Evaluating Vision–Language Models on Structural Reasoning in Entity–Relationship Diagrams

Anonymous ACL submission

Abstract

Vision-language models (VLMs) are increasingly used to interpret diagrams embedded in documents, slides, and online content, yet it remains unclear whether they can reliably reason over structured visual languages. We study this question for Entity-Relationship Diagrams (ERDs), where correctness depends on both geometric structure and formal design rules. Although modern VLMs can often verbalize ERD constraints in text, they frequently fail to apply the same rules when reasoning over diagram images. To evaluate this gap, we introduce **ERClass**, a benchmark for structural error classification in ERDs that measures visually grounded rule application rather than question answering or graph reconstruction. ERClass contains 8,940 diagrams generated by injecting 11 structural error types into 745 clean ERDs. Commercial and open-weight VLMs consistently perform much better on shape-based violations than on connection-based violations: commercial models achieve 0.73 macro-F1 on shape recognition but only 0.31 on connection recognition, while a symbolic computer-vision baseline outperforms most of the VLMs except GPT-5.3 on connection reasoning. These results suggest that current VLMs lack robust visual grounding for structured diagram reasoning, particularly for topological connectivity. Code, dataset, and evaluation pipeline are available at https://anonymous.4open.science/r/ERClass-2D28.

1 Introduction

Structured diagrams are widely used to represent relational and procedural information in domains such as software engineering, database design, scientific communication, and education (Barwise and Etchemendy, 1995; Shimojima, 2015; Jamnik, 2001). Although such diagrams are often created using specialized tools that maintain native symbolic or graph-based representations, they are frequently shared and consumed as visual artifacts embedded in documents, presentation slides, screenshots, social media posts, or online discussions (Hu et al., 2024; Arbaz et al., 2024; Petrovic et al., 2025). In these settings, the original structured representation is typically unavailable, preventing traditional symbolic tools from directly verifying or processing the underlying semantics.

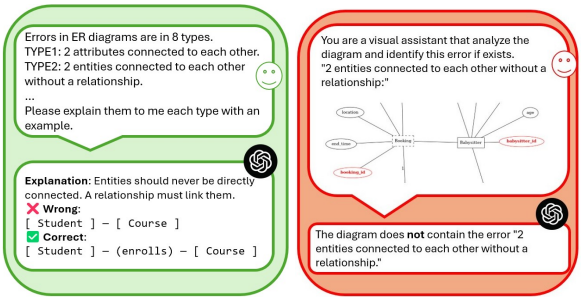

Figure 1: Illustration of the gap between verbal description and visual error identification. The model explains an ERD violation in text but fails to localize the corresponding error in the diagram.

Entity-Relationship Diagrams (ERDs) are an important example of this problem. ERDs are central to database design and encode schema semantics through a compact visual language with explicit geometric and topological constraints (Larkin and Simon, 1987). Unlike many general diagram-understanding tasks, ERDs provide deterministic structural correctness criteria: entities, relationships, attributes, and their connections must satisfy well-defined visual and topological rules. While database-design tools can deterministically verify ERD correctness when symbolic representations are available, such verification becomes substantially more difficult once diagrams are encountered only as rendered images embedded in documents or online content.

Recent advances in vision-language models (VLMs) have positioned them as increasingly general-purpose systems for document and visual-content understanding (Liu et al., 2024; Bai et al.,

2025; Achiam et al., 2023). Modern multimodal systems can interpret figures, tables, charts, and diagrams directly from images, making them natural candidates for ERD understanding and structural verification. However, correctly interpreting ERDs requires more than semantic recognition: models must distinguish geometric primitives, trace visual connections, and verify whether the recovered topology satisfies formal ERD constraints. Misidentified shapes, incorrectly traced edges, or hallucinated relationships can therefore produce structurally invalid interpretations even when the model appears semantically confident.

Figure 1 illustrates this challenge. Although the model correctly explains an ERD rule violation in text, it fails to identify the corresponding structural error in the diagram itself. This reveals a disconnect between declarative rule knowledge and visually grounded structural verification.

Prior work shows that VLMs struggle with geometric and connectivity understanding in isolation (Hou et al., 2024; Rahmanzadehgervi et al., 2024; Kamoi et al., 2024; Rudman et al., 2025). However, these studies do not directly evaluate whether models can apply formal diagrammatic rules to visual structure. Motivated by recent work emphasizing diagnostic task formulations for diagram understanding (Suri et al., 2025; Mancoridis et al., 2025), we introduce ERClass, a benchmark for structural ERD error classification. Rather than evaluating question answering or graph reconstruction, ERClass measures visually grounded structural verification: models must determine whether a diagram violates a structural constraint, identify the violation type, and localize the responsible visual element or invalid connection.

ERClass contains 8,940 diagrams generated from 745 clean ERDs with 11 controlled structural error types. We evaluate recent commercial and open-weight VLMs under prompt-based inference and observe a consistent gap between shape recognition and connection verification. Across models, connection-related violations remain substantially harder than shape-based violations, and a symbolic computer-vision baseline outperforms most of the evaluated VLMs on connection recognition despite using no language priors. Through controlled random-label, label-permuted, paired, and few-shot analyses, we further show that models frequently rely on semantic plausibility rather than grounded topology extraction.

Our main contributions are the following:

- We introduce ERClass, a benchmark for structural ERD error classification that evaluates whether models can apply formal diagrammatic constraints directly to visual structure.
- We construct a controlled benchmark of 8,940 diagrams generated from 745 clean ERDs with 11 structural error types spanning both shape- and connection-based violations.
- We evaluate recent commercial and open-weight VLMs under prompt-based inference and show a consistent gap between shape recognition and connection verification, with connection-related violations remaining substantially more difficult.
- Through random-label, label-permuted, paired, and few-shot analyses, we show that current VLMs rely heavily on semantic plausibility while struggling to robustly ground diagram topology through explicit edge tracing.

Together, our results suggest that structured diagrams provide a useful setting for evaluating grounded structural verification in multimodal systems, particularly for tasks requiring reliable topology extraction and edge-level visual grounding.

2 Related Work

Recent work evaluates VLMs across a wide range of diagram and document-understanding tasks. General-purpose multimodal benchmarks assess visual comprehension across diverse image types (Zhang et al., 2024a), while large-scale studies continue to report substantial gaps between VLMs and human performance (De Bari et al., 2024). More specialized benchmarks study reasoning over flowcharts (Pan et al., 2024; Arbaz et al., 2024; Suri et al., 2025), scientific figures and technical documents (Hu et al., 2024; Ye et al., 2024), engineering schematics (Munde, 2025), and relational schemas for hallucination detection in LLMs (Oh et al., 2024). Existing diagram benchmarks, however, primarily focus on downstream tasks such as question answering, captioning, graph reconstruction, or image-to-code generation. While useful, these tasks often entangle visual perception, language generation, and task-specific reasoning, making it difficult to isolate the underlying source of model failures. Recent work such as FollowTheFlow (Suri et al., 2025) therefore argues for more diagnostic task formulations that expose hallucinated reasoning paths and grounding failures in diagram understanding systems.

A growing body of work further suggests that

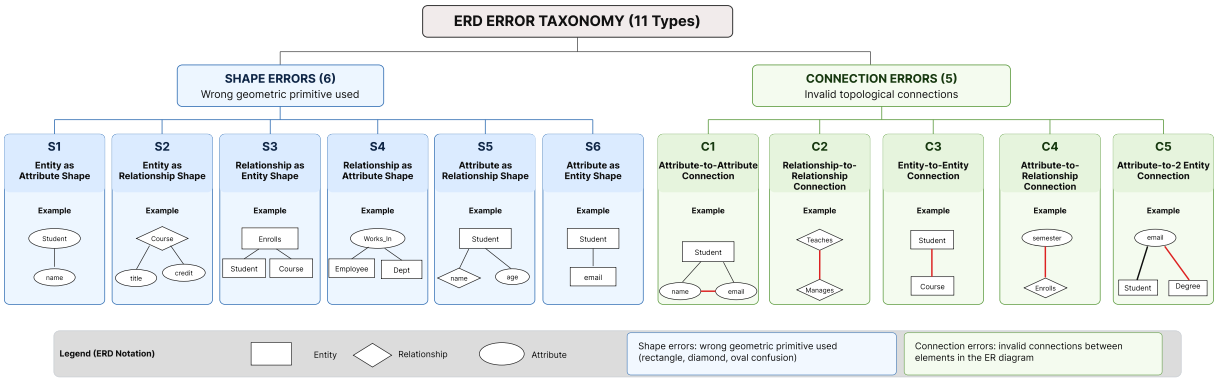

Figure 2: Taxonomy of eleven ERD error types. Shape errors (S1–S6) replace an ERD element with the wrong geometric primitive, while connection errors (C1–C5) introduce invalid topological links between elements (shown in red). Each diagram contains exactly one injected error.

VLMs frequently rely on pattern matching rather than grounded visual reasoning. Potemkin understanding (Mancoridis et al., 2025) shows that models can verbalize concepts correctly while failing to apply them in grounded settings. Similar limitations have been observed in spatial and causal reasoning (Hou et al., 2024), mathematical reasoning (Zhang et al., 2024b), and visually grounded QA (Kamoi et al., 2024). Models also remain weak at geometric and topological operations such as shape recognition (Rudman et al., 2025), geometric reasoning (Alonso et al., 2025), and graph or line connectivity inference (Rahmanzadehgervi et al., 2024; Li and Jiang, 2025). Prior work on diagrammatic representation further emphasizes that structured visual languages combine symbolic meaning with spatial and topological structure (Barwise and Etchemendy, 1995; Shimojima, 2015). Similar challenges arise in circuit diagrams (Mehta et al., 2024), chemical structure diagrams (Burgess et al., 2025), and mathematical proof diagrams (Zhang et al., 2024b), where correct interpretation depends not only on recognizing objects and labels, but also on extracting relations among visual elements.

Our work builds on these observations by introducing structural ERD error classification as a controlled diagnostic task for diagram understanding. Unlike question answering or graph reconstruction, our formulation directly evaluates whether a model can apply formal ERD rules to visual structure. We further isolate visual connectivity reasoning through an edge-following benchmark, allowing us to disentangle semantic plausibility from grounded topology perception.

3 ERClass Dataset

To evaluate whether VLMs can reason over structured diagrammatic syntax, we construct an ERD error detection benchmark using a controlled multi-stage pipeline. The benchmark contains both structurally correct ER diagrams and error-injected diagrams. Each erroneous diagram contains exactly one injected violation, allowing us to evaluate whether models can identify not only that an error exists, but also the specific diagram element or connection responsible for it.

Stage 1: ERD Generation and Collection. We generate 200 synthetic ER scenarios using GPT across diverse domains, including sports, healthcare, travel, retail, and education. We additionally collect 148 real-world ERDs from textbooks, instructional materials, tutorial websites, and practitioner-created examples, divided into Web and Instructional subsets.

Stage 2: Formal Specification. Each diagram is converted into a structured JSON specification encoding entities, attributes, relationships, and connections. This representation enables controlled error injection and ground-truth annotation. All schemas are rendered using Graphviz (Graphviz, 2024); real-world schemas are re-rendered after error injection to control for rendering style while preserving realistic schema structure.

The resulting dataset spans a broad range of structural complexity. Synthetic schemas increase approximately linearly across difficulty tiers, with average node and edge counts rising from 22.03/22.62 in the easy split to 42.19/42.58 in the hard split. Real-world schemas span a comparable range: textbook-derived ERDs contain on average 18.46 nodes and 15.46 edges, while Internet-derived schemas average 29.08 nodes and 27.98

Subset	Clean	Single Error	Dual Error	Total
Easy	199	2,189	199	2,388
Medium	199	2,189	199	2,388
Hard	199	2,189	199	2,388
Web	100	1,100	100	1,200
Instruct.	48	528	48	576
Total	**745**	**8,195**	**745**	**8,940**

Table 1: Composition of the ERClass benchmark. Each clean diagram is transformed into multiple erroneous variants using the eleven error types shown in Figure 2. Dual-error diagrams contain two simultaneous injected violations and are used for robustness evaluation. Context-free and label-permuted variants are excluded from this summary table.

edges, closely matching the medium synthetic tier.

Our goal is not to maximize rendering realism but to isolate grounded structural verification under minimally confounded visual conditions. By controlling rendering style and injecting exactly one structural violation per diagram, the benchmark establishes a lower-bound evaluation setting: if models fail to reliably trace and verify topology in clean, unambiguous ERDs, performance on noisier real-world renderings is unlikely to improve.

Stage 3: Controlled Error Injection. For each clean JSON specification, we introduce exactly one structural error from the taxonomy in Figure 2. The taxonomy contains eleven error types divided into two groups: six *shape errors* and five *connection errors*. Shape errors replace an ERD element with the wrong geometric primitive, while connection errors introduce invalid topological links between elements. Injecting errors at the specification level ensures precise ground truth and avoids visual artifacts that could reveal the injected error.

Dataset. The benchmark contains 745 clean ERDs spanning synthetic and real-world sources. Applying the eleven structural error types produces 8,195 single-error diagrams, while an additional 745 dual-error diagrams are constructed for robustness evaluation, resulting in 8,940 diagrams total.

Error Taxonomy. Figure 2 summarizes the eleven error types. Shape errors replace ERD elements with incorrect geometric primitives, while connection errors introduce invalid topological links between element types. The taxonomy is informed by prior work on common mistakes in ER modeling and conceptual database design education (Rashkovits and Lavy, 2021).

4 Experiments and Error Cause Analysis

We evaluate whether VLMs can detect and localize structural violations in ERDs. Rather than treating ERD understanding as a single flat classification problem, we decompose evaluation into five tasks targeting correctness detection, shape recognition, connection verification, full-taxonomy classification, and paired structural comparison.

4.1 Experimental Setup

We evaluate models in five ERD reasoning settings. Each task includes both erroneous and structurally correct diagrams, enabling evaluation of both false positives and missed violations.

Full Taxonomy Recognition. Models classify each diagram into the complete 12-class label space consisting of eleven error categories and one correct class.

Shape Error Recognition. Models identify one of the six shape-based error categories (S1–S6) or determine that the diagram is correct.

Connection Error Recognition. Models identify one of the five connection-based error categories (C1–C5) or determine that the diagram is correct.

Binary Error Detection. Models determine if a diagram is structurally correct or contains any form of error, independent of the specific error type.

Paired Error Recognition. Models are given both a correct ERD and a corresponding error-injected version, and must identify the injected structural modification.

Models. We evaluate recent proprietary and open-weight VLMs, including GPT (Achiam et al., 2023), Gemini (AI, 2025), Claude (Anthropic, 2025), Qwen (Tongyi Lab, 2023), Llama (Touvron et al., 2023), and Gemma (Team et al., 2024).

Evaluation Metric. We report Macro-F1 across all experiments. A prediction is counted as correct only if the model outputs both the correct error category and the corresponding localized error target in a valid structured JSON format. The required output schema consists of two fields: `error_type` (a string from the 11-class taxonomy or `correct`) and `erroneous_element` (the diagram element or invalid connection responsible for the violation).

For shape-related violations, the localized target corresponds to the modified ERD element whose geometric primitive is incorrect. For connection-related violations, the localized target corresponds to the invalid connection itself, represented by its

two endpoint elements.

Element labels are matched against ground truth using fuzzy string matching with a similarity threshold of 0.85, after lowercasing and removing spaces, underscores, hyphens, and punctuation. This normalization prevents superficial formatting differences from penalizing otherwise correct predictions. Responses that omit required fields, produce malformed JSON, predict an error type without localization, or localize the wrong element or connection are scored as incorrect for that instance. Malformed or non-parseable outputs were rare across prompting settings, indicating that observed performance differences are not primarily driven by output-format compliance.

This joint scoring protocol prevents models from receiving credit for generic "diagram is incorrect" responses without grounding the violation in the diagram itself. A random label predictor that produces no valid structured localization therefore scores 0%, rather than 50% on the binary task or $\frac{1}{12}$ on the full-taxonomy task. In addition to Macro-F1, we report confusion matrices and per-type behavior analyses to identify systematic biases and recurring failure patterns across models and error categories.

Declarative Knowledge Probe. To verify that poor performance is not caused by missing ERD knowledge, we administer a text-only probe in which models explain the eleven structural error types without diagram input. All evaluated models correctly describe the associated geometric and topological constraints, indicating that observed failures arise from grounding rather than missing declarative knowledge.

Traditional Computer Vision (CV) Baseline. In addition to VLMs, we evaluate a rule-based baseline built on OpenCV (Bradski, 2000) that detects geometric primitives and traces connections using explicit visual heuristics, with no language priors or semantic knowledge. The pipeline classifies each contour as rectangle, diamond, or oval via circularity and convex-hull geometry, maps each shape to its ERD role deterministically, and infers connection errors by checking whether adjacent node-role pairs are valid under ERD rules. An optional CLIP ViT-B/32 (Radford et al., 2021) zero-shot step resolves geometrically ambiguous contours.

We include this baseline purely as a diagnostic: it establishes how much of the task is solvable by explicit visual structure alone, without learned representations or semantic reasoning. We do not claim it generalizes beyond the controlled Graphviz rendering used in ERClass— pixel geometry and fixed thresholds are inherently resolution- and style-dependent. However, this dependency is expected and acceptable for a diagnostic baseline. The more striking result is that even this unoptimized, heuristic system outperforms most of the evaluated VLMs on connection recognition, suggesting that the bottleneck for VLMs is not the visual difficulty of the task but their inability to apply grounded topological reasoning. A robust solution will likely require combining explicit structural extraction with VLM-based semantic reasoning.

Model	Full	Shape	Conn.	Binary	Pair
Claude-4.6	0.43	0.71	0.11	0.73	0.41
Gemini-3.1	0.49	0.69	0.39	0.78	0.45
GPT-5.3	**0.62**	**0.79**	**0.42**	**0.81**	**0.54**
Gemma-3	0.1	0.06	0.05	0.40	0.03
Llama-4.0	0.00	0.03	0.00	0.35	0.01
Qwen-3.5	0.36	0.52	0.32	0.58	0.46
CV Base.	0.46	0.47	0.36	0.50	0.47
VLM Avg.	0.51	0.73	0.30	0.77	0.47

Table 2: Macro-F1 across the five ERD reasoning tasks. Bold indicates the best score in each column.

4.2 Main Results

We report the overall Macro-F1 results on the five studies in Table 2.

Full-taxonomy recognition. We begin with the full-taxonomy setting because it most closely reflects realistic ERD verification: the model must determine whether a diagram is correct and, if not, identify the exact structural violation. GPT-5.3 Chat achieves the best individual performance with 0.6177 Macro-F1, followed by Gemini-3.1 Flash Lite at 0.4887 and Claude-4.6 at 0.4384. The commercial VLM average reaches 0.5149, showing that recent proprietary models can solve a non-trivial portion of the task. However, performance remains far from reliable given that all diagrams contain controlled and unambiguous rule violations. Open-weight models perform substantially worse, with Gemma-3 at 0.0946 and Llama-4.0 at 0.0000, while Qwen-3.5 reaches 0.3646.

Shape versus connection errors. The diagnostic shape and connection studies explain the full-taxonomy results. Shape errors are substantially easier than connection errors: the commercial VLM average reaches 0.7333 on shape recognition but only 0.3085 on connection recognition. GPT-5.3 follows the same pattern, dropping from 0.7917 to 0.4196, while Claude-4.6 shows the largest gap (0.7181 vs. 0.1151). This indicates that VLMs can often recognize incorrect ERD symbols, but

struggle when the task requires tracing edges and verifying endpoints.

This gap aligns with the attention analysis in Figure 4. Shape errors produce relatively diagonal confusion matrices, whereas connection errors show substantially higher off-diagonal confusion, especially among edge-related classes. These failures are therefore not random mistakes, but systematic difficulties in edge tracing and topology grounding.

Binary detection. Binary detection is easier than full-taxonomy recognition because the model needs to determine whether some structural violation exists. GPT-5.3 reaches 0.82, and the commercial VLM average reaches 0.78. However, strong binary performance should not be interpreted as reliable ERD understanding. Models can often detect that something appears visually unusual without correctly identifying the underlying structural violation. The large gap between binary detection (0.78 commercial average) and full-taxonomy recognition (0.51 commercial average) thus shows that coarse anomaly detection is substantially easier than grounded structural diagnosis.

CV baseline comparison. The CV baseline uses explicit geometric processing rather than language or semantic priors. It remains competitive in the dual-error setting (0.493 mean Macro-F1 versus 0.525 for commercial VLMs) and outperforms most of the models on connection recognition in the generated-diagram benchmark. This supports our central claim: VLMs benefit from semantic and visual priors, but lack robust inductive bias for edge tracing and topological verification. The strongest future direction is therefore likely a hybrid approach combining explicit visual structure extraction with VLM-based semantic reasoning.

4.3 Structural Analysis

Paired recognition. The paired setting provides the correct ERD alongside the corrupted diagram, removing ambiguity about the intended schema, notation, and layout. This improves performance only partially. GPT-5.3 reaches 0.5465, while the commercial VLM average reaches 0.4726. The fact that paired recognition remains substantially below shape recognition and close to full-taxonomy recognition suggests that the primary bottleneck is not simply identifying what the diagram should look like. Even when a correct reference diagram is available, models still struggle to localize the exact structural modification, particularly for connection-related violations.

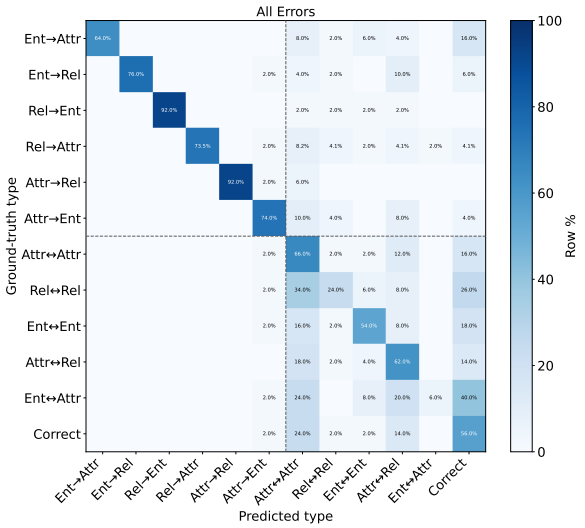

Figure 3: Confusion matrices for the full-taxonomy setting. It requires distinguishing among all structural violations simultaneously, while the paired setting provides the reference diagram and asks the model to identify the injected change. Despite access to the reference, models still exhibit substantial confusion among connection-based errors, suggesting failures in edge-level visual grounding rather than schema ambiguity alone.

Difficulty	Binary	Shape	Conn.	Full	Paired
Easy	0.861	0.877	0.459	0.700	0.659
Medium	0.757	0.747	0.258	0.489	0.472
Hard	0.737	0.685	0.278	0.469	0.482

Table 3: Macro-F1 by difficulty tier (commercial VLM average). Connection recognition degrades non-monotonically, while shape recognition declines steadily with increasing structural complexity.

Difficulty scaling. As shown in Table 3, recognition degrades with diagram complexity: full-taxonomy Macro-F1 drops from 0.70 on easy diagrams to 0.49 on medium and 0.47 on hard. Shape recognition declines steadily (0.88 $\rightarrow$ 0.75 $\rightarrow$ 0.69), while connection recognition drops sharply from easy to medium (0.46 $\rightarrow$ 0.26) before a partial recovery on hard diagrams (0.28). This recovery does not indicate improved topology grounding; rather, the medium tier contains denser edge crossings and more visually ambiguous layouts, making endpoint verification substantially harder. The primary bottleneck is therefore not diagram size itself, but reliable topology extraction under structurally ambiguous conditions.

Dual-error recognition. Real-world ERDs frequently contain multiple simultaneous inconsistencies. Hence, we additionally evaluate diagrams containing two simultaneous structural violations. Performance drops substantially across all mod-

Condition	Full	Shape	Conn.
Original	0.700	0.877	0.459
Random label	0.578	0.718	0.405
Label permuted	0.509	0.631	0.349

Table 4: Effect of label perturbations on easy-tier Macro-F1 (commercial VLM average). Both conditions preserve visual structure; performance drops confirm reliance on semantic compatibility priors.

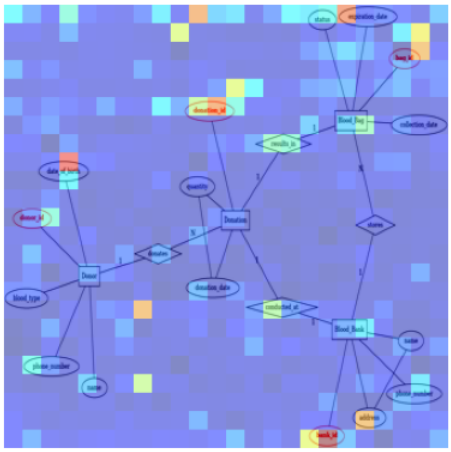

Figure 4: The model's attention map shows a focus on semantically relevant text elements rather than visual connections, failing to trace the connecting lines that define the error.

els: averaged across tiers, the three commercial VLMs drop from a mean Macro-F1 of 0.595 on single-error diagrams to 0.525 in the dual-error setting. Models often identify that a diagram is invalid but recover only one of the two violations, or confuse the interaction between them. This suggests VLMs lack a stable global representation of diagram topology and instead rely on local cues that degrade under multiple interacting violations.

Generated versus real-world-context ERDs. We compare fully synthetic ERDs with ERDs whose schemas are derived from real-world sources, while rendering both sets using the same Graphviz pipeline after error injection. Thus, this comparison controls for visual style and primarily tests the effect of schema context and structural regularities rather than rendering noise. Because the real-world-context diagrams are closest in size to the easy synthetic tier, we compare them against easy diagrams. Connection-related recognition improves from 0.405 Macro-F1 on easy synthetic ERDs to 0.615 on real-world-context ERDs averaged across commercial VLMs, while shape-related recognition decreases from 0.920 to 0.724. This suggests that realistic schema context can help models infer plausible connections, but does not necessarily improve grounded geometric recognition. The result reinforces our broader finding: VLMs benefit from semantic regularities, yet still struggle to ground ERD structure visually.

4.4 Diagnostic Studies

To isolate the role of semantic context, we construct controlled variants that modify textual labels while preserving visual structure and injected violations. Performance differences are thus attributable directly to semantic perturbations. OCR is not a confound: average transcription accuracy is 95.5% across models on held-out diagrams.

Random-label ERDs. Replacing all labels with random three-character strings drops mean Macro-F1 from 0.7 to 0.58. The effect is not uniform: some connection errors improve slightly, suggesting meaningful labels can interfere with visual reasoning by inducing hallucinated relationships. Shape errors degrade more sharply (S2 $\approx$16%, S3 $\approx$17%), indicating that element recognition relies on semantic context rather than geometry.

Label-permuted ERDs. Reassigning original labels to different elements while preserving graph structure produces the largest degradation overall (0.700 $\rightarrow$ 0.509). Connection errors are most affected, confirming that models rely heavily on semantic compatibility priors and cannot recover element identity or connectivity from visual topology alone when labels conflict with structure.

Few-shot prompting. Few-shot demonstrations produce the largest gains of any intervention, increasing mean Macro-F1 from 0.686 to 0.848 across commercial VLMs on easy diagrams. Connection errors benefit most: averaged across commercial VLMs, C4 improves from 0.509 to 0.801, C5 from 0.153 to 0.822, and C1 from 0.504 to 0.782. Shape errors also improve, though already-high categories show smaller gains. The underlying asymmetry does not fully disappear: models benefit most when demonstrations explicitly expose the visual structure of the error, suggesting that few-shot prompting mainly aligns attention toward relevant visual patterns rather than inducing stable topological reasoning.

4.5 Consistency Across Model Generations

We additionally compare successive releases within major VLM families to examine whether the observed failures reflect transient version effects or more persistent limitations. Although newer gen-

erations consistently improve over their predecessors, the gains remain relatively modest. More importantly, the same qualitative failure patterns persist across releases: models perform substantially better on shape-based violations than on connection-based violations, and all families continue to struggle with topology-sensitive errors requiring edge tracing and endpoint verification. This suggests that improvements in scale and general multimodal capability do not readily translate into robust grounded structural verification.

4.6 Qualitative Attention Analysis.

In ERDs, identifying a connection error requires tracing edges between elements rather than relying solely on their semantic labels. To qualitatively examine the grounding behavior underlying connection-error failures, we visualize the spatial attention patterns of Qwen-3.5 during prediction. We emphasize that these visualizations are illustrative rather than causal explanations of model behavior. Across connection-error examples, we consistently observe the same qualitative pattern: attention concentrates primarily on semantically salient text regions while assigning comparatively less emphasis to edge segments and endpoints. Figure 4 shows a representative example.

For each generated output token, we extract attention weights over all visual tokens from the final transformer layer and aggregate them into a single spatial heatmap, which is then bilinearly upsampled to the original image resolution. The resulting visualization shows substantially higher activation around textual node labels than along the edge responsible for the structural violation. In the example shown in Figure 4, the model incorrectly predicts a connection between "status" and "ag_id," whereas the actual Error Type 1 violation involves the edge connecting "name" and "address" (green arrow). The observed attention distribution is consistent with the hypothesis that the model prioritizes semantically plausible label associations over explicit edge verification during prediction.

This qualitative behavior aligns closely with the quantitative findings from the label-permuted experiments in Table 4, where reassigning labels to incorrect elements produced the largest performance degradation across all conditions. Together, these results suggest that current VLMs rely heavily on semantic compatibility priors while struggling to robustly ground topological relations through explicit edge tracing and endpoint verification.

Overall, these controlled studies point to the same limitation. Across random-label, label-permuted, few-shot, and dual-error settings, the largest failures consistently emerge in connection-related violations requiring reliable edge tracing and endpoint grounding. Semantic context, prompting, and reference diagrams can partially shift model behavior, but none resolve the core difficulty of reliably tracing and grounding topology in complex diagram structures.

5 Conclusion

We formulate ERD understanding as a controlled structural error-classification task that evaluates visually grounded reasoning rather than question answering or graph reconstruction. To study this problem, we introduce ERClass benchmark: an ERD error-classification dataset with controlled structural violations.

Across recent commercial and open-weight VLMs, we observe a consistent gap between shape recognition and connection reasoning. Models perform substantially better on shape-based violations, while a symbolic computer-vision baseline outperforms most of the evaluated VLMs on connection recognition. Random-label, label-permuted, paired, and few-shot analyses further show that models often rely on semantic plausibility rather than grounded topology perception. Together, these findings reveal three recurring failure modes in current VLMs: unreliable edge tracing, weak geometric discrimination, and semantic hallucination.

Our results suggest that progress in diagram understanding requires not only stronger multimodal models, but also evaluation paradigms that isolate visual grounding from semantic reasoning.

6 Limitations

Our benchmarks focus primarily on ERDs, and additional work is needed to determine whether the same failure modes generalize to broader classes of technical diagrams. Moreover, our evaluations are restricted to prompt-based inference without external symbolic tools or neuro-symbolic post-processing, so the results should not be interpreted as upper bounds on hybrid systems. Finally, although our experiments provide strong evidence for failures in edge tracing and topology perception, they do not fully explain the underlying mechanisms responsible for these limitations in transformer-based VLMs.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Google AI. 2025. Gemini language model. Accessed July 31, 2025.

Iñigo Alonso, Gorka Azkune, Ander Salaberria, Jeremy Barnes, and Oier Lopez de Lacalle. 2025. Vision-language models struggle to align entities across modalities. *arXiv preprint arXiv:2503.03854*.

Anthropic. 2025. Claude 4.0 sonnet. Large language model.

Abdul Arbaz, Heng Fan, Junhua Ding, Meikang Qiu, and Yunhe Feng. 2024. Genflowchart: parsing and understanding flowchart using generative ai. In *International Conference on Knowledge Science, Engineering and Management*, pages 99–111. Springer.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Jon Barwise and John Etchemendy. 1995. Heterogeneous logic. *Diagrammatic reasoning: Cognitive and computational perspectives*, pages 211–234.

Gary Bradski. 2000. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123.

James Burgess, Jeffrey J Nirschl, Laura Bravo-Sánchez, Alejandro Lozano, Sanket Rajan Gupte, Jesus G Galaz-Montoya, Yuhui Zhang, Yuchang Su, Disha Bhowmik, Zachary Coman, and 1 others. 2025. Microvqa: A multimodal reasoning benchmark for microscopy-based scientific research. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19552–19564.

Daniele De Bari, Giacomo Garaccione, Riccardo Coppola, Marco Torchiano, and Luca Ardito. 2024. Evaluating large language models in exercises of uml class diagram modeling. In *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 393–399.

Graphviz. 2024. Graphviz: Open Source Graph Visualization Software Documentation. https://graphviz.org/doc/. Accessed: May 26, 2026.

Yifan Hou, Buse Giledereli, Yilei Tu, and Mrinmaya Sachan. 2024. Do vision-language models really understand visual language? *arXiv preprint arXiv:2410.00193*.

Anwen Hu, Yaya Shi, Haiyang Xu, Jiabo Ye, Qinghao Ye, Ming Yan, Chenliang Li, Qi Qian, Ji Zhang, and Fei Huang. 2024. mplug-paperowl: Scientific diagram analysis with the multimodal large language model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6929–6938.

Mateja Jamnik. 2001. *Mathematical Reasoning with Diagrams: From Intuition to Automation*. CSLI Publications.

Ryo Kamoi, Yusen Zhang, Sarkar Snigdha Sarathi Das, Ranran Haoran Zhang, and Rui Zhang. 2024. Visonlyqa: Large vision language models still struggle with visual perception of geometric information. *arXiv preprint arXiv:2412.00947*.

Jill H Larkin and Herbert A Simon. 1987. Why a diagram is (sometimes) worth ten thousand words. *Cognitive science*, 11(1):65–100.

Ruizhou Li and Haiyun Jiang. 2025. Graph-to-vision: Multi-graph understanding and reasoning using vision-language models. *arXiv preprint arXiv:2503.21435*.

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

Marina Mancoridis, Bec Weeks, Keyon Vafa, and Sendhil Mullainathan. 2025. Potemkin understanding in large language models. *arXiv preprint arXiv:2506.21521*.

Rahul Mehta, Bhavyajeet Singh, Vasudeva Varma, and Manish Gupta. 2024. Circuitvqa: A visual question answering dataset for electrical circuit images. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 440–460. Springer.

Aniket Munde. 2025. Evaluation of vision language model on uml diagrams.

Jio Oh, Soyeon Kim, Junseok Seo, Jindong Wang, Ruochen Xu, Xing Xie, and Steven Whang. 2024. Erbench: An entity-relationship based automatically verifiable hallucination benchmark for large language models. *Advances in Neural Information Processing Systems*, 37:53064–53101.

Huitong Pan, Qi Zhang, Cornelia Caragea, Eduard Dragut, and Longin Jan Latecki. 2024. Flowlearn: Evaluating large vision-language models on flowchart understanding. In *ECAI 2024*, pages 73–80. IOS Press.

Nenad Petrovic, Yurui Zhang, Moaad Maaroufi, Kuo-Yi Chao, Lukasz Mazur, Fengjunjie Pan, Vahid Zolfaghari, and Alois Knoll. 2025. Multi-modal summarization in model-based engineering: Automotive software development case study.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. 2024. Vision language models are blind. In *Proceedings of the Asian Conference on Computer Vision*, pages 18–34.

Rami Rashkovits and Ilana Lavy. 2021. Mapping common errors in entity relationship diagram design of novice designers. *International Journal of Database Management Systems*.

William Rudman, Michal Golovanevsky, Amir Bar, Vedant Palit, Yann LeCun, Carsten Eickhoff, and Ritambhara Singh. 2025. Forgotten polygons: Multimodal large language models are shape-blind. *arXiv preprint arXiv:2502.15969*.

Atsushi Shimojima. 2015. Semantic properties of diagrams and their cognitive potentials. *Studies in Logic, Language and Information*.

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 22496–22519.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Tongyi Lab. 2023. Qwen: A large-scale language model developed by alibaba cloud. Accessed July 31, 2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Junyi Ye, Ankan Dash, Wenpeng Yin, and Guiling Wang. 2024. Beyond end-to-end vlms: Leveraging intermediate text representations for superior flowchart understanding. *arXiv preprint arXiv:2412.16420*.

Enming Zhang, Ruobing Yao, Huanyong Liu, Junhui Yu, and Jiale Wang. 2024a. First multi-dimensional evaluation of flowchart comprehension for multimodal large language models. *arXiv preprint arXiv:2406.10057*.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, and 1 others. 2024b. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer.

A Additional Details for Error-new

This appendix documents the prompts, model breakdowns, CV-baseline implementation, representative examples, and Graphviz rendering settings used for the ERROR-NEW experiments. All paths below are relative to the repository root.

A.1 Error Taxonomy

A.2 Schema Generation Prompt and Real-World Sources

Synthetic schemas were generated by prompting GPT with scenario-grounded instructions that control diagram complexity per tier. The generation prompt specifies the target number of entities for each difficulty level, producing JSON with entities, attributes, primary keys, relationships, participating entities, and cardinalities. A representative generation prompt is:

> You are a database designer. Generate an ER schema for the everyday scenario "<SCENARIO>". The schema must have exactly <N_ENTITIES> entities, each with 3–5 attributes including a primary key. Create <N_RELATIONSHIPS> relationships connecting the entities with appropriate cardinalities (1:1, 1:N, or M:N). Include at least one weak entity with an identifying relationship. Return JSON only with keys: entities (each with attributes, primary_keys, weak), relationships (each with entities, cardinality, name, weak). Use clear, domain-relevant labels and avoid duplicate entity or relationship names.

The difficulty tiers were controlled by constraining the entity and relationship counts: easy schemas target 4 entities and 3–4 relationships; medium schemas target 5 entities and 4–5 relationships with additional features (derived, composite, and multivalued attributes); hard schemas target 7–8 entities and 7–8 relationships with all extended ER features (IsA hierarchies, derived/composite/multivalued attributes, and n-ary relationships). The target was 200 everyday scenarios per tier, later filtered to 199 usable schemas per difficulty level.

The real-world set contains 148 schemas: 48 from textbooks and 100 from internet sources (university tutorials, practitioner examples, and public educational websites). Diagrams were cleaned by converting each source to the project JSON schema, normalizing labels, removing duplicate/unreadable examples, and rendering with the same Graphviz pipeline before error injection.

A.3 Correct Dataset Statistics

Table 6 summarizes the structural complexity of correct (non-errored) schemas across all splits. Nodes include entities, relationships, and attributes; edges include attribute-to-entity connections and relationship-to-entity participation links.

The synthetic tiers scale approximately linearly in complexity: hard schemas have nearly twice the nodes and edges of easy schemas. The real-world sources span a comparable range: books schemas are closest to the easy tier in total node count (18.46 vs. 22.03) but have fewer attributes per entity since textbook examples often omit attribute lists for brevity. Internet schemas are closest to the medium tier (29.08 vs. 27.83 nodes) and have attribute density comparable to the synthetic sets. The hard tier exceeds both real-world sources in average complexity, ensuring that the evaluation probes model behavior on denser diagrams than those typically encountered in educational materials.

Across all 745 schemas (597 synthetic + 148 real-world), the dataset contains a total of 4,117 entities, 3,761 relationships, and 19,851 attributes before error injection. Each schema produces 11 errored variants (one per error code S1–S6, C1–C5) plus one dual-error variant, yielding over 8,000 errored diagrams for the full evaluation.

A.4 Vision-Language Model Prompts

All VLM calls used temperature 0.0 and required strict JSON output. The prompts below are the prompt templates used by `ErrorNew/Codes/erd_eval_common.py`; long lines are wrapped for the two-column ACL layout, but the instruction content and output schemas are preserved.

Shared JSON-Only Prefix.

> STRICT OUTPUT – violations will break automated scoring: Reply with exactly ONE JSON object and nothing else (no prose, no markdown, no code fences). Do not add keys beyond those listed in the template (no explanation, no reasoning). Start your reply with { and end with }.

Taxonomy Text Included in Prompts.

> Shape errors occur when an element uses the wrong shape: S1 – Entity drawn as an ellipse (should be rectangle); S2 – Entity drawn as a diamond (should be rectangle); S3 – Relationship drawn as a rectangle (should be diamond); S4 – Relationship drawn as an ellipse (should be diamond); S5 – Attribute drawn as a diamond (should be ellipse); S6 – Attribute drawn as a rectangle (should be ellipse).

Table 5: Error taxonomy used in ERROR-NEW. Shape errors replace an ER element with the wrong visual shape. Connection errors connect ER element types that should not be directly connected.

Code	Name	Operational definition
S1	Entity as Attribute Shape	Entity drawn as an ellipse instead of a rectangle
S2	Entity as Relationship Shape	Entity drawn as a diamond instead of a rectangle
S3	Relationship as Entity Shape	Relationship drawn as a rectangle instead of a diamond
S4	Relationship as Attribute Shape	Relationship drawn as an ellipse instead of a diamond
S5	Attribute as Relationship Shape	Attribute drawn as a diamond instead of an ellipse
S6	Attribute as Entity Shape	Attribute drawn as a rectangle instead of an ellipse
C1	Attribute-to-Attribute Connection	Attribute connected directly to another attribute
C2	Relationship-to-Relationship Connection	Relationship connected directly to another relationship
C3	Entity-to-Entity Connection	Entity connected directly to another entity
C4	Attribute-to-Relationship Connection	Attribute connected directly to a relationship
C5	Entity-to-Attribute Connection	Entity connected directly to an attribute

Table 6: Dataset statistics by split. Values are averages per schema. Nodes = entities + relationships + attributes. Edges = attribute-to-entity links + relationship participation links.

Split	Schemas	Avg Entities	Avg Rels	Avg Attrs	Avg Nodes	Avg Edges
Easy	199	4.04	3.89	14.10	22.03	22.62
Medium	199	5.09	4.79	17.95	27.83	28.28
Hard	199	7.59	7.59	27.01	42.19	42.58
Books	48	5.15	4.96	8.35	18.46	15.46
Internet	100	6.16	5.48	17.44	29.08	27.98

Connection errors occur when elements are connected incorrectly: C1 – Attribute connected directly to another attribute; C2 – Relationship connected directly to another relationship; C3 – Entity connected directly to another entity (no relationship diamond); C4 – Attribute connected directly to a relationship; C5 – Entity connected directly to an attribute.

Study 1: Binary Scoped Error Detection.

TASK – binary check for exactly one error type: `<CODE>`. You receive one ER diagram image. It is an erroneous variant produced to test only this rule; ignore every other error category, since they do not apply here. The prompt then includes the single taxonomy line for `<CODE>` only. Question: does error `<CODE>` appear in this diagram? Emit JSON only – three keys, no prose, no markdown. If `<CODE>` is present, `has_error` is true, `TYPE` must be exactly the string `<CODE>` with no other code, and `Elements` is the comma-separated graph element name(s) involved in that error only. If `<CODE>` is absent, `has_error` is false, `TYPE` is null, and `Elements` is null. Return only: `{"has_error": true/false, "TYPE": "<CODE or null>", "Elements": "<comma-separated element id(s) or null>"}`.

Study 2: Shape-Only Detection.

You are an expert in Entity-Relationship (ER) diagrams. This task is shape errors only. Do not report connection issues (C1–C5); use only the shape-error categories S1–S6 listed above. Examine the ER diagram. If no shape error, set `has_shape_error` false and `error_type` / `error_element` to null. Return ONLY this JSON shape, three keys and nothing else: `{"has_shape_error": true/false, "error_type": "<S1-S6 or null>", "error_element": "<affected element id or null>"}`.

Study 3: Connection-Only Detection.

You are an expert in Entity-Relationship (ER) diagrams. This task is connection errors only. Do not report wrong shapes (S1–S6); use only the connection-error categories C1–C5 listed above. Examine the ER diagram. If no connection error, set `has_connection_error` false and `error_type` / `error_elements` to null. Return ONLY this JSON shape, three keys and nothing else: `{"has_connection_error": true/false, "error_type": "<C1-C5 or null>", "error_elements": "<comma-separated involved element ids or null>"}`.

Study 4: Full-Taxonomy Detection.

You are an expert in Entity-Relationship (ER) diagrams. The prompt includes the full S1–S6 and C1–C5 taxonomy listed above. Examine the ER diagram. If no error, set `has_error` false and `TYPE` / `Elements` null. Return ONLY this JSON shape, three keys and nothing else: `{"has_error": true/false, "TYPE":`

"<S1-S6 or C1-C5 or null>", "Elements": "<comma-separated node id(s) involved, or null>"}.

Study 5: Pairwise Detection.

You are an expert in Entity-Relationship (ER) diagrams. The prompt includes the full S1–S6 and C1–C5 taxonomy listed above. You are given two ER diagrams for the same underlying schema, labelled IMAGE_A and IMAGE_B. One image is the correct reference; the other has exactly one structural error from the categories above. Return ONLY this JSON shape, three keys and nothing else: {"error_in": "A", "error_type": "<S1-S6 or C1-C5>", "error_element": "<affected element id or null>"}. Use error_in exactly "A" or "B", the image that contains the error.

A.5 Graphviz Generation Setting

Correct ER diagrams and errored variants were rendered with Graphviz. The generator uses Digraph(format="png", engine="dot") and sets the graph layout to sfdp. The relevant rendering parameters are:

Rendering configuration: format=png, engine=dot, layout=sfdp; graph attributes overlap=prism, splines=spline, sep=+6, pad=0.3, K=0.7, start=random, and outputorder=edgesfirst. Node attributes use rectangular node shapes, style=rounded, fontsize=13, width=1.2, height=0.5, margin=0.25,0.18, and penwidth=1.6. Edge attributes are penwidth=1.5, arrowsize=0.7, and len=1.3.

A.6 CV Baseline

The CV baseline is the rule-based erd_analyzer_shape_based_v2 pipeline. It detects visual shapes and connecting edges, optionally uses OCR for element names, and maps invalid shape pairs to connection errors. Shape-error inference is enabled so that an invalid local connection can also imply a likely wrong-shape repair.

Implementation. The analyzer pads each diagram with a white border, thresholds the grayscale image, applies Canny edges, and closes contours with morphology before detecting candidate ERD nodes. Candidate contours are filtered by area and aspect ratio, then classified as rectangles, diamonds, or ovals using geometry-first rules: diamond geometry is checked first, oval geometry second, and rectangle geometry only after sharp-corner and area-fill tests. CLIP can be used as a fallback; the hard run used CLIP enabled, OCR element pairing enabled, shape-error inference enabled, 25 images per class, and included 25 correct Graphviz-hard diagrams. Detected shapes are mapped to ER roles by rectangle → entity, diamond → relationship, and oval → attribute.

Connections are detected in a separate edge mask. The implementation removes detected node regions from the line mask, closes remaining line fragments, finds line contours, uses the farthest pair of points as an approximate connection segment, and attaches each endpoint to the nearest detected node within a fixed radius. The only valid direct role pair is entity–relationship. Direct attribute–attribute, relationship–relationship, entity–entity, attribute–relationship, and entity–attribute pairs are mapped to C1–C5. When shape-error inference is enabled, the same invalid local pair is also used as a repair heuristic: if changing one endpoint's intended role would make the pair entity–relationship, the endpoint's current visual shape is mapped to the corresponding S1–S6 code. OCR is used only for the Elements field: EasyOCR text blocks are assigned to detected nodes by spatial containment or nearest-node distance, and the resulting labels are exported with each predicted error code.

Across all three splits, the CV baseline achieves a binary accuracy of 0.917 (easy: 1200 samples, med: 300, hard: 300). Type accuracy is 0.503 on easy, 0.640 on medium, and 0.673 on hard; element accuracy is 0.426, 0.564, and 0.535 respectively. On the hard split per-class breakdown (25 examples each), type accuracy is 1.00 for S1, S4, C1, C4, and C5; 0.96 for S3; 0.80 for S2 and C3; 0.52 for C2; and 0.00 for S5, S6, and correct diagrams. Binary accuracy is 1.00 for all error classes and 0.00 for correct diagrams.

The CV baseline results should be interpreted as a topology-oriented sanity check rather than a complete ERD understanding system. Its binary accuracy is inflated by detecting errors in all erroneous diagrams, while its false-positive behavior on correct diagrams shows poor calibration. Nevertheless, its type and element scores are meaningful because they measure how much can be recovered from explicit shape and edge extraction alone. The fact that this simple symbolic system competes with or exceeds VLMs on several connection categories supports the paper's conclusion that explicit topology extraction remains important for ERD evaluation.

A.7 Per-Model Error-Type Breakdown

Table 7 reports the per-model, per-error-type total F1 on the easy semantic setting. Each cell is the mean of per-class total F1 over the studies in which that error code is evaluated; success requires both the correct error type and an overlapping `Elements` prediction. The pattern makes the main failure mode explicit: shape errors are generally easier than connection errors, especially C1–C5. The table should therefore be read not only as a leaderboard, but as an error-diagnostic matrix: high scores on S1–S6 show that most frontier VLMs can identify the coarse ERD primitive once the object is localized, whereas the sharp drop on C1–C5 shows that the same models are much less reliable at tracing edges and verifying endpoint compatibility. The contrast with the CV baseline is also informative: the baseline is weak on some shape substitutions, but it is competitive or stronger on several connection categories because it explicitly reconstructs edges rather than relying on semantic plausibility.

Table 7 also reveals several model-family biases. GPT and Gemini are the most stable across the full taxonomy, while Claude is strong on shape substitutions but degrades sharply on direct-connection violations. The open-weight models show the largest gap between shape and connection recognition, suggesting that their failures are not caused by lack of ERD terminology alone. The CV baseline has the opposite profile: it does not understand labels semantically, but its explicit edge extraction gives it a strong advantage on C3 and moderate gains on C5.

A.8 Few-Shot Prompting Details

The few-shot condition uses the same output schema and error taxonomy as the zero-shot condition, but prepends labelled examples before the query image. The legacy one-shot implementation in `ErrorOld/Error/Codes/error_few_api.py` selected one held-out example per error type from the non-evaluation portion of each error folder; the example consisted of the diagram image and its ground-truth JSON. The prompt then asked the model to analyze the target diagram and return only the required JSON. For ERROR-NEW, the few-shot setting is stored as `easy-fewshot`; the comparison CSV is `ErrorNew/EvalResults/reports/comparisons/02_zeroshot_vs_fewshot_by_error_type.csv`.

The few-shot results show that demonstrations help every model family, but the largest relative gains occur for weaker open-weight models. This indicates that part of the zero-shot error comes from output calibration and task framing. However, few-shot prompting does not remove the connection-recognition bottleneck: even after demonstrations, the hardest cases remain those requiring precise endpoint disambiguation.

Few-shot template: labelled ERD example(s) with their ground-truth JSON are shown before the query image, followed by the instruction "Now analyze the following ER diagram." The required output remains exactly the same JSON object as the corresponding zero-shot study. Examples are selected from held-out examples in the same dataset family, excluding the evaluated image set.

A.9 Declarative Knowledge Probe

Before image evaluation, the models were also given a text-only declarative probe to check whether they could state the taxonomy without visual reasoning. The probe did not include images and was not scored with the diagram-evaluation metrics. It asked models to define S1–S6 and C1–C5 in plain text; all evaluated models returned definitions matching the taxonomy in Table 5.

Text-only knowledge probe question: "Define each ER-diagram structural error code S1, S2, S3, S4, S5, S6, C1, C2, C3, C4, and C5. For each code, state the wrong shape or wrong direct connection and the correct ER convention it violates." A sample expected response is: S1 is an entity drawn as an attribute ellipse instead of an entity rectangle; C3 is a direct entity-to-entity connection without an intervening relationship diamond.

A.10 OCR and Element-Localization Measurement

The reported element score is not a standalone OCR word-error rate. It is measured after the model predicts the correct error type: the predicted `Elements` string is normalized, tokenized, and compared with the metadata `Elements` field using token or substring overlap. Thus, this metric captures both text transcription and localization of the element involved in the error. Table 10 gives the aggregate conditional element-overlap rates by split and model.

Table 10 separates type recognition from element localization. GPT has the highest conditional element-overlap rate, but the decline from Easy to

Table 7: Per-model by error-code total F1 on easy semantic ERROR-NEW. Values are averaged across applicable studies.

Code	Claude	Gemini	GPT	Gemma	Llama	Qwen	CV
S1	0.90	0.89	0.90	0.19	0.06	0.78	0.65
S2	0.95	0.88	0.97	0.21	0.22	0.77	0.95
S3	0.97	0.96	0.97	0.44	0.16	0.83	0.94
S4	0.98	0.95	0.96	0.10	0.15	0.57	0.65
S5	0.96	0.87	0.96	0.08	0.12	0.88	0.00
S6	0.73	0.87	0.90	0.09	0.00	0.43	0.00
C1	0.23	0.63	0.65	0.02	0.06	0.18	0.18
C2	0.18	0.40	0.54	0.04	0.17	0.37	0.10
C3	0.21	0.48	0.76	0.15	0.09	0.40	0.93
C4	0.44	0.53	0.56	0.00	0.05	0.22	0.18
C5	0.15	0.21	0.11	0.14	0.01	0.26	0.44

Table 8: Difficulty scaling by model. Values are mean total F1 over the 11 error codes, averaged across applicable studies for each tier.

Tier	Claude	Gemini	GPT	Gemma	Llama	Qwen	CV
Easy	0.608	0.698	0.753	0.134	0.099	0.517	0.456
Med	0.445	0.486	0.620	0.195	0.080	0.320	0.576
Hard	0.486	0.518	0.631	0.240	0.168	0.374	0.577

Med shows that naming the correct involved element becomes harder as diagrams become denser. This supports the main paper's claim that failures are not merely label-recognition errors: even when the model selects the right error code, it may localize the wrong endpoint or hallucinate a nearby semantically plausible element.

The CV baseline detected an error in every erroneous image in all three reported runs (zero false negatives), but it also flagged every correct image as erroneous (100/100 false positives for easy, 25/25 for medium, and 25/25 for hard). This makes it a useful sanity-check baseline for localization and shape/edge extraction, but not a calibrated no-error detector.

A.11 Representative Errored Diagrams

This section presents representative examples of each error category alongside their ground-truth metadata and a correct reference diagram.

Shape Error Example (S4). Figure 6 shows the Arrest Procedure schema with a shape error. The relationship "Performs" is drawn as an ellipse (attribute shape) instead of the correct diamond shape, corresponding to error code S4. The ground-truth metadata is: TYPE = S4, Elements = Performs.

Connection Error Example (C4). Figure 7 shows the AI Chatbot Conversation schema with a connection error. The attribute "User_user_id" is connected directly to the relationship "covers", violating ER conventions (attributes should attach to entities, not relationships). The ground-truth metadata is: TYPE = C4, Elements = User_user_id, covers.

Dual Error Example. Figure 8 shows the Beach Vacation schema with two simultaneous errors: (1) the entity "Accommodation" is drawn as a diamond (relationship shape), corresponding to S2; and (2) the relationships "books" and "offers" are connected directly to each other, corresponding to C2. The ground-truth metadata is: TYPE = S2, Elements = Accommodation; TYPE = C2, Elements = books, offers.

Correct Reference Diagram. Figure 9 shows the same Beach Vacation schema rendered correctly for comparison. All entities use rectangles, relationships use diamonds, attributes use ellipses, and connections follow valid ER conventions.

Table 9: Zero-shot vs. few-shot total F1, averaged across the 11 error types on easy semantic diagrams.

Model	Zero-shot	Few-shot	Relative gain
Claude	0.608	0.787	29.4%
Gemini	0.698	0.865	24.0%
GPT	0.753	0.892	18.5%
Gemma	0.134	0.235	75.0%
Llama	0.099	0.221	123.4%
Qwen	0.517	0.717	38.5%
CV	0.456	0.664	45.7%

Table 10: Mean total F1 pooled across studies, requiring both correct error type and correct element localization. Higher values indicate stronger joint type+element performance.

Split	Claude	Gemini	GPT	Gemma	Llama	Qwen	CV
Easy	0.631	0.734	0.769	0.140	0.093	0.553	0.542
Med	0.472	0.522	0.640	0.195	0.074	0.356	0.662
Hard	0.427	0.478	0.616	0.218	0.158	0.334	0.561

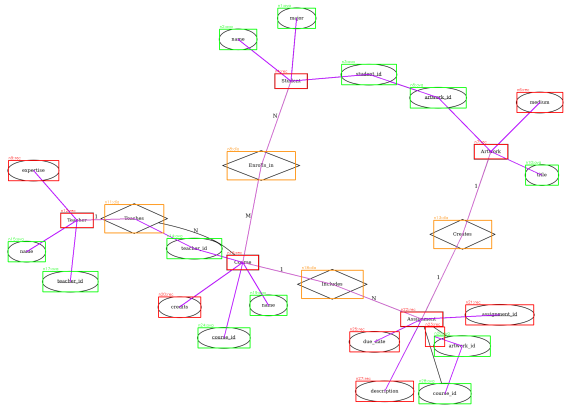

Figure 5: CV-baseline debug visualization on the Art Class schema. Detected shapes are highlighted with colored bounding boxes: green for ovals (attributes), red for rectangles (entities), and orange for diamonds (relationships). Purple lines indicate detected edge connections. The overlay shows how the rule-based pipeline classifies node geometry and traces inter-node edges before mapping shape–role violations to error codes.

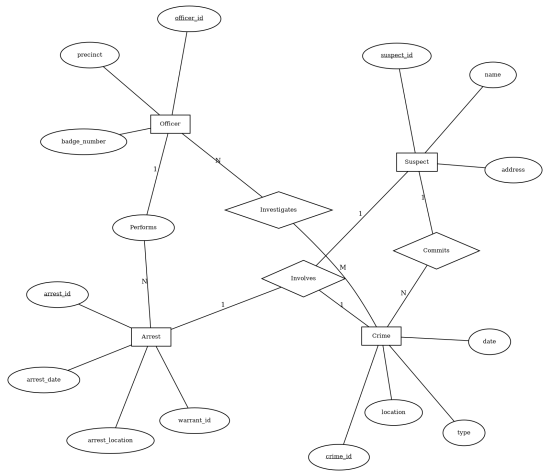

Figure 6: Shape error S4 in the Arrest Procedure schema. The relationship "Performs" (linking Officer to Arrest) is incorrectly rendered as an ellipse instead of a diamond, making it visually indistinguishable from an attribute node. Models must recognize that a node labelled with a verb and connected between two entities should use diamond geometry.

A.12 Schema Generation Prompt and Real-World Sources

Synthetic schemas were generated in a first stage by prompting GPT to produce scenario-grounded ER specifications: entities, attributes, primary keys, relationships, relationship endpoints, and cardinalities. The target was 200 everyday scenarios per synthetic tier, later filtered to 199 usable schemas for easy/medium/hard. A reviewer-facing generation prompt should be archived with the released data; if the exact prompt is omitted from the main paper, include it here as follows.

Schema-generation prompt placeholder: "Generate an ER schema for the scenario <SCENARIO>. Return JSON only with entities, attributes, primary keys, relationship names, participating entities, and cardinalities. Use clear, domain-relevant labels and avoid duplicate entity or relationship names."

The real-world set contains 148 schemas: 48

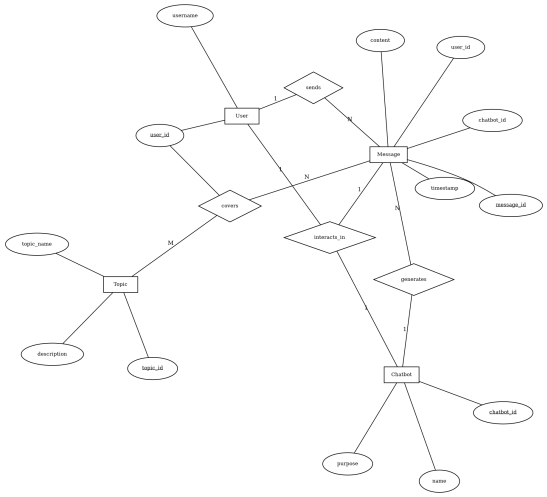

Figure 7: Connection error C4 in the AI Chatbot Conversation schema. The attribute "User_user_id" is connected directly to the relationship "covers" instead of being attached to its parent entity "User". This error type requires models to trace edge endpoints and verify that attribute nodes connect only to entity nodes.

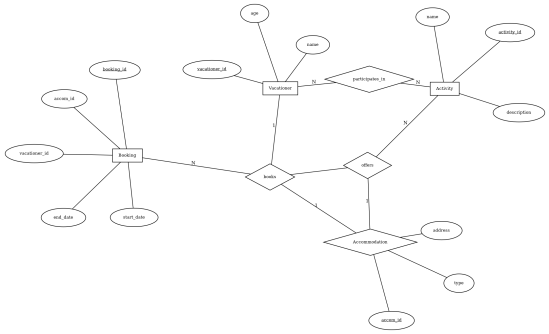

Figure 8: Dual-error diagram for the Beach Vacation schema. Two errors co-occur: the entity "Accommodation" is rendered as a diamond instead of a rectangle (S2), and the relationships "books" and "offers" are directly connected to each other without an intervening entity (C2). Dual-error cases test whether models can identify multiple violations in a single diagram.

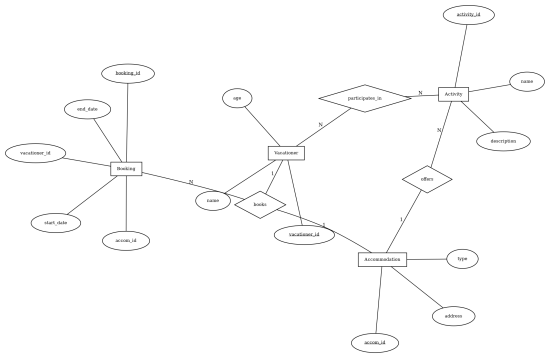

Figure 9: Correct Graphviz rendering of the Beach Vacation schema. Entities (Vacationer, Booking, Activity, Accommodation) are rectangles, relationships (participates_in, books, offers) are diamonds, and attributes are ellipses. Comparing with Figure 8 highlights the injected shape and connection violations.

from books and 100 from internet sources. Collection sources were textbooks, university assignment/tutorial material, practitioner examples, and public tutorial websites. Diagrams were cleaned by converting each source to the project JSON schema, normalizing labels, removing duplicate/unreadable examples, and rendering with the same Graphviz pipeline before error injection.

A.13 Error-Injection Example Pair

Figures 9 and 8 serve as a before/after injection pair using the Beach Vacation schema. The correct rendering (Figure 9) follows all ER conventions, while the errored variant (Figure 8) contains two injected violations (S2 and C2). Comparing the two makes the taxonomy concrete: the entity "Accommodation" changes from a rectangle to a diamond, and the relationship pair "books"–"offers" gains an invalid direct edge.

A.14 Additional Validation and Failure Modes

Manual validation checked that generated schema specifications conformed to ER conventions before rendering and error injection. For a camera-ready version, report the number of manually reviewed schemas, the annotator count, and an agreement statistic if multiple annotators were used. We also recommend including one model failure beyond the attention-map example: for example, a correct diagram where a model hallucinates a relationship-shape error, or a connection-error diagram where the model identifies the right error family but names the wrong endpoint.

A.15 Reproducibility and Reviewer-Facing Notes

The main scripts used to produce and inspect ERROR-NEW are:

- `ErrorNew/Codes/error.py`: Graphviz rendering and error injection.
- `ErrorNew/Codes/generate_label_variant_datasets.py`: context-free and label-permuted variants.
- `ErrorNew/Codes/erd_eval_common.py`:

model registry, prompt builders, image loading, and API clients.

- `ErrorNew/CV-Baseline/run_shape_analyzer_dataset.py`: CV-baseline batch runner.
- `ErrorNew/CV-Baseline/export_cv_to_eval_results.py`: conversion of CV output to VLM-compatible study CSVs.

For VLM evaluation, OpenRouter calls used a chat-completions endpoint with `temperature=0.0`; Ollama calls used `/api/chat` with `temperature=0.0`, `think=false`, and JSON-only prompts. The evaluation code can optionally resize images before API calls through `ERD_EVAL_IMAGE_MAX_SIDE`, and can limit per-class samples through `ERD_EVAL_FRACTION_PER_CLASS` or `max_per_class`.

The ground-truth metadata uses the keys `TYPE` and `Elements`. For single-error diagrams, `TYPE` is one of `S1-S6` or `C1-C5`. For dual-error diagrams, metadata is stored in `TYPE_DUAL`; evaluation scripts treat these as dual-error examples and, when a single class label is needed, use the first recorded error as the primary ground truth.