OmniSch: A Multimodal PCB Schematic Benchmark for Structured Diagram Visual Reasoning

Taiting Lu[1*], Kaiyuan Lin[1*], Yuxin Tian[2], Yubo Wang[1], Muchuan Wang[2], Sharique Khatri[1], Akshit Kartik[1], Yixi Wang[1], Amey Santosh Rane[3], Yida Wang[4], Yifan Yang[5✉], Yi-Chao Chen[4], Yincheng Jin[3], and Mahanth Gowda[1✉]

[1] Pennsylvania State University, Pennsylvania, USA
[2] Independent Researcher
[3] Binghamton University, New York, USA
[4] Shanghai Jiao Tong University, Shanghai, China
[5] Microsoft Research, Shanghai, China
mahanth.gowda@psu.edu, yifanyang@microsoft.com

Abstract. Recent large multimodal models (LMMs) have made rapid progress on visual grounding, document understanding, and diagram reasoning tasks. However, their ability to convert Printed Circuit Board (PCB) schematic diagrams into machine-readable, spatially weighted netlist graphs that capture component attributes, connectivity, and geometry remains largely underexplored, even though such graph representations are the backbone of practical electronic design automation (EDA) workflows. To bridge this gap, we introduce **OmniSch**, the first comprehensive benchmark designed to assess LMMs on schematic understanding and spatial netlist graph construction. OmniSch contains 1,854 real-world schematic diagrams and supports four tasks: **(1) visual grounding** for schematic entities, with $109.9K$ grounded instances aligning $423.4K$ diagram semantic labels to their visual regions; **(2) diagram-to-graph reasoning**, which requires understanding topological relationships among diagram elements; **(3) geometric reasoning**, constructing layout-dependent edge weights for each connection; and **(4) tool-augmented agentic reasoning** for visual search, which invokes external tools to accomplish (1) - (3). Our results reveal substantial gaps in current LMMs' ability to interpret schematic engineering artifacts, including unreliable fine-grained grounding, brittle layout-to-graph parsing, inconsistent global connectivity reasoning, and inefficient visual exploration.

Keywords: Visual Grounding · PCB Schematic · Structured Graph Understanding

*Equal contribution.

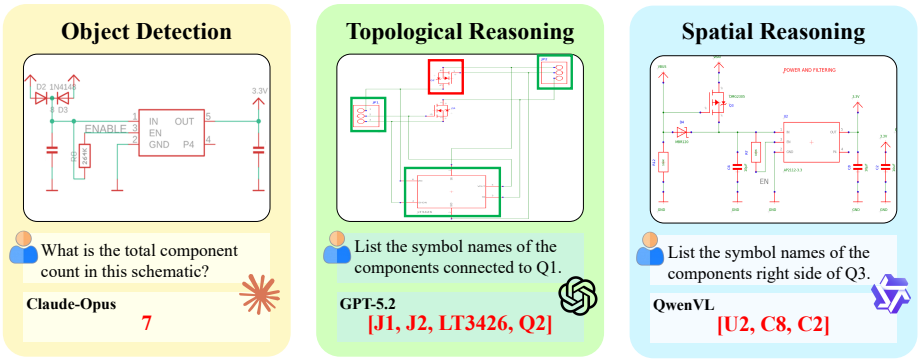


Fig. 1: Large multimodal models fail to reliably perform core visual understanding tasks on structured schematic diagrams. Errors in component detection, topological connectivity reasoning, and spatial layout interpretation highlight persistent limitations in handling the compositional and relational complexity inherent in schematic diagrams.

1 Introduction

Printed Circuit Board (PCB) schematic diagrams are essential design representations for electronic design automation (EDA) systems, spanning applications from everyday smartphones to industrial automotive platforms. However, most real-world schematic designs are proprietary and rarely publicly available, and even when accessible they require labor-intensive expert annotation, leaving most open design knowledge confined to textbooks, research papers, and patents, where circuits are presented as human-readable diagrams rather than machine-parsable netlists. To enable downstream EDA workflows, schematic diagrams must be converted into machine-readable netlists that preserve component connectivity and parameters for analysis, simulation, and layout synthesis. Recently, several works [11,20,44] have employed YOLO models [15] to automate this conversion from textbook schematics to netlists and have released public datasets. As shown in Table 1, existing datasets fall short of capturing real-world complexity, covering only a limited number of entity types and samples compared to the thousands of component types in practical designs.

In recent years, large multimodal models (LMMs), such as GPT-5 [31] and Gemini 2.5 Flash-Lite [16], have demonstrated strong capabilities in open-ended reasoning, geometric reasoning, multimodal grounding, and graph understanding [13, 14, 26, 29, 42]. These advances suggest new opportunities for applying LMMs to engineering workflows, particularly for structured visual artifacts such as PCB schematic diagrams and for generating their corresponding netlist graphs. However, schematic-to-netlist graph generation remains largely underexplored and lacks standardized benchmarks, raising a key question: *Can current LMMs reliably recover component attributes, fine-grained connectivity, and geometry-aware edges from real-world, densely cluttered schematics?*

To answer this question, we conducted preliminary experiments with several state-of-the-art LMMs, including Claude Opus 4.6 [7], GPT-5.2 [31], and

Qwen3.5-VL [10]. These experiments evaluated their performance on schematic diagram understanding tasks, including object detection, topological connectivity reasoning, and spatial layout interpretation. As illustrated in Fig. 1, each model fails at least one of these tasks. These results indicate that current LMMs still struggle to achieve precise visual grounding and topology-consistent reasoning, limiting their effectiveness in schematic-to-netlist generation, which requires accurate recognition of schematic entities, faithful reconstruction of connectivity, and geometry-aware reasoning over layout-dependent spatial relationships. Recent benchmarks, such as Netlistify [20], MASALA-CHAI [11], and PCB-Bench [24], have begun to evaluate LMMs on schematic diagram understanding. However, they primarily focus on simplified schematics with relatively few entities, such as Simulation Program with Integrated Circuit Emphasis (SPICE)-style circuits, underscoring the need for benchmarks that enable a more comprehensive evaluation of LMMs.

To bridge this gap, we introduce **OmniSch**, a large-scale benchmark of real-world schematic diagrams that enables comprehensive evaluation of LMMs across a diverse set of multimodal schematic understanding tasks. As shown in Fig. 2, OmniSch provides 1,854 high-quality real-world schematic designs with comprehensive annotations of $109.9K$ symbols ($3.4K$ unique types) and $245.4K$ pins (with bounding boxes and semantic attributes), together with $423.4K$ text instances and $219.8K$ net graphs with spatial weights and semantic attributes. We further propose structured evaluation protocols to assess netlist graphs generated by LMMs. Our benchmark evaluates four core capabilities: **(i) visual grounding** of schematic entities and text, **(ii) diagram-to-graph reasoning** for electrical connectivity extraction, **(iii) geometry-aware spatial reasoning** for constructing spatially weighted netlist graphs, and **(iv) tool-augmented agentic visual search** for locating query-specified circuit entities (components, connections, and graphs) within schematic diagrams. In summary, the contributions of this work are threefold:

1. **We present OmniSch, a comprehensive benchmark for evaluating LMMs on schematic-to-graph reasoning,** providing 1,854 real-world schematic designs with fine-grained annotations of symbols, pins, text instances, and spatially weighted net graphs, together with structured evaluation protocols.
2. **We systematically evaluate state-of-the-art LMMs for schematic-to-netlist graph generation under multiple settings,** including single-shot prompting and tool-augmented agentic workflows with active visual search, to quantify the impact of iterative visual exploration across diverse schematic understanding tasks.
3. **We provide unified task formats and standardized evaluation protocols for schematic understanding,** enabling reproducible benchmarking and consistent comparison of tool-augmented agentic frameworks for schematic-to-netlist graph generation. Code and datasets are available at `https://omnisch.com/`.

Table 1: Comparison of publicly available schematic-to-netlist benchmarks and datasets. Abbreviations: Sch.—schematic diagram; Sp.Netlist—spatially weighted netlist.

Dataset	Images	Avg. Sym. Img	Symbol Cat.	BBox	Spatial Netlist	EDA Render Engine	Annotation				Year
							Symb.	Pin	Text	Net	
CGHD [41]	2,424	42	45	✓	✗	✗	✓	✗	✓	B-Net	2021
AMSNet [39]	894	10	12	✗	✗	✗	✓	✗	✗	B-Net	2024
Masala-CHAI [11]	7,500	12	12	✗	✗	✗	✓	✗	✗	B-Net	2024
AMSNet 2.0 [36]	2,686	10	12	✗	✗	✗	✓	✗	✗	B-Net	2025
Image2Net [44]	2,914	15	22	✗	✗	✗	✓	✗	✗	B-Net	2025
MuaLLM [1]	2,914	29	22	✗	✗	✗	✓	✗	✗	B-Net	2025
Netlistify [20]	100,000	11	16	✓	✗	✗	✓	✗	✗	B-Net	2025
SINA [5]	662	–	10	✓	✗	✗	✓	✗	✗	SW-Net	2026
OmniSch (ours)	**1,854**	**59**	**3,480**	✓	✓	✓	✓	✓	✓	SW-Net	2026

2 Related Work

2.1 Visual Grounding

LMMs such as Kosmos-2, Ferret, Groma, GLAMM, and LLaVA-Grounding [25, 27, 32, 34, 46] have rapidly evolved beyond global image–text alignment to produce verifiable region-level outputs through location tokens or box-annotated generation, enabling phrase grounding and referring expression comprehension within a single instruction. Current visual grounding methods [27,32,34,46] typically follow either end-to-end architectures or referential grounding formulations. Concurrently, emerging benchmarks [28, 47, 48] increasingly emphasize challenging settings such as dense visual clutter and compositional reasoning to better expose the remaining limitations of current approaches. More recently, agentic visual search has emerged as a complementary paradigm for visual grounding, framing the task as an iterative decision-making process, as demonstrated by MM-ReAct, DeepEyes, and V* [43, 45, 49]. Despite this strong progress, LMM-based visual grounding remains an unsolved problem in several important settings, including fine-grained localization in crowded scenes, structured diagrams, and relational referring expressions.

2.2 Benchmarks for Schematic-to-Netlist

In recent years, schematic-to-netlist conversion has advanced alongside the release of several public datasets and evaluation benchmarks. AMSNet [39] provides 894 analog mixed-signal (AMS) schematic diagrams paired with SPICE netlists, while AMSNet 2.0 [37] extends this direction to 2,686 circuits by adding Spectre-formatted netlists, OpenAccess digital schematics, and positional annotations. Netlistify [20] introduces a modular deep learning pipeline together with a 100,000-image dataset containing component annotations (e.g., bounding boxes and orientations) and predefined labels for net extraction, targeting HSPICE-compatible netlist reconstruction. Beyond printed analog schematics, CGHD [41] provides a dataset of 2,424 hand-drawn circuit diagrams. More recently, PCB-Bench [24] has broadened the scope to PCB placement, routing,

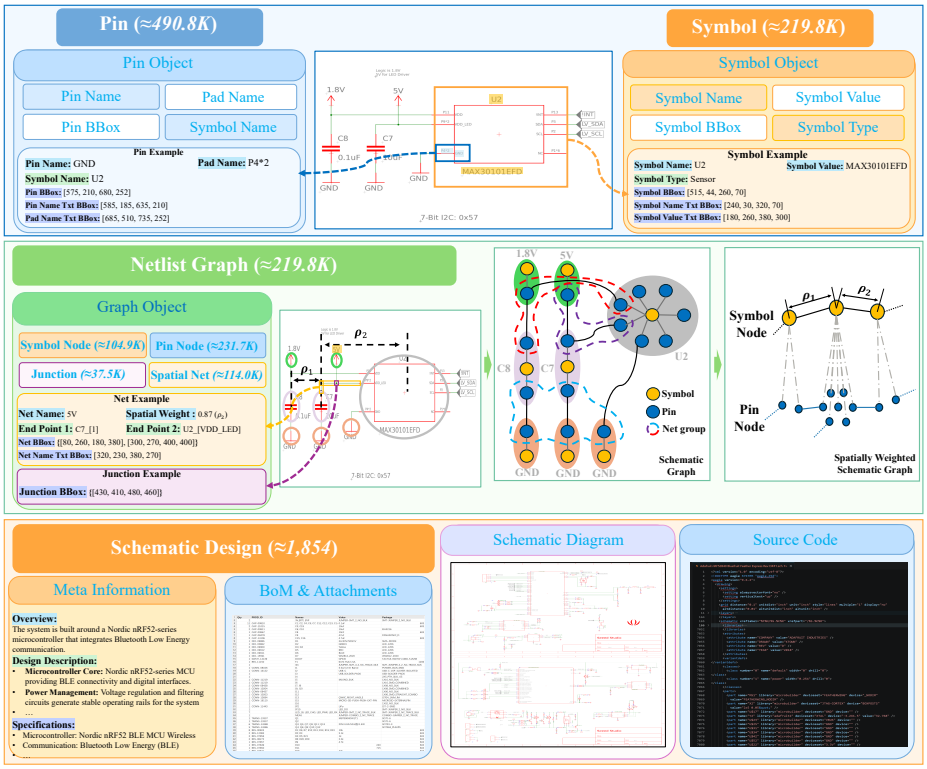


Fig. 2: Overview of OmniSch benchmark with representative cases.

and design comprehension, including approximately 3,700 question-answering instances spanning 174 schematic designs. However, current datasets primarily focus on SPICE-style schematic diagrams, which typically contain a limited number of entity types and samples compared with the thousands of component types found in real-world schematic designs. Similarly, existing methods remain confined to recognizing a narrow, predefined set of symbols under fixed rules, lacking the multimodal reasoning capabilities required to interpret authentic schematic design artifacts.

3 Benchmark Construction

3.1 Task Formulation

To provide a comprehensive evaluation framework for schematic-to-netlist generation, we define five core capabilities. Detailed descriptions of these capabilities are as follows.

- **Instance Detection.** Accurately localizing electronic components and pins is fundamental to schematic understanding. This capability is evaluated through symbol detection and pin detection tasks, in which models must produce precise bounding boxes in dense, symbol-rich technical diagrams.

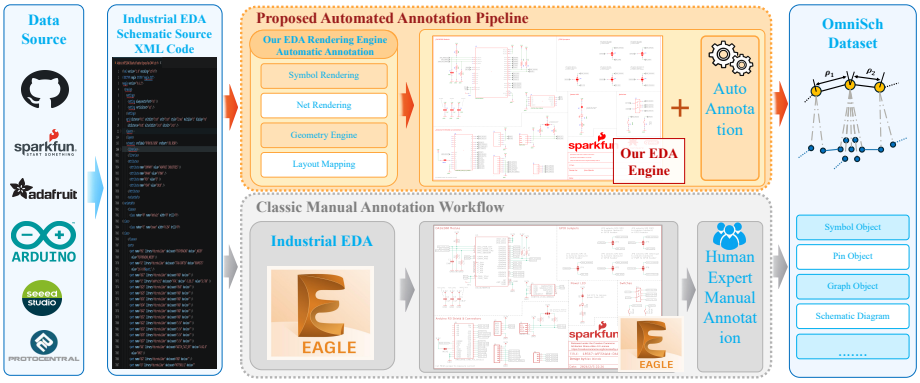


Fig. 3: Comparison between different data annotation paradigms. (a) Manual annotation: relying on human experts for labeling. **(b) Auto annotation:** our OmniSch directly renders schematic source code and automatically labels data via our EDA rendering engine.

• **Symbol Classification.** Recognizing component types and their orientations is essential for establishing the functional identity of circuit elements. This capability is assessed through symbol type classification and orientation recognition tasks, requiring domain-specific visual discrimination across diverse schematic styles.

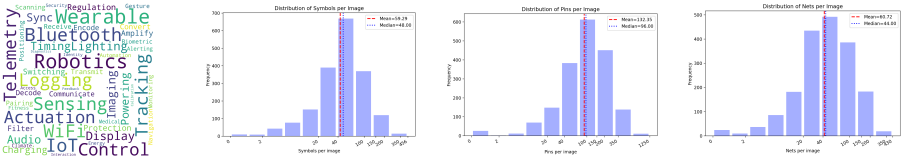
• **Text Referring.** Schematics contain densely distributed text serving distinct semantic roles, including symbol names, parameter values, net labels, and pin annotations. The ability to ground each text instance to its associated symbol or pin is evaluated through text-to-symbol linking and semantic role disambiguation tasks.

• **Topological Relation Reasoning.** Understanding electrical connectivity from visual layouts requires multi-hop structural reasoning beyond local perception. This capability is assessed through pin–symbol membership prediction and pin–pin connectivity prediction tasks, in which models must infer how components are electrically connected from spatial context.

• **Heterogeneous Graph Construction.** Beyond pairwise connectivity reasoning, converting an entire schematic into a unified structured representation requires holistic diagram parsing. This capability is evaluated by requiring models to generate a complete spatially weighted heterogeneous graph comprising symbol nodes, pin nodes, membership edges, and net-induced connectivity edges, faithfully encoding both the hierarchical composition and the global topology of the circuit in a single pass.

3.2 Annotation Curation

The OmniSch benchmark is constructed through a carefully designed pipeline for data collection and processing, as illustrated in Fig. 3. We began by collecting real-world schematics from multiple open-source communities, such as SparkFun, Arduino, Adafruit, and GitHub [2, 8, 19, 33, 35, 38]. Based on these sources, we



(a) 20 Topic distribution. **(b)** Symbols per image. **(c)** Pins per image. **(d)** Nets per image.

Fig. 4: Statistical overview of the OmniSch benchmark. The dataset encompasses a diverse range of electronic domains, comprising 1-440 symbols, 1-1200 pins, 1-400 nets, and 1-1600 text instances. This large-scale diversity provides a comprehensive benchmark for the automatic generation and evaluation of schematic netlists.

curate schematic design files created with the industrial EDA tool Autodesk EAGLE [9], because its XML-based schematic format enables scalable extraction of diagram instances and connectivity for automated annotation.

Unlike traditional workflows that rely on time-consuming, costly, and labor-intensive manual annotation (Fig. 3), we propose a source-driven automated pipeline via our custom EDA generative rendering engine. Our EDA generative rendering engine parses EAGLE XML source files and re-simulates the native schematic rendering process to produce high-fidelity schematic images using EAGLE-defined rendering primitives, while simultaneously exporting pixel-aligned annotations at multiple granularities—from fine-grained entity localization and semantic labeling (e.g., symbol/pin bounding boxes and attributes) to net-level connectivity represented as spatially weighted netlist graphs. Two senior EDA engineers further validated all 1,854 schematics/annotations, with no observed errors.

3.3 Statistics of OmniSch Benchmark

As illustrated in Fig. 4 and Fig. 2, OmniSch comprises 1,854 real-world schematic diagrams spanning diverse application domains (e.g., robotics, wireless, and sensing), providing a rich resource for supervised training and fine-tuning on practical schematic understanding and EDA tasks.

• **Schematic Diagram Instances.** We construct a fine-grained instance-level annotation set that localizes and semantically describes the visual elements in a schematic, including symbols, pins, and text, as illustrated in Fig. 2. For each symbol instance, we annotate its bounding box, symbol type, orientation, and semantic attributes (symbol name and symbol value). For each pin instance, we annotate its bounding box, parent symbol ID, and pin-level attributes (pin name and pad name). For each text instance, we annotate its location, content, and reference target.

• **Connectivity Annotations.** We construct topological net-level connectivity annotations by explicitly annotating pin–pin edges that indicate which pins are electrically connected in the schematic. Each edge is augmented with a spatial weight computed as the Euclidean distance between the centers of the two parent symbols to which the connected pins belong, normalized by the schematic

image size. When a connection is labeled in the schematic, we additionally annotate the corresponding net name as an attribute of the same edge.

• **Schematic Graphs.** We construct a heterogeneous, spatially weighted schematic graph that unifies instances and connectivity into a single representation. The graph contains symbol nodes and pin nodes, includes membership edges linking each pin to its parent symbol, and includes net-induced connectivity edges between pins (and optionally derived symbol links), each associated with net attributes and spatial weights. This representation preserves the global topology while explicitly retaining the local clustering structure that reflects the functional blocks represented in the schematic.

3.4 Evaluation Criteria

We adopt seven evaluation metrics, each tailored to a specific task category. In the following, we present an overview of these evaluation metrics and their applicability to different tasks.

• **Instance Detection Type.** To evaluate the model's ability to localize symbols and pins, we employ the F1 score to assess whether each ground-truth instance is successfully detected, without requiring precise spatial overlap at this stage.

• **Semantic Attribute Type.** For text referring and semantic attribute extraction, we adopt the F1 score with exact string matching to determine whether the grounded text precisely matches the ground-truth annotation. Approximate or similarity-based matching is deliberately excluded, as real-world schematic-to-netlist conversion demands rigorous character-level correctness in component designators and parameter values.

• **Localization Type.** For symbol and pin bounding box evaluation, the IoU score is applied to quantify the spatial overlap between predicted regions and ground-truth annotations, measuring the model's fine-grained localization precision.

• **Connectivity Type.** To evaluate topological relation reasoning, we employ the F1 score to assess the matching accuracy of predicted pin–pin connections against ground-truth nets and additionally report accuracy to measure the coverage of correctly detected nets across the entire circuit.

• **Graph Structure Type.** For heterogeneous graph construction, we adopt the IoU score over edge sets to evaluate the structural overlap between the predicted graph and the ground-truth graph, capturing how faithfully the generated representation preserves both membership and connectivity edges.

• **Spatial Layout Type.** To assess whether the predicted graph preserves the relative spatial arrangement of the original schematic, we employ Kendall's tau [22] rank correlation coefficient to measure the consistency of pairwise spatial orderings between predicted and ground-truth node positions.

3.5 Netlist Graph Matching Protocol

To the best of our knowledge, there is no standardized protocol for matching netlist graphs predicted by LMMs with ground-truth netlist graphs. We represent each circuit as a typed heterogeneous graph $G = (V^{\text{sym}} \cup V^{\text{pin}}, E^{\text{mem}} \cup E^{\text{conn}})$, where symbol nodes denote component instances and pin nodes denote their physical pins. Membership edges E^{mem} connect symbols to their pins, while connectivity edges E^{conn} connect pins that belong to the same electrical net. Direct graph comparison is challenging because predicted and ground-truth graphs do not share node identities, and geometric IoU is unreliable under imprecise LMM bounding boxes. We therefore first establish node correspondence and then evaluate attributes, edges, and graph consistency under the induced mapping.

We perform alignment hierarchically. Symbol nodes are first matched globally between the predicted graph G_p and the ground-truth graph G_g using maximum-weight bipartite matching. Then, for each matched symbol pair, their associated pin nodes are matched locally under the parent-symbol constraint. This ensures that pin-level connectivity is compared only after the corresponding component instances have been aligned.

We use two complementary matching scores. The first is a semantic score, which tests whether the model recovers the correct component and pin identities:

$$\begin{aligned} S^{\text{sem}}_{\text{sym}}(\hat{u}, u) = w_t \mathbf{1}[\hat{T} = T] + w_n S_{\text{txt}}(\hat{n}, n) + w_v S_{\text{txt}}(\hat{v}, v) \\ + w_p S_{\text{exp}}(\hat{d}^{\text{pin}}, d^{\text{pin}}) + w_e S_{\text{exp}}(\hat{d}^{\text{net}}, d^{\text{net}}), \end{aligned} \tag{1}$$

where T, n, and v denote the component type, reference name, and value, while d^{pin} and d^{net} are lightweight structural tie-breakers measuring pin count and net connectivity. For pin nodes, we use

$$S^{\text{sem}}_{\text{pin}}(\hat{q}, q) = w_n S_{\text{txt}}(\hat{t}, t) + w_p S_{\text{txt}}(\hat{p}, p) + w_a S_{\text{txt}}(\hat{a}, a) + w_c S_{\text{exp}}(\hat{c}, c), \tag{2}$$

where t, p, and a are the net name, pin name, and pad name, and c is the size of the connected component containing the pin in the pin–pin graph.

The second is a structure-aware score, which evaluates whether the predicted graph preserves topology and layout when textual labels are not used:

$$\begin{aligned} S^{\text{str}}_{\text{sym}}(\hat{u}, u) = w_p S_{\text{exp}}(\hat{d}^{\text{pin}}, d^{\text{pin}}) + w_e S_{\text{exp}}(\hat{d}^{\text{net}}, d^{\text{net}}) + w_s S_{\text{exp}}(\hat{d}^{\text{sym}}, d^{\text{sym}}) \\ + w_r S_{\text{pos}}(\hat{r}, r), \end{aligned} \tag{3}$$

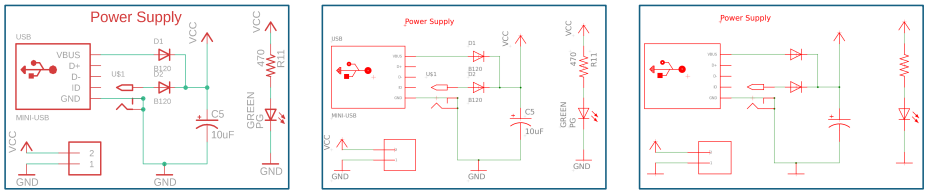
where d^{sym} denotes symbol-level adjacency degree and S_{pos} measures agreement in coarse relative position. For pins, the structure-aware score uses only connectivity context:

$$S^{\text{str}}_{\text{pin}}(\hat{q}, q) = w_c S_{\text{exp}}(\hat{c}, c). \tag{4}$$

Here S_{txt} is a normalized soft string similarity and $S_{\text{exp}}(a, b) = \exp(-|a - b|/\tau)$ is an exponential similarity for scalar structural features. The semantic protocol emphasizes labels and attributes, whereas the structure-aware protocol removes textual cues and matches nodes using topology and coarse layout. After alignment, all downstream metrics are computed on the matched graph pairs.

4 Experiments and Findings

4.1 Experimental Setup



(a) Diagram by industrial EDA with DPI 150 (b) Diagram by our EDA with DPI 150 (c) Diagram by our EDA without symbol text

Fig. 5: Synthesized schematic variations generated by our custom EDA generative rendering engine. The first diagram shows the original export from the industrial EDA tool; all remaining diagrams are rendered by our engine under controlled variations. (a) industrial EDA export; (b) our EDA export; (c) our EDA export without symbol names and values.

Study Setup. We evaluated the following LMMs: GPT-5.2 [31], GPT-5-mini [31], Claude-Sonnet-4.6 [7], Gemini-2.5-Flash-Lite [16], Gemini-3.1-Pro-Preview [17], Qwen3-8B-Instruct [10], Qwen3-VL-235B-A22B [10], Ministral-14B [4], and LLaMA-4-Maverick-400B [3]. We evaluated these models alongside a classical vision pipeline baseline and our proposed agentic framework, as described below. For training the classical pipeline components (e.g., object detectors), we randomly split OmniSch into 80% training, 10% validation, and 10% test sets, with the test set held out for evaluation across all models. We conduct three studies: **Study 1.** We evaluate all tested LMMs under a unified **zero-shot** setting on the test split of OmniSch across the three dataset variants in Fig. 5: real schematics, synthesized schematics with textual labels, and synthesized schematics without symbol-name or symbol-value text. This assesses their end-to-end ability to convert a schematic diagram into a netlist graph without intermediate supervision. For completeness, we also report end-to-end results for our classical vision pipeline and the proposed agentic framework on the same test split. **Study 2.** We conduct an ablation study on the test split of OmniSch to compare direct end-to-end prompting, few-shot prompting, and engineer-guided chain-of-thought prompting for extracting netlist graphs from schematic images. This study quantifies the contribution of each prompting strategy to end-to-end schematic understanding and graph construction. **Study 3.** We apply a unified tool-augmented agentic workflow across all evaluated LMMs to quantify the contribution of tool-augmented active visual search to attribute-level recognition of symbol names, symbol values, pin names, pad names, and net names.

Implementation of Classical Vision Baseline. We construct a modular classical vision pipeline as a baseline for schematic understanding. The pipeline combines YOLO11 [21] for visual perception, PaddleOCR [40] for text extraction, and rule-based modules to recover symbols, pins, nets, and circuit connectivity. We further develop a multimodal text grounding model to associate OCR tokens

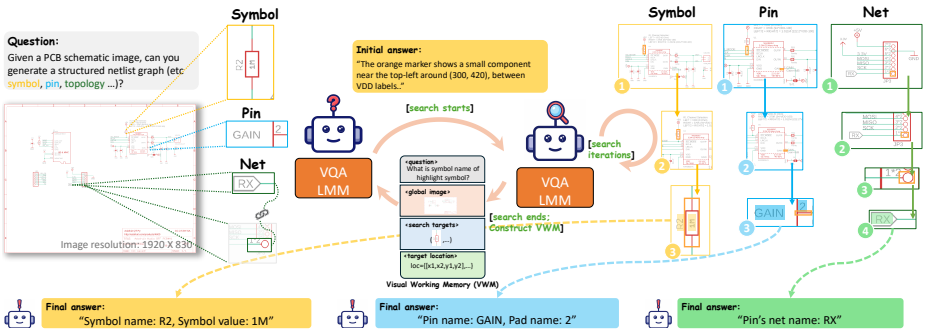


Fig. 6: Overview of the ReAct-based agentic framework for evaluating how LMMs use tools and how tool use affects schematic-to-netlist conversion performance.

with schematic instances and extract symbol names, values, and pin/pad labels, which are then integrated into a global netlist.

Implementation of the LMM Agentic Framework. As shown in Fig. 6, we implement a ReAct-based evaluation framework to benchmark LMMs on VQA-style schematic understanding, focusing on text referring tasks including symbol names, component values, pin names, pad names, and net names. Given a high-resolution schematic and a structured query, LMMs iteratively crop image regions and refine predictions through active visual search.

4.2 Main Results

Task 1: End-to-End Schematic-to-Netlist Generation. As shown in Table 2, the tested LMMs consistently underperform the YOLO11-based classical vision pipeline in end-to-end schematic-to-netlist generation by a large margin. Among LMMs, *Claude Sonnet 4.6*, *Gemini 3.1 Pro*, *Gemini 2.5 Flash*, and *GPT-5.2* achieve the strongest overall results. *Claude Sonnet 4.6* performs best on synthesized schematics for symbol and pin detection, while *Gemini 2.5 Flash* leads LMMs in symbol detection on real schematics. *Gemini 3.1 Pro* achieves the best LMM pin detection on real schematics and the highest Kendall's τ, but relies heavily on text cues, with performance degrading when symbol names and values are removed. Overall, LMMs still lag behind the classical pipeline in graph reconstruction quality. Although several LMMs slightly exceed the classical pipeline in net-name accuracy on real image inputs, their much lower net-name coverage shows that they correctly label only a limited subset of nets.

Task 2: Ablation Analysis of LMM Prompting Strategies for Netlist Construction from Schematic Images. Table 3 compares prompting strategies for GPT-5.2. Few-shot prompting improves symbol detection and symbol-level attributes, but these gains do not improve topology reconstruction, with graph IoU remaining nearly unchanged and Kendall's τ near zero. Adding explicit instructions further degrades overall performance, especially pin detection, net-name coverage, and graph IoU. These results suggest that prompting helps

Table 2: Evaluation performance of existing LMMs on end-to-end schematic-to-netlist generation. Image inputs include real schematics (*Real*; Fig. 5(a)), synthesized schematics with textual labels (*Synth.*; Fig. 5(b)), and synthesized schematics without symbol-name and symbol-value text (*Synth. w/o txt*; Fig. 5(c)). **Bold** and underlined values denote the best and second-best results among LMMs for each metric.

Model	Size	Image Input	Symbol				Pin			Net		Graph	
			Detection	Attribute			Detection	Attribute		Net-name			
			F1	Name	Value	Type	F1	PinName	PadName	Acc	Cov	IoU	Kendall's τ
GPT 5.2	-	Real	0.5461	0.1720	0.2215	0.3272	0.3534	0.1750	0.4993	0.2924	0.3764	0.1064	0.0924
		Synth.	0.4692	0.1631	0.2174	0.3266	0.2822	0.4205	0.5435	0.2676	0.3589	0.1159	0.0593
		Synth. w/o txt	0.4958	–	–	**0.3970**	0.3508	0.4001	0.5008	0.1674	0.2477	0.0434	0.0114
GPT-5 mini	-	Real	0.3131	0.1924	0.2170	0.2943	0.3096	0.3443	0.5517	0.1605	0.2282	0.0751	0.0934
		Synth.	0.3156	**0.1639**	0.2044	0.3087	0.2953	0.4515	**0.5528**	0.1333	0.1895	0.0734	0.1850
		Synth. w/o txt	0.2464	–	–	0.3105	0.2648	0.4127	**0.5544**	0.0642	0.0978	0.0248	0.0037
Claude Sonnet 4.6	-	Real	0.6205	0.1395	0.2576	**0.5136**	**0.4883**	0.0241	0.4762	0.2756	0.3976	0.1236	0.1911
		Synth.	**0.5073**	0.1154	**0.2417**	**0.3472**	**0.3407**	0.0029	0.5065	0.2564	0.3664	0.1117	0.1951
		Synth. w/o txt	**0.5273**	–	–	0.3375	**0.3908**	0.0194	0.4468	**0.2028**	**0.2908**	**0.0513**	0.0148
Gemini 2.5 Flash	-	Real	**0.7057**	**0.2901**	**0.3494**	0.4982	0.4414	0.3524	0.5185	**0.3887**	**0.5605**	**0.1945**	0.2651
		Synth.	0.4516	0.1442	0.2192	0.3380	0.3156	**0.4882**	0.5124	**0.2739**	**0.3904**	**0.1205**	0.0713
		Synth. w/o txt	0.3077	–	–	0.2342	0.2721	0.3971	0.5161	0.0102	0.0215	0.0064	0.0018
Gemini 3.1 Pro-Preview	-	Real	0.6406	0.2780	0.3241	0.4123	0.4726	0.0269	0.0906	0.3617	0.4514	0.1151	**0.6438**
		Synth.	0.3655	0.1107	0.1829	0.2654	0.2252	0.0026	0.1320	0.1727	0.2501	0.0630	**0.5268**
		Synth. w/o txt	0.2647	–	–	0.0948	0.0701	0.1052	0.1010	0.0004	0.0008	0.0003	**0.0824**
Llama 4 Maverick	400B	Real	0.2728	0.0902	0.1971	0.2652	0.1900	**0.4915**	0.5394	0.0800	0.1368	0.0391	-0.0046
		Synth.	0.2510	0.1113	0.1882	0.2466	0.1741	0.4571	0.4604	0.0656	0.1076	0.0330	0.0134
		Synth. w/o txt	0.0878	–	–	0.1004	0.0705	**0.4252**	0.3261	0.0023	0.0050	0.0014	-0.0152
Ministral 14B	14B	Real	0.4076	0.1071	0.2066	0.3734	0.3295	0.3268	**0.6332**	0.1354	0.2282	0.0963	0.0335
		Synth.	0.5023	0.0959	0.1945	0.3430	0.3120	0.3119	0.5111	0.1292	0.2634	0.0631	0.0457
		Synth. w/o txt	0.0704	–	–	0.0754	0.0594	0.3877	0.3720	0.0004	0.0010	0.0003	-0.0509
Qwen3-VL 8B	8B	Real	0.1040	0.0268	0.0340	0.0354	0.0187	0.0070	0.0104	0.0038	0.0089	0.0016	0.0000
		Synth.	0.1060	0.0256	0.0325	0.0347	0.0188	0.0071	0.0103	0.0040	0.0090	0.0016	0.0000
		Synth. w/o txt	0.0043	–	–	0.0008	0.0006	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Qwen3-VL 235B-A22B	235B	Real	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Synth.	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		Synth. w/o txt	0.0000	–	–	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Classical vision Pipeline	-	Real	0.8812	0.5078	0.2487	0.7875	0.9174	0.6245	0.4100	0.2581	0.9451	0.2114	0.0399
		Synth.	0.9518	0.6087	0.5977	0.8319	0.9047	0.8170	0.7347	0.3444	0.9369	0.1841	0.1263
		Synth. w/o txt	0.9199	–	–	0.8317	0.8554	0.9812	0.8536	0.5922	0.9346	0.2894	0.0201

Note. Qwen3-VL-235B-A22B consistently returned empty JSON outputs. Consequently, all reported metrics are 0.

local recognition but has limited, and sometimes negative, impact on structural reasoning.

Task 3: ReAct-Based Visual Search Performance. Table 4 evaluates each model's perception, spatial understanding, and logical reasoning by measuring attribute-level recognition performance for symbol names and values, as well as pin names and pads, under zoom-based inference with and without bounding-box guidance. Overall, providing precise bounding boxes consistently and substantially improves attribute-matching precision, indicating that explicit localization cues are critical for reliable fine-grained recognition. In addition, we introduce an API that enables LMMs to iteratively scroll/pan the viewport based on their intermediate responses, allowing them to actively refine what they "see." This interactive window control yields a marked performance gain and demonstrates the effectiveness of external perception for schematic understanding.

4.3 Main Finding

Finding 1. Most LMMs rely heavily on textual cues for symbol recognition. As shown in Table 2, removing symbol name and value texts generally reduces detection performance. For example, the F1 score of *Gemini 2.5 Flash* drops from 0.4516 to 0.3077. In contrast, *GPT-5.2* and *Claude Sonnet 4.6* remain comparatively robust, indicating greater reliance on visual structure. Interest-

Table 3: Ablation study on prompting strategies for GPT-5.2 on schematic-to-netlist generation. We compare zero-shot prompting, few-shot prompting, and few-shot prompting with explicit instructions.

Prompt	Symbol				Pin			Net		Graph	
	Detection	Attribute			Detection	Attribute		Net-name			
	F1	Name	Value	Type	F1	PinName	PadName	Acc	Cov	IoU	Kendall's τ
Zero-shot	0.4885	0.1356	0.2178	0.2862	**0.2700**	0.5250	**0.5446**	0.2258	**0.3275**	**0.1052**	**0.0003**
Few-shot	**0.6216**	**0.1564**	**0.3026**	**0.3358**	0.2612	0.0299	0.4244	**0.2450**	0.3252	0.1017	-0.0019
Few-shot + Instruction	0.4690	0.1464	0.2001	0.2877	0.0902	**0.5525**	0.5169	0.0727	0.1099	0.0278	-0.0001

Table 4: Results of the agentic evaluation. *w/o GT BBox* denotes full-image search, where LMMs must localize the target before performing visual search; *w/ GT BBox* denotes detector-assisted search, where a bounding box is provided and LMMs only expand it without re-localizing the target. Input images use the schematic in Fig. 5(b). Δ denotes the relative improvement (%) of *w/ GT BBox* vs. *w/o GT BBox*; * denotes metrics computed only on instances with visible text.

Model	Size	Setting	Symbol					Pin				Net
			Name	Name*	Value	Value*	Type	PinName	PinName*	PadName	PadName*	NetName
Commercial chatbot systems												
GPT-5.2	-	w/o GT BBox	0.6948	0.6457	0.6610	0.6507	0.6886	**0.7892**	0.7621	0.2654	0.1298	–
		w/ GT BBox	0.8565	0.8482	0.8088	0.8252	0.8458	**0.9056**	0.8567	**0.8426**	0.9473	0.6926
		Δ	+23.28%	+31.36%	+22.36%	+26.82%	+22.83%	+14.76%	+12.41%	+217.57%	+629.43%	–
GPT-5 Mini	-	w/o GT BBox	0.5539	0.5835	0.4983	0.4797	0.5292	0.6285	0.5545	0.1311	0.1138	–
		w/ GT BBox	0.8932	0.8739	0.8437	0.8799	0.8717	0.8694	0.7615	0.7839	0.8196	0.6517
		Δ	+61.26%	+45.36%	+69.31%	+72.01%	+64.72%	+38.33%	+37.34%	+497.94%	+620.21%	–
Claude-Sonnet-4.6	-	w/o GT BBox	0.5070	0.7414	0.6710	0.7368	0.7452	0.4982	0.5944	0.4022	0.6210	–
		w/ GT BBox	0.7955	0.7210	0.7430	0.7589	0.7202	0.8984	**0.8864**	0.6400	**0.9526**	0.6411
		Δ	+56.91%	+0.71%	+10.73%	+3.00%	-3.36%	+80.33%	+49.11%	+59.15%	+53.38%	–
Gemini-2.5-Flash-Lite	-	w/o GT BBox	0.2148	0.3991	0.2280	0.2129	0.2482	0.1052	0.1504	0.0998	0.0708	–
		w/ GT BBox	0.5123	0.7467	0.5181	0.4776	0.6156	0.8473	0.7870	0.7184	0.7220	0.3873
		Δ	+138.49%	+87.06%	+127.24%	+124.33%	+148.02%	+705.42%	+423.01%	+619.64%	+920.34%	–
Gemini-3.1-Pro-Preview	-	w/o GT BBox	**0.8093**	**0.8726**	**0.7833**	**0.8151**	**0.7870**	0.6943	**0.8211**	**0.5824**	**0.8927**	–
		w/ GT BBox	**0.8958**	**0.8983**	**0.8591**	**0.9013**	**0.8764**	0.8382	0.8140	0.7841	0.8545	**0.8911**
		Δ	+10.69%	+2.95%	+9.68%	+10.58%	+11.36%	+20.72%	-0.86%	+34.62%	-4.28%	–
Open-source MLLMs												
LLaMA-4-Maverick-400B	400B	w/o GT BBox	0.1270	0.1699	0.1389	0.0876	0.2034	0.3346	0.0455	0.1780	0.0273	–
		w/ GT BBox	0.7934	0.7229	0.7421	0.7579	0.7196	0.7586	0.4964	0.5276	0.5323	0.3758
		Δ	+524.72%	+325.37%	+434.41%	+765.41%	+253.77%	+126.70%	+991.21%	+196.40%	+1849.45%	–
Ministral-14B	14B	w/o GT BBox	0.0990	0.0784	0.1324	0.0421	0.1031	0.1725	0.0380	0.0524	0.0051	–
		w/ GT BBox	0.4179	0.7626	0.3882	0.3622	0.4399	0.3960	0.2742	0.3457	0.1301	0.1113
		Δ	+322.12%	+872.70%	+193.25%	+760.57%	+326.61%	+129.57%	+621.58%	+559.92%	+2450.98%	–
Qwen3-8B-Instruct	8B	w/o GT BBox	0.2337	0.0368	0.1126	0.0280	0.0361	0.6653	0.0106	0.5544	0.0013	–
		w/ GT BBox	0.5917	0.7057	0.5872	0.5725	0.6067	0.7418	0.5723	0.6207	0.4153	0.3021
		Δ	+153.20%	+2039.67%	+421.40%	+1948.21%	+1581.99%	+11.50%	+5299.06%	+11.96%	+31792.31%	–
Qwen3-VL-235B-A22B	235B	w/o GT BBox	0.1336	0.2315	0.1570	0.1187	0.2150	0.2099	0.1027	0.0909	0.0130	–
		w/ GT BBox	0.5753	0.7871	0.5792	0.5809	0.6566	0.7282	0.8283	0.7233	0.8380	0.4532
		Δ	+330.64%	+277.44%	+268.92%	+640.69%	+205.39%	+246.88%	+706.40%	+695.99%	+6346.15%	–

ingly, symbol name and value texts may even introduce noise for these models, although other textual elements (e.g., pin and pad names) can still provide useful contextual cues.

Finding 2. Topology reasoning remains a major challenge for current LMMs. Although symbol and pin detection achieve moderate performance, graph-level IoU remains low across models (Table 2), indicating inaccurate connectivity reconstruction. For example, *Gemini 2.5 Flash* achieves strong symbol detection (F1 = 0.7057 on real schematics) but only 0.1945 graph IoU, showing that accurate component detection does not necessarily lead to correct topology inference. The large variation in Kendall's τ further suggests instability in modeling global connectivity, highlighting circuit topology reasoning as a key bottleneck in end-to-end schematic understanding.

Finding 3. As shown in Table 2, pin-level perception is substantially more challenging than symbol detection. Across most models, pin detection, pin name

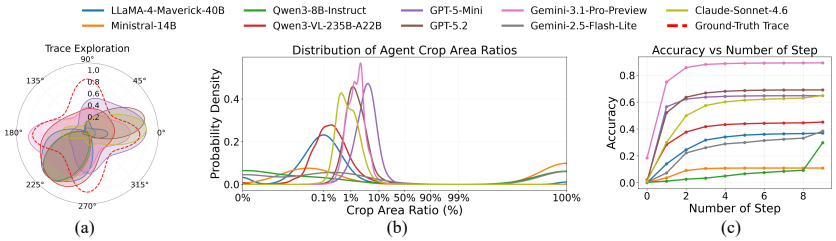


Fig. 7: Behavior analysis of LLM agents for net name tracing. (a) Exploration direction compared with the ground-truth trace. (b) Distribution of explored crop area ratios. (c) Tracing accuracy across different exploration steps.

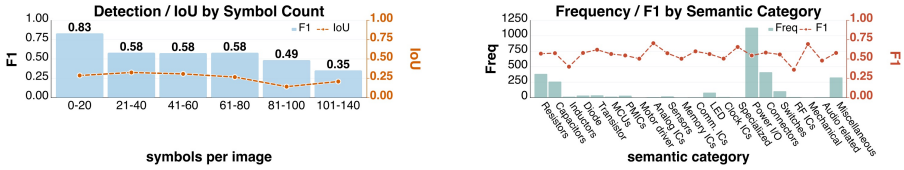
recognition, and pin value recognition consistently lag behind symbol detection. For example, on real schematics, *Gemini 2.5 Flash* achieves 0.7057 for symbol detection but only 0.4414 for pin detection. This gap is largely due to the high visual density of schematics: pins are represented by tiny primitives, whereas symbols have larger, more distinctive geometric structures. As a result, LMMs tend to focus on salient symbol features while overlooking fine-grained pin-level details and associated textual attributes.

Finding 4. Net name recognition and wire-text association remain challenging for current LMMs. Since net names are often overlaid on wire segments, models must correctly associate text with the corresponding traces. As shown in Table 2, performance remains limited across all models. Even the best-performing model, *Gemini 2.5 Flash*, achieves only 0.5605 net name coverage and 0.3887 accuracy, indicating that many net labels are missed or incorrectly assigned. These results highlight line tracing and text-to-wire association as persistent bottlenecks in schematic understanding.

Finding 5. Few-shot prompting improves GPT-5.2's attribute-level recognition, but adding explicit graph-construction instructions degrades performance across nearly all metrics. As shown in Table 3, few-shot prompting increases symbol detection F1 from 0.4885 to 0.6216 and symbol value matching from 0.2178 to 0.3026, whereas additional procedural instructions reduce both attribute-level and graph-level performance. This suggests that GPT-5.2 does not reliably follow explicit graph-construction rules, instead relying on holistic reasoning rather than deterministic procedural execution.

Finding 6. Table 4 shows that, when tool assistance is enabled, Gemini-3.1-Pro-Preview achieves the strongest overall performance on symbol and pin attribute recognition. This gain is largely driven by its ability to precisely zoom into the relevant image regions, which improves fine-grained attribute recognition. Consistent with this observation, Fig. 7 indicates that Gemini-3.1 exhibits the most concentrated crop distribution (i.e., the highest probability density ratio), suggesting more accurate and stable localization of target symbols compared with other models and, consequently, better downstream performance.

Finding 7. In Table 2, the model *Qwen3-VL-235B-A22B* obtains zero scores across all metrics because it consistently returns empty schema outputs for every evaluated image. Although all API calls succeeded, the model returned syntactically valid responses without the required structured fields. As a result, no valid



(a) Detection Accuracy & IoU vs. Density **(b)** Accuracy vs Symbol Frequency.

Fig. 8: Analysis of symbol detection performance with respect to schematic complexity and symbol frequency.

nodes or edges could be extracted during evaluation. We retain these results in the table for completeness and transparency, as they reflect the model's practical behavior under the benchmark's structured output requirements.

Finding 8. As shown in Fig. 8a, performance degrades with increasing schematic complexity, dropping below 50% F1 for schematics containing more than 80 symbols per image. As shown in Fig. 8b, we analyze symbol detection performance across the 24 semantic categories and their occurrence frequencies. Despite category frequencies varying by more than an order of magnitude, F1 remains within a relatively narrow range (approximately 0.5–0.7), indicating little correlation between component frequency and detection performance. Since visually similar components often belong to the same semantic category, our evaluation additionally requires correct symbol names and values, suggesting that the benchmark tests whether LMMs can distinguish fine-grained variants, including rare components.

Table 5: Cross-EDA-Tool Evaluation of Schematic-to-Netlist Graph IoU.

Model	Altium	EasyEDA	Multisim	Proteus	OrCAD
GPT-5.2	0.1548	0.0808	0.1461	0.1366	0.1360
Gemini 2.5-Flash	0.1214	0.1173	0.0212	0.0432	0.0115

Finding 9. We also evaluate frontier LMMs on schematics from five closed-source commercial EDA toolchains (Altium [6], EasyEDA [18], Multisim [30], OrCAD [12], and Proteus [23]) to evaluate cross-tool generalization. For each toolchain, we curate a dataset of 10 schematic images with manually annotated ground truth. As shown in Table 5, frontier LMMs consistently perform poorly on graph IoU across different EDA environments.

5 Conclusion

We introduce OmniSch, a large-scale real-world schematic benchmark with comprehensive symbol, pin, text, and net graph annotations and structured protocols for netlist graph generation. Using this benchmark, we systematically evaluate representative LMMs under single-shot prompting and tool-augmented agentic pipelines, identifying key factors affecting performance and showing that tool-augmented agents substantially improve schematic-to-netlist accuracy through active visual search. We hope OmniSch will facilitate future research on structured diagram understanding with LMMs.

References

1. Abbineni, P., Aldowaish, S., Liechty, C., Noorzad, S., Ghazizadeh, A., Fayazi, M.: MuaLLM: A multimodal large language model agent for circuit design assistance with hybrid contextual retrieval-augmented generation. arXiv preprint arXiv:2508.08137 (2025)
2. Adafruit Industries: Adafruit Industries. https://www.adafruit.com (2025), accessed: 2026-06-30
3. AI, M.: LLaMA 4 technical report. https://ai.meta.com (2025), accessed: 2026-06-30
4. AI, M.: Ministral 14B. https://mistral.ai (2024), accessed: 2026-06-30
5. Aldowaish, S., Karumanchi, Y., Chiang, K.C., Noorzad, S., Fayazi, M.: SINA: A circuit schematic image-to-netlist generator using artificial intelligence. arXiv preprint arXiv:2601.22114 (2026)
6. Altium LLC: Altium Designer. https://www.altium.com/altium-designer (2025), accessed: 2026-06-30
7. Anthropic: Claude Sonnet 4.6. https://www.anthropic.com (2025), claude model family; Accessed: 2026-06-30
8. Arduino: Arduino: Open-source electronics platform. https://www.arduino.cc (2026), accessed: 2026-06-30
9. Autodesk Inc.: Autodesk EAGLE: PCB design and schematic software. https://www.autodesk.com/products/eagle (2024), accessed: 2026-06-30
10. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-VL technical report. arXiv preprint arXiv:2511.21631 (2025)
11. Bhandari, J., Bhat, V., He, Y., Rahmani, H., Garg, S., Karri, R.: MASALA-CHAI: A large-scale SPICE netlist dataset for analog circuits by harnessing AI. arXiv preprint arXiv:2411.14299 (2024)
12. Cadence Design Systems: OrCAD X. https://www.cadence.com/en_US/home/tools/pcb-design-and-analysis/orcad.html (2025), accessed: 2026-06-30
13. Chen, B., Xu, Z., Kirmani, S., Ichter, B., Sadigh, D., Guibas, L., Xia, F.: SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14455–14465 (2024)
14. Cheng, A.C., Yin, H., Fu, Y., Guo, Q., Yang, R., Kautz, J., Wang, X., Liu, S.: SpatialRGPT: Grounded spatial reasoning in vision-language models. Advances in Neural Information Processing Systems **37**, 135062–135093 (2024)
15. Cheng, T., Song, L., Ge, Y., Liu, W., Wang, X., Shan, Y.: YOLO-World: Real-time open-vocabulary object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16901–16911 (2024)
16. DeepMind, G.: Gemini 2.5 Flash-Lite. https://deepmind.google (2025), accessed: 2026-06-30
17. DeepMind, G.: Gemini 3.1 Pro Preview. https://deepmind.google (2026), accessed: 2026-06-30
18. EasyEDA: EasyEDA: Online PCB design and circuit simulator. https://easyeda.com (2025), accessed: 2026-06-30
19. GitHub, Inc.: GitHub: Software development platform. https://github.com (2026), accessed: 2026-06-30
20. Huang, C.Y., Chen, H.I., Ho, H.W., Kang, P.H., Lin, M.P.H., Liu, W.H., Ren, H.: Netlistify: Transforming circuit schematics into netlists with deep learning. In: 2025

ACM/IEEE 7th Symposium on Machine Learning for CAD (MLCAD). pp. 1–8. IEEE (2025)
21. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO11. https://github.com/ultralytics/ultralytics (2024), accessed: 2026-06-30
22. Kendall, M.G.: A new measure of rank correlation. Biometrika **30**(1/2), 81–93 (1938)
23. Labcenter Electronics: Proteus Design Suite. https://www.labcenter.com (2025), accessed: 2026-06-30
24. Li, J., Chen, L., YANG, B., Zhu, J., Wang, Y., Ma, Y., Yang, M.: PCB-Bench: Benchmarking LLMs for printed circuit board placement and routing. In: The Fourteenth International Conference on Learning Representations
25. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. arXiv preprint arXiv:2304.08485 (2023)
26. Luo, Z., Song, X., Huang, H., Lian, J., Zhang, C., Jiang, J., Xie, X., Jin, H.: GraphInstruct: Empowering large language models with graph understanding and reasoning capability. arXiv preprint arXiv:2403.04483 (2024)
27. Ma, C., Jiang, Y., Wu, J., Yuan, Z., Qi, X.: Groma: Localized visual tokenization for grounding multimodal large language models. arXiv preprint arXiv:2404.13013 (2024)
28. Ma, C., et al.: When visual grounding meets gigapixel-level large-scale scenes: Benchmark and approach. In: CVPR (2024)
29. Mouselinos, S., Michalewski, H., Malinowski, M.: Beyond lines and circles: Unveiling the geometric reasoning gap in large language models. In: Findings of the Association for Computational Linguistics: EMNLP 2024. pp. 6192–6222 (2024)
30. National Instruments: NI Multisim. https://www.multisim.com/ (2025), accessed: 2026-06-30
31. OpenAI: GPT-5.2. https://openai.com (2025), openAI model release; Accessed: 2026-06-30
32. Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., Ma, S., Wei, F.: Kosmos-2: Grounding multimodal large language models to the world. arXiv preprint arXiv:2306.14824 (2023)
33. ProtoCentral: ProtoCentral: Open source medical electronics. https://protocentral.com (2025), accessed: 2026-06-30
34. Rasheed, H., et al.: GLaMM: Pixel grounding large multimodal model. arXiv preprint arXiv:2311.03356 (2024)
35. Seeed Technology Co., Ltd.: Seeed Studio. https://www.seeedstudio.com (2025), accessed: 2026-06-30
36. Shi, Y., Tao, Z., Gao, Y., Huang, L., Wang, H., Yu, Z., Lin, T.J., He, L.: AMSNet 2.0: A large AMS database with AI segmentation for net detection (2025), https://arxiv.org/abs/2505.09155, accessed: 2026-06-30
37. Shi, Y., Tao, Z., Gao, Y., Huang, L., Wang, H., Yu, Z., Lin, T.J., He, L.: AMSNet 2.0: A large AMS database with AI segmentation for net detection. In: 2025 IEEE International Conference on LLM-Aided Design (ICLAD). pp. 242–248. IEEE (2025)
38. SparkFun Electronics: SparkFun Electronics. https://www.sparkfun.com (2025), accessed: 2026-06-30
39. Tao, Z., Shi, Y., Huo, Y., Ye, R., Li, Z., Huang, L., Wu, C., Bai, N., Yu, Z., Lin, T.J., et al.: AMSNet: Netlist dataset for AMS circuits. In: 2024 IEEE LLM Aided Design Workshop (LAD). pp. 1–5. IEEE (2024)
40. Team, P.: PaddleOCR 3.0 technical report. arXiv preprint arXiv:2409.01704 (2024), https://arxiv.org/abs/2409.01704, accessed: 2026-06-30

41. Thoma, F., Bayer, J., Li, Y., Dengel, A.: A public ground-truth dataset for handwritten circuit diagram images. In: International Conference on Document Analysis and Recognition. pp. 20–27. Springer (2021)
42. Wang, Y., Lu, T., Liu, R., Yang, L., Yang, Y., Chen, Z., Wang, Y., Liu, Y., Lin, K., Chen, X., et al.: A large language model powered integrated circuit footprint geometry understanding. arXiv preprint arXiv:2508.03725 (2025)
43. Wu, P., Xie, S.: V*: Guided visual search as a core mechanism in multimodal LLMs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13084–13094 (2024)
44. Xu, H., Liu, C., Wang, Q., Huang, W., Xu, Y., Chen, W., Peng, A., Li, Z., Li, B., Qi, L., et al.: Image2Net: Datasets, benchmark and hybrid framework to convert analog circuit diagrams into netlists. In: 2025 International Symposium of Electronics Design Automation (ISEDA). pp. 807–816. IEEE (2025)
45. Yang, Z., Liu, Z., Wang, X., Wang, Z., Yu, Y., Yang, Z., et al.: MM-ReAct: Prompting ChatGPT for multimodal reasoning and action. arXiv preprint arXiv:2303.11381 (2023)
46. You, H., Zhang, H., Gan, Z., Du, X., Zhang, B., Wang, Z., Cao, L., Chang, S.F., Yang, Y.: Ferret: Refer and ground anything anywhere at any granularity. arXiv preprint arXiv:2310.07704 (2023)
47. Zeng, Y., et al.: Investigating compositional challenges in vision-language models for visual grounding. In: CVPR (2024)
48. Zhao, T., et al.: RGBT-Ground benchmark: Visual grounding beyond RGB in complex real-world scenarios. arXiv preprint arXiv:2512.24561 (2025)
49. Zheng, Z., Yang, M., Hong, J., Zhao, C., Xu, G., Yang, L., Shen, C., Yu, X.: DeepEyes: Incentivizing "thinking with images" via reinforcement learning. arXiv preprint arXiv:2505.14362 (2025)