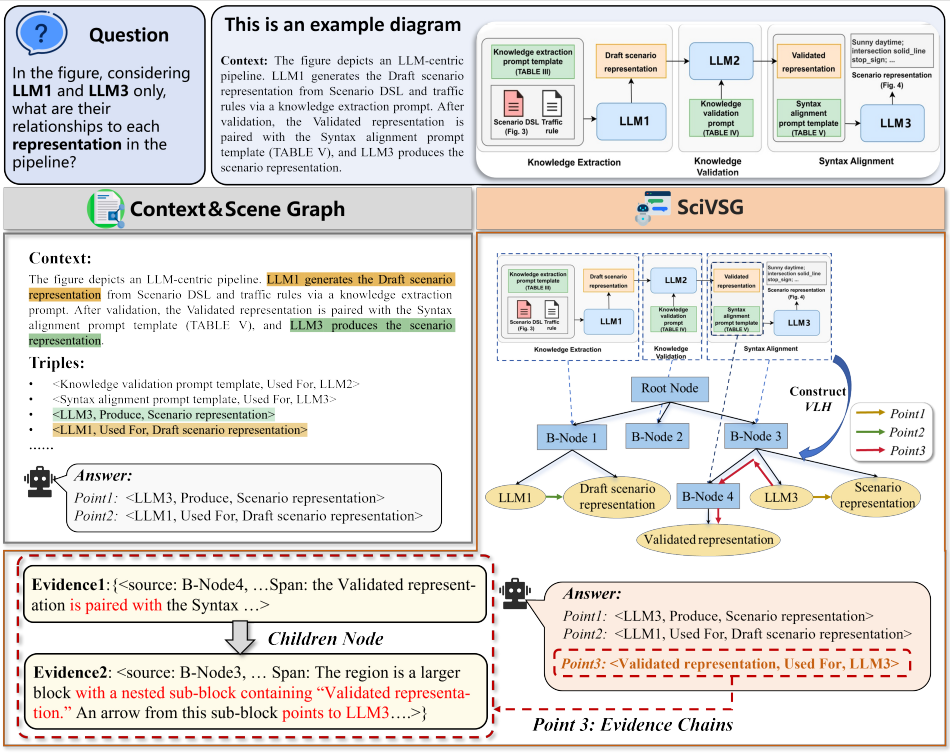

Figure B.1: Illustration of the traceable QA process in SciVSG. **Qualitative comparison:** We compare a Context & Scene Graph representation with SciVSG on an example diagram-based QA instance. Under the same question, the context-plus-graph setting recovers only Point1 and Point2, supported by textual cues and the corresponding extracted relations. In contrast, SciVSG additionally enables Point3 by exploiting the module-level hierarchy induced from diagram layout, which constrains reasoning to the correct structural scope and exposes a module-specific dependency missed by the baseline. Moreover, SciVSG offers fine-grained traceability: each predicted point is linked to the relevant VLH module, localized to the corresponding diagram region, and justified by an explicit evidence span aligned from node-attached text sources.