ICLAgent: Integrated Circuit Footprint Geometry Labeling via LMM-empowered Multi-Agent Framework

Yida Wang[1*], Yixin Liu[1*], Taiting Lu[2], Lanqing Yang[1✉], Dian Ding[1], Juntao Zhou[1], Yifan Yang[3✉], Yi-Chao Chen[1], and Mahanth Gowda[2]

[1] Shanghai Jiao Tong University, Shanghai, China
[2] Pennsylvania State University, Pennsylvania, USA
[3] Microsoft, Shanghai, China
yanglanqing@sjtu.edu.cn, yifanyang@microsoft.com

Abstract. Integrated circuit (IC) footprint geometry labeling refers to the process of converting pin diagrams in IC datasheets into machine-readable geometric parameters. This task is critical in printed circuit board (PCB) design and component assembly, as accurate labeling ensures proper IC placement and reliable connectivity. The process is challenged by unstructured annotations, complex footprint arrangements, and abstract geometric diagrams, making fully automated labeling methods inadequate. Traditional EDA tools require heavy manual input. Existing automation methods, such as OCR or object detection, fail to capture the implicit geometric relationships in IC diagrams, leaving the labeling task incomplete. Recent work has shown that end-to-end large multimodal models (LMMs) can perform IC geometry labeling. However, by treating the task as a black box, it is prone to shortcut learning and lack interpretability. In this work, we introduce ICLAgent, the first multi-agent framework for fully automated IC footprint geometry labeling that explicitly models the workflow of expert engineers to produce more interpretable and reliable labeling outcomes. Furthermore, we present ICAgent-Instruct, the first dynamic planning and reasoning dataset tailored for IC footprint geometry labeling. Extensive experiments show that ICLAgent improves overall accuracy by 10.3% compared to the previous SOTA method and by 79.5% compared to manual annotation. Despite using only simple supervised fine-tuning on a 7B model (Qwen2-VL-7B), ICLAgent surpasses general-purpose LMMs such as GPT-5 (by 94.6%) and Gemini-3 Pro (by 15.2%).

Keywords: Multi-Agent Framework · IC Footprint Geometry Labeling

1 Introduction

Integrated circuits (ICs) are miniaturized electronic systems with many interconnected components. IC pins are represented in footprint diagrams, which

*Equal contribution.

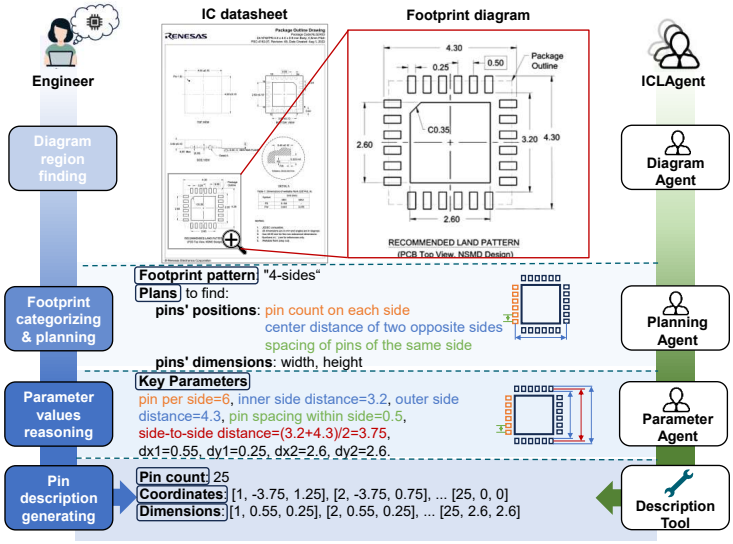

Fig. 1: ICLAgent simulates the human expert engineer's reasoning workflow. Typical workflow for a PCB engineer to label IC footprint pin geometry information involves four sequential steps: **(1) searching for the diagram** in the datasheet, **(2) identifying the footprint type** (*e.g.*, whether the IC has pins on 2 sides, 4 sides, or arranged in a grid) and **planning for the key parameters** to be extracted, **(3) searching for and calculating the key parameter values,** and **(4) manually entering the extracted key parameters into EDA tools** to generate footprint drawings with geometric information. ICLAgent mimics the above workflow and replaces each key step with a task-specific LMM agent, aiming at fully automated IC footprint labeling.

are often referred to as "Suggest Pads" or "Land Patterns" in IC datasheets (as shown in Fig. 1 top). In printed circuit board (PCB) design, engineers must precisely label IC pin footprints (*i.e.*, identifying pin counts, coordinates, and dimensions), since this ensures correct component placement and reliable electrical connections. Many component vendors, such as Texas Instruments [42] and CUI [6], provide footprint description diagrams within their IC datasheets. In addition, third-party libraries like Digi-Key [13] and SnapEDA [38] offer supplementary footprint resources. Despite the abundance of available IC datasheets and footprint diagrams (*e.g.*, over 13 million electronic products in Digi-Key), there remains a lack of efficient tools capable of converting these abstract geometric diagrams into machine-readable numerical descriptions. This gap slows down PCB design and manufacturing, as engineers must manually interpret diagrams, a process that is time-consuming, error-prone, and detrimental to design cycle speed and reliability [3].

Traditional electronic design automation (EDA) tools require expert operation and lack automation for interpreting datasheet diagrams. As a result, they are inefficient and difficult to use for large-scale or rapid PCB design. Although

deep-learning methods such as OCR [11, 12] and object detection [24, 40] are widely used for PDF text extraction and content understanding, they primarily focus on text comprehension and image description tasks. Consequently, they are inadequate for handling the complex text-symbol geometric understanding required for footprint diagrams, especially when implicit information is omitted. Large multimodal models (LMMs) advance this capability by deep reasoning over the text-image relationships and have demonstrated remarkable capabilities across a range of domains, including general-purpose reasoning [17, 22, 46], geometric and mathematical reasoning [15, 26, 29], image understanding [18, 45], and medical diagnosis [23]. Building on this progress, LLM4-IC8K [47] introduces a two-stage training framework that fine-tunes an LMM for automated IC footprint geometry labeling. However, by treating the task as a black box, these end-to-end methods neglect the intermediate reasoning steps that PCB engineers naturally follow, such as locating the diagram, identifying IC types, and extracting parameters (as shown in Fig. 1). These steps are essential for ensuring interpretability and reliability. In contrast, our multi-agent design explicitly mirrors this human workflow through diagram detection, type classification, parameter extraction, and description generation.

Recent advancements in agent-based systems [8,20,34,51] have demonstrated impressive capabilities across a wide range of domains, including strategic gameplay [48], complex task automation [36], multimodal content generation [54], medical diagnosis [39, 49], and embodied agent interaction [37]. Agent-based LMM systems enable dynamic task decomposition and strategic planning, making them particularly well-suited for domains that involve expert knowledge, complex reasoning workflows, and multi-step decision-making pipelines. However, no prior work has applied such frameworks to IC geometry labeling, leaving a gap that our work addresses.

Motivated by the natural reasoning process employed by PCB engineers during IC footprint labeling tasks, we propose **ICLAgent, a novel framework that enables step-by-step dynamic reasoning, explicitly simulating the expert workflow used by real-world PCB engineers**. ICLAgent employs multiple task-specific agents to construct a multi-stage workflow for generating pin description labels of IC footprints, as illustrated in Fig. 1. Our main contribution in advancing agent-based IC footprint labeling is fourfold:

1. **We introduce ICLAgent, the first multi-agent framework that emulates the IC footprint understanding and labeling workflow of expert PCB engineers** by incorporating IC footprint classification, dynamic planning, and key parameter extraction through multi-step reasoning.
2. **We develop ICAgent-Instruct, the first dynamic planning and reasoning dataset specifically designed for IC footprint geometry labeling**. This dataset is tailored to fine-tune LMMs with planning strategies and geometric understanding reflective of expert-level PCB engineering knowledge. Codes and datasets are available in `https://github.com/IC-LMM/ICLAgent`.

3. **We propose a multi-stage training strategy that sequentially fine-tunes three task-specific LMM agents**, each specializing in a distinct subtask: footprint diagram localization, footprint classification and parameter planning, and key parameter extraction.
4. **We evaluate our framework extensively on the IC footprint geometry labeling benchmark and each workflow stage**. The results demonstrate our approach's effectiveness in providing interpretable and practical reasoning for IC footprint labeling. Our method uses only SFT training with the lightweight base model Qwen2-VL-7B [46]. In contrast, state-of-the-art general LMMs, such as GPT-5 [32], fail to complete the labeling task.

2 Related Work

2.1 LMM Agents

LMM agents are intelligent systems powered by LMMs, capable of perceiving the environment, reasoning about goals, and executing actions. These capabilities enable them to autonomously perform complex tasks traditionally requiring human expertise, while adapting to task-specific requirements [27]. Recent advances in LMM agents have demonstrated notable success in simulating domain-expert workflows across a range of applications, including strategic gameplay [48], software development [35, 54], creative content generation [52], embodied interaction [37], and clinical diagnosis [39, 49]. LMM agents have also shown growing potential in the PCB engineering industry. Most existing applications focus on EDA, encompassing domain-specific assistant chatbots [9, 19, 25], hardware description language (HDL) generation [7, 50], and code verification and analysis [33]. These efforts primarily address the macroscopic aspects of PCB design, often neglecting the foundational components, such as detailed IC footprint information. These footprints are typically embedded within lengthy datasheet PDFs in abstract geometric diagram form, which still require significant manual effort to convert into structured numeric descriptions. We introduce the first LMM agent system devoted to addressing the IC footprint labeling problem, laying the foundation for automating various downstream EDA tasks.

2.2 Automated IC Footprint Labeling

Traditional PCB component geometry generation is a labor-intensive and time-consuming process that involves manually interpreting datasheets, creating footprints, generating symbols, and mapping signals [14,30]. As modern designs often involve hundreds of components, this manual approach becomes a significant bottleneck [28], susceptible to inconsistencies, human error, and outdated libraries due to frequent specification updates [1, 41]. These challenges highlight the critical need for automated, data-driven solutions to enhance both the efficiency and reliability of PCB design workflows. Existing PCB labeling methods primarily focus on the segmentation or classification of IC footprints [30, 53]. These approaches fail to address the geometric information of individual IC pins, limiting

the automated labeling process to a higher level, overlooking the intricate details of pin-level geometry. Although object detection methods [24, 40] can assist in counting IC pins and estimating their relative sizes, they struggle with footprint diagrams that omit implicit information. Optical Character Recognition (OCR) methods [11, 12] can extract textual data from datasheet images, but cannot interpret the physical or geometric meaning of numerical labels. Some works combine object detection and OCR for diagram object detection [21] or for converting legacy schematic diagrams [31]. However, these methods do not address the gap between the extracted annotations and the underlying geometric knowledge. LLM4-IC8K [47] introduces a two-stage training framework that fine-tunes an LMM for automated IC footprint geometry understanding and labeling. However, this end-to-end approach treats the IC footprint labeling task as a black-box problem, neglecting the intermediate reasoning processes employed by expert PCB engineers. In contrast, our framework introduces step-by-step dynamic reasoning, explicitly simulating the workflow of real-world PCB engineers, which leads to more interpretable and reliable footprint labeling performance.

3 IC Geometry Understanding Problem

3.1 IC Footprint Labeling Workflow

The problem of understanding IC geometry labeling can be delineated into three primary components: (1) determining the number of pins in the IC diagram, (2) analyzing the spatial arrangement (i.e., coordinates) of pins, and (3) discerning the shape and dimensions of each individual pin. While experienced engineers may be able to estimate the number of pins in low-pin-count diagrams through intuition, the precise coordinates and dimensions of the pins cannot be ascertained instantaneously. Furthermore, IC modules often comprise over a hundred pins, rendering manual and immediate pin counting impractical. Therefore, engineers typically adopt a step-by-step approach to deconstruct the diagrams and extract key parameters that facilitate accurate identification of IC footprints.

When identifying an IC footprint, an engineer typically first determines the number of pins depicted in the diagram. This is achieved by analyzing the geometric structure of the diagram, such as counting the number of rows and columns in a uniform pin array or the number of pins per edge in a centrosymmetric ring configuration. Subsequently, the engineer interprets the spatial relationships among the pins based on the geometric annotations provided in the diagram, including row spacing, column spacing, and distances from the module center to the edges. The engineer then assesses the dimensions of individual pins using the labeled measurements, such as height and width for rectangular pins or diameter for circular pins. Finally, by summarizing this geometric information, the engineer determines the pin count, pin coordinates, and pin dimensions of the IC footprint. We propose a multi-agent framework designed to emulate the step-by-step reasoning process of an expert engineer, thereby enhancing the geometric understanding of LMMs when interpreting IC footprint diagrams. We utilize domain priors as a reasoning scaffold to resolve the inherent ambiguity

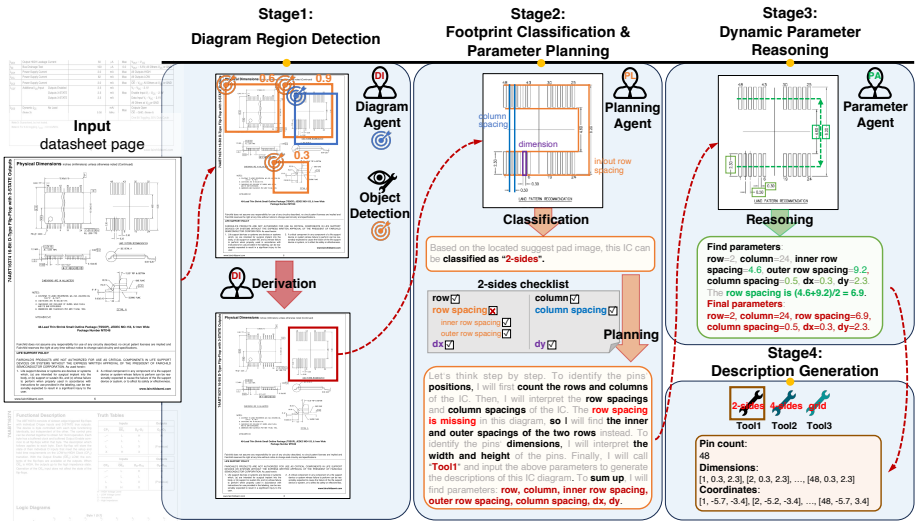

Fig. 2: Overview of ICLAgent. The reasoning pipeline includes four sequential stages. First, the **Diagram Agent** detects and locates the region of the footprint diagram in the datasheet page, and sends the region information to the Planning Agent. The **Planning Agent** then analyzes the arrangement pattern of the IC and generates instructions to guide the key parameter extraction process. After receiving instructions from the Planning Agent, the **Parameter Agent** finds accessible key parameters in the diagram and calculates implicit key parameters based on given instructions. Finally, ICLAgent calls pin **description generation** tools based on the IC type classification to generate a detailed geometric description for each IC pin.

in technical schematics. This hierarchical decomposition enables the model to adaptively interpret diverse footprint layouts that lack uniform rules.

3.2 Challenges

A primary challenge for LMMs in interpreting IC footprint diagrams lies in accurately identifying the key parameters presented in the diagrams and comprehending the relationships between numeric labels and graphical symbols. Often, these critical parameters are not explicitly provided and require additional reasoning to infer. For instance, as illustrated in Fig. 2, the center-to-center distance (green lines in the right-most diagram) between two pin rows is not directly labeled. Instead, it must be derived from the relationships among the outer-edge distance, the inner-edge distance of the pins, and their spatial arrangement (orange lines in the middle diagram). The agent must also be capable of adapting its reasoning workflow to accommodate different types of IC modules. Due to the variability in the spatial arrangement of pins across module types, the key parameters that need to be extracted, as well as the tools required for generating accurate descriptions of these parameters, differ accordingly. Therefore, it is essential to design reasoning workflows that account for the diverse types of IC pin

arrangements and support multi-step parameter inference to enable human-like reasoning capabilities in LMMs for IC footprint geometry understanding.

4 Methods

4.1 Reasoning Workflow

As described in Sec. 3.1, human engineers typically follow a four-step workflow to label IC footprints. To replicate this process, we design ICLAgent as a multi-agent system. Each agent corresponds to one of these steps, ensuring that the overall workflow mirrors the reasoning process of real engineers while remaining interpretable: (1) identifying the region containing the target IC footprint diagram, (2) recognizing the footprint's arrangement type and determining the key parameters required to infer the pin coordinates and dimensions, (3) locating and computing these key parameters based on the numeric annotations provided in the diagram, and (4) generating a comprehensive description of the pins using the extracted parameters. To emulate expert-level understanding of IC footprint geometry labeling, we develop a multi-agent system capable of dynamic diagram region detection, classification of IC pin arrangement types, strategic planning for key parameter extraction, and geometric reasoning over the identified parameters. As illustrated in Fig. 2, the IC footprint geometry labeling workflow can be arranged into four key stages with three task-specific agents:

1) Diagram Region Detection: When given a PDF page of an IC datasheet containing the IC footprint diagram, ICLAgent adopts the **Diagram Agent** to identify the region of the target diagram described in a four-element number pair (x, y, dx, dy), where x, y denote the coordinates of the upper left corner of the diagram region, and dx, dy denote the width and height of the region in proportions of the datasheet image sizes. The Diagram Agent first invokes an object-detection tool (YOLO v12 [43]) to identify candidate regions of interest (ROIs) corresponding to the footprint diagrams, each with an associated confidence score (orange boxes in Stage 1 in Fig. 2). It then cross-validates these candidates using its own diagram-detection judgments (blue box) and fuses the results to yield the final detection region (red box). The detected diagram region enables downstream agents to focus on the relevant diagram, minimizing interference from surrounding, unrelated diagrams and text.

2) Footprint Classification and Parameter Planning: After identifying the diagram region, ICLAgent deploys the **Planning Agent**, which processes region-specific prompts alongside the diagram image. The Planning Agent classifies the IC footprint type based on the spatial arrangement of its pins. To ensure representativeness while maintaining generality, we categorize common IC footprints into four classes based on their pin arrangement patterns [4], as illustrated in Fig. 3: 2-sides, 4-sides, grid, and other. The first three categories account for over 80% of all IC footprints and collectively represent the majority of standard IC footprint templates. Next, the Planning Agent identifies key parameters necessary to infer pin coordinates and dimensions based on the previously determined arrangement type. It follows an IC-type-specific checklist derived from the

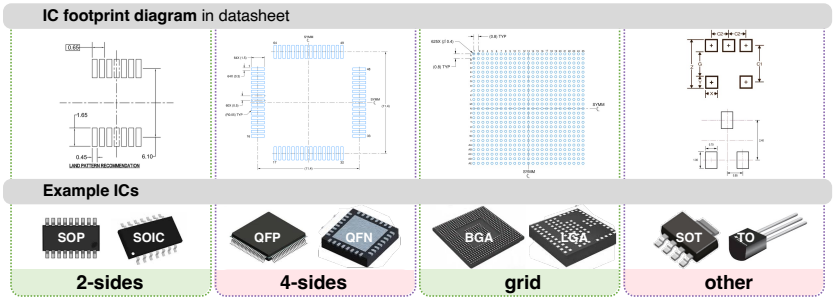

Fig. 3: We categorize IC packages into four groups based on pin distribution: **dual-side packages** (*e.g.*, SOP, SOIC), **quad-side packages** (*e.g.*, QFP, QFN), **grid array packages** (*e.g.*, BGA, LGA), and **others** (*e.g.*, SOT, TO). For simplicity, we refer to them as "2-sides", "4-sides", "grid", and "other" throughout this paper. "2-sides", "4-sides", and "grid" footprints are symmetric and can be structurally described by several key parameters, while "other" footprints are asymmetric but often contain fewer pins.

classification result, which includes all key parameters that may appear in the IC diagram of the specified type. The agent evaluates both existing parameters and those missing but essential for accurate footprint identification. Specifically, the Planning Agent first identifies the key parameters explicitly annotated in the diagram. For parameters not directly provided, the agent performs parameter reasoning to determine how these values can be derived from the available annotations. Finally, the agent compiles a summary of all relevant key parameters, which guides the downstream parameter extraction process. IC footprints categorized as "other" typically exhibit irregular pin arrangements and usually contain fewer than 20 pins. As a result, identifying pin coordinates and dimensions is less complex. Therefore, we adopt a direct approach to describe the pin count, coordinates, and dimensions for footprints in this category.

3) Dynamic Parameter Reasoning: Under the guidance of the Planning Agent, the **Parameter Agent** identifies and interprets the key parameters expressed numerically within the diagram. For parameters that are not explicitly annotated, the Parameter Agent applies the reasoning instructions specified by the Planning Agent to compute the required values. Finally, it compiles the extracted and inferred parameters into a type-specific standardized format.

4) Description Generation: Based on the identified pin arrangement type, ICLAgent invokes the appropriate description generation tool corresponding to "2-sides", "4-sides", or "grid" footprints. These tools are Python scripts that take the standardized key parameters as input and produce pin descriptions aligned with the three IC footprint labeling tasks outlined in Sec. 3.1: determining the pin count, computing the pin coordinates, and specifying the pin dimensions. For "other" footprints, the pin descriptions are directly inferred through the collaborative processing of the Planning Agent and the Parameter Agent and require no description generation tool.

4.2 Dataset Construction

To effectively train our agent-based model to mimic the complex reasoning workflow of IC footprint understanding, we construct the ICAgent-Instruct dataset. It contains real datasheet diagrams with expert annotations and contains three subsets aligned with our agents' subtasks: (1) diagram detection, (2) parameter planning, and (3) parameter extraction. Each subset reflects the reasoning steps that engineers use, ensuring that **training data encodes expert-level strategies rather than only end results**.

Our work builds on the real-world datasheets from the ICGeo8K dataset introduced by [47], the only existing IC footprint labeling dataset based on real-world sources. It contains 4,138 IC footprint entries spanning over 2,000 manufacturers and 10 package types (*e.g.*, QFN, BGA, SOIC). To ensure data quality, we filter out diagrams that are mislabeled, occluded, or incorrectly classified as 2D footprint diagrams. After this cleaning process, we obtain a total of 3,737 valid sequences, comprising 3,337 training samples from the original real-world training subset and all 400 test samples (excluded during the fine-tuning process) from ICGeo8K. The datasets are constructed using the datasheet page images and their corresponding pin descriptions. The data generation process strictly adheres to the reasoning workflow outlined in Sec. 4.1. In the end, we construct three novel datasets based on the real-world samples: the **Diagram Detection Dataset**, the **Parameter Planning Dataset**, and the **Parameter Extraction Dataset**.

1) Diagram Detection Dataset: We begin by identifying the regions containing footprint diagrams within the datasheet pages. To facilitate this process, we use Label Studio [44] to annotate the bounding boxes corresponding to the footprint diagram regions. Each footprint diagram bounding box is manually annotated by (1) its upper-left corner coordinates and (2) its width and height. This annotated dataset is used to train the Diagram Agent to identify the IC footprint diagram regions accurately.

2) Parameter Planning Dataset: The core reasoning capability of ICLAgent lies in the workflow planning conducted by the Planning Agent. To train the LMM to perform strategic parameter understanding in a human-expert manner, we propose a four-step dataset construction procedure. First, we manually annotate the arrangement types of the identified footprint diagrams. Next, based on the classified type, annotators are presented with a set of candidate key parameters and are instructed to fill in only those parameters that are explicitly provided within the footprint diagram. Then, we identify whether any standard key parameters are missing from the diagram (*e.g.*, a missing center distance such as in Fig. 2) but can be computed using the available labeled parameters (*e.g.*, by averaging the outer-edge and inner-edge distances). This reasoning process is then added to the planning workflow. If the provided parameters are insufficient to derive the standard key parameters, the model is instructed to examine the surrounding context within the datasheet page to infer the missing information. Finally, the Planning Agent is instructed to summarize all explicit parameters present in the diagram that are involved in the planning process, pro-

viding a structured input for the downstream Parameter Agent. The aforementioned planning steps, combined with the corresponding datasheet images and annotated diagram bounding boxes, constitute the Parameter Planning Dataset.

3) Parameter Extraction Dataset: The workflow of the Parameter Agent is guided by the output parameter sets generated by the Planning Agent. For each data sample, the Parameter Agent is tasked with extracting the standard key parameters corresponding to the identified footprint arrangement type. The outputs from the Planning Agent are first examined to identify any missing standard key parameters. If standard key parameters are missing, the Parameter Agent initiates a reasoning step to compute them using the available non-standard parameters. Finally, the Parameter Agent consolidates the results into a complete set of standard key parameters. For IC footprints classified as "other", we utilize the data labels from ICGeo8K to extract their pin count, pin coordinates, and pin dimensions. ICGeo8K data labels are also employed to identify missing pins in grid-arranged footprints with large pin counts.

In summary, we construct three novel datasets derived from the real-world ICGeo8K dataset, each designed to capture different aspects of the dynamic reasoning process involved in IC footprint geometry labeling.

4.3 Model Training

To implement an effective multi-agent framework, we adopt a three-step training strategy aimed at developing specialized functional agents. At the core of each agent lies a powerful vision-language model, Qwen2-VL-7B [46]. We fine-tune this LMM on the three task-specific reasoning datasets described in Sec. 4.2, resulting in three dedicated models tailored to support the corresponding workflow agents.

We begin by training the Diagram Agent using the Diagram Detection Dataset to enable accurate localization of IC footprint regions. We also train an object detection tool using the same Diagram Detection Dataset to acquire IC ROI candidates. Next, the Planning Agent is trained on the Parameter Planning Dataset to develop its ability to classify footprint types and plan procedures to identify key parameters. Finally, the Parameter Agent is trained using the Parameter Extraction Dataset, allowing it to extract and compute standard key parameters based on the Planning Agent outputs. At each training stage, we incorporate chain-of-thought (CoT) training to promote step-by-step reasoning and ensure the models develop a robust and interpretable understanding of the tasks. The 7B-parameter multi-agent system was fine-tuned in only 5 hours using 2 NVIDIA A100-40GB GPUs. Regarding inference latency, the framework achieves an average processing speed of 13.5 seconds per datasheet page.

5 Experiments

5.1 Experiment Setup

Evaluation Metrics. We adopt the same evaluation metrics as LLM4-IC8K which evaluate a model's ability to faithfully translate geometric relationships

Table 1: Comparison of QA performance with general LMMs on 3 tasks in mean + std format. The best-performing method is **in-bold**, and the second-best is underlined.

Methods	Overall (IoU_{IC} %) ↑	Task 1		Task 2 (d_{pin})↓	Task 3 (IoU_{pin} %)↑
		MAE↓	RMSE↓		
General Large Multimodal Models					
Gemini 3 Pro [16]	68.6 ± 0.6	0.44 ± 0.08	3.46 ± 0.41	3.56 ± 0.20	82.5 ± 0.3
Gemini 2.5 Pro [10]	55.7 ± 0.7	0.57 ± 0.13	4.78 ± 0.42	5.58 ± 0.48	70.1 ± 0.3
GPT-5 [32]	40.6 ± 1.1	0.59 ± 0.14	4.13 ± 0.19	4.18 ± 0.22	65.4 ± 0.1
GPT-4o [22]	11.1 ± 0.4	8.21 ± 0.47	23.04 ± 0.22	4.01 ± 0.02	45.6 ± 0.3
Claude Sonnet 4 [2]	10.9 ± 0.7	2.03 ± 0.44	12.55 ± 0.96	3.62 ± 0.23	12.4 ± 0.4
Qwen3-VL-235B [5]	9.8 ± 0.2	0.46 ± 0.17	3.79 ± 0.97	5.58 ± 0.17	44.3 ± 0.2
Manual Industry Baseline					
Manual EDA	44	-	-	2.98	58
IC Footprint Labeling Large Multimodal Models					
LLM4-IC8K [47]	71.6±0.5	**0.35±0.07**	**2.81±0.08**	1.11±0.02	88.0±0.3
ICLAgent	**79.0±0.2**	0.37±0.05	3.25±0.14	**1.07±0.03**	**95.6±0.2**

from technical schematics: IoU_{IC} measures the Intersection over Union (IoU) ratio between predicted and ground-truth footprints; Mean Absolute Error (MAE) and Root Mean Square Error ($RMSE$) measure pin count accuracy (*task 1*); d_{pin} measures coordinate precision (*task 2*); IoU_{pin} measures pin dimension accuracy (*task 3*). For intermediate agents, we report the following metrics. For **Stage 1 (diagram detection)**, $IoU_{diagram}$ measures the IoU between predicted and ground-truth diagram bounding boxes. For **Stage 2 (classification and planning)**, Type Classification Accuracy (TCA) measures the footprint type classification accuracy, and Key Parameter Accuracy (KPA) measures whether the Planning Agent identifies the correct set of key parameters. For **Stage 3 (parameter reasoning)**, Parameter Completion rate (PC) measures the recovery rate of required parameters, while MAE and $RMSE$ capture the numerical accuracy of predicted values.

Evaluation Baseline. We evaluate our framework on ICGeoQA [47], a benchmark consisting of 400 representative real-world IC datasheet pages, and compare its performance against the current state-of-the-art method for IC footprint labeling, LLM4-IC8K [47]. For a fair comparison, all baselines were trained and tested under the same hardware settings. **General-purpose LMMs** were evaluated in few-shot settings (1 example case per footprint type, 4 in total) for compatible comparisons and consistent output formats, while the **specialized model** (*i.e.*, LLM4-IC8K) used its reported fine-tuning pipeline. All models receive complete PDF images as input. Metrics were derived via pattern-matching scripts, and invalid or non-numeric responses were treated as zero-score failures to ensure a rigorous assessment. To define the boundary of engineering viability, we establish a **manual industry baseline** by measuring the deviation of production-ready EDA designs from their source diagrams.

5.2 Overall Comparison

We evaluate the overall performance of our framework and compare it against both the baseline method and SOTA general-purpose LMMs (Tab. 1). All mod-

Table 2: Comparing performances on three stages with general-purpose LMMs.

Methods	Stage1	Stage2		Stage3		
	$IoU_{diagram}$ %↑	TCA %↑	KPA %↑	PC %↑	MAE↓	RMSE↓
GPT-5 [32]	2.9 ± 0.6	79.5 ± 0.2	93.1 ± 0.5	79.8 ± 0.4	1.94 ± 0.06	4.18 ± 0.38
Gemini 2.5 Pro [10]	3.9 ± 0.7	84.6 ± 0.2	86.6 ± 0.4	80.4 ± 0.1	1.50 ± 0.04	3.55 ± 0.17
Qwen2.5-VL-237B [5]	3.0 ± 0.3	28.3 ± 0.3	24.1 ± 0.4	81.6 ± 0.5	2.21 ± 0.28	10.13 ± 0.15
ICLAgent	**70.2±0.4**	**100**	**97±0.1**	**82.3±0.4**	**0.22±0.03**	**0.39±0.06**

Table 3: Ablation test on reasoning stages.

Methods	IoU_{IC}%↑
Full stages (ICLAgent)	**79.0±0.2**
without stage1	72.3±0.3
without stage2	70.2±0.2
without stage1&stage2	68.2±0.4
end-to-end	65.7±0.1

els process the complete PDF page images to ensure a standardized comparison across all methods. ICLAgent achieves an overall IoU_{IC} of 79.0% on the ICGeoQA benchmark, substantially outperforming all SOTA general-purpose LMMs, proving that specialized multi-agent architectures and expert datasets are more effective for encoding intricate IC domain knowledge than raw model scaling. The manual benchmark demonstrates that while engineers often introduce subjective variations, a 44% IoU_{IC} alignment remains industrially usable. ICLAgent's higher score indicates a more precise interpretation of IC footprint diagrams, fulfilling industrial standards. While ICLAgent shows sub-optimal performance in IC pin counting compared to LLM4-IC8K, the differences are negligible, with an average error of less than 1 pin, and it still outperforms all general LMMs. Additionally, ICLAgent demonstrates superior capability in pin coordinate and dimension estimation, enhancing the overall geometric understanding accuracy by 10.3% compared to the SOTA task-specific LMM and surpassing the industry manual baseline by 79.5%. This is due to the simplicity and stability of the framework, which focuses on handling concise key parameters rather than directly estimating the complex and lengthy pin geometry descriptions. ICLAgent encompasses expert-level knowledge for IC footprint identification and geometric reasoning, while emulating the step-by-step reasoning workflow of PCB engineers to support a dynamic labeling process. By adhering closely to the procedural logic employed by human experts, ICLAgent mitigates the risks of hallucinations and shortcut learning commonly observed in end-to-end fine-tuned LMMs. As a result, **ICLAgent achieves superior performance in IC footprint geometry understanding and labeling.**

5.3 Component Evaluation

We evaluate the performance of each task-specific agent within the pipeline (Tab. 2). For Stage 1, all models process the complete PDF page images. To isolate the performance of Stage 2 and Stage 3, we compare all methods using both the full PDF pages and the ground-truth diagram regions.

Evaluation on Diagram Region Detection (Stage 1). We evaluate the performance of the Diagram Agent on the ICGeoQA benchmark and compare it with general-purpose LMMs. Our Diagram Agent achieves an $IoU_{diagram}$ of 70.2% in accurately locating diagram regions within datasheet pages, significantly outperforming the baseline models. General LMMs face challenges in accurately locating the correct diagram regions without specialized training or

extensive prior knowledge. This result highlights the importance of training a dedicated agent for diagram region detection in IC footprint understanding.

Evaluation on Footprint Classification and Parameter Planning (Stage 2). We evaluate the Planning Agent on footprint arrangement classification and parameter planning. The Planning Agent achieves a TCA of 100% and a KPA of 97%, demonstrating its effectiveness in accurately identifying footprint arrangement types and generating precise, footprint-specific plans for key parameter extraction. Despite their failure in diagram region detection, GPT-5 and Gemini 2.5 Pro demonstrate strong performance in identifying footprint types and recognizing parameters. This suggests that general LMMs typically process information at the whole-image level, while ICLAgent leverages zoomed-in views for more precise detail extraction.

Evaluation on Dynamic Parameter Reasoning (Stage 3). The Parameter Agent achieves a PC of 82.3% for key parameter recovery, with an MAE of 0.22 and an RMSE of 0.39 for parameter value recognition and reasoning. While the Parameter Agent does not recover all parameters specified in the instructions, its precise numeric value retrieval and parameter calculation process significantly reduces errors in the overall parameter extraction.

5.4 Stage Ablation

We conduct an ablation study to evaluate the contribution of each stage in our reasoning workflow, as presented in Tab. 3. In one setting, Stage 1 is removed, requiring downstream agents to locate footprint diagrams without explicit guidance. In another setting, Stage 2 is excluded, and the downstream agent is tasked with identifying and extracting key parameters independently. The third setting excludes both Stage 1 and Stage 2 and directly infers key parameter values from datasheet pages without structured guidance. Additionally, we compare with a fully end-to-end baseline, following the training strategy of LLM4-IC8K, which directly generates pin descriptions without explicitly reasoning through intermediate key parameters.

Removing **Stage 1** reduces IoU by 8.5%, showing that precise region localization is crucial. Removing **Stage 2** drops performance by 11.1%, confirming that explicit planning avoids confusion during extraction. With only **Stage 3**, IoU_{IC} falls by 13.7%. Interestingly, we observe that the performance of the **Stage-3-only** still exceeds that of the black-box **end-to-end LMM** approach. This is attributed to the simplicity and structure of the parameter extraction process, which enables the LMM to recover key geometric parameters more effectively than directly estimating the coordinates and dimensions of each pin. These results demonstrate that each stage contributes substantially, and that decomposed reasoning is more reliable than end-to-end black-box training.

5.5 Per-IC-type Performance Analysis

Tab. 4 details performance across IC types. As "other" ICs utilize an end-to-end pipeline, Stage 2 and 3 metrics are omitted. While ICLAgent is robust

Table 4: Per-IC-type Performance Analysis.

IC Type	Ratio (%)	Stage 1 ($IoU_{diagram}$% ↑)	Stage 2 (KPA % ↑)	Stage 3 (PC % ↑)	Overall (IoU_{IC} % ↑)	Task 1 (MAE ↓)	Task 2 (d_{pin} ↓)	Task 3 (IoU_{pin}% ↑)
2-sides	41.3	86	95.6	81.4	88.2	0.08	0.55	95.9
4-sides	31.7	58.8	98.3	80.4	73.2	0.43	0.38	95.6
grid	7.3	97.2	97.8	89.3	84.4	2.25	0.18	98.0
other	19.6	45.7	–	–	68.3	0.20	3.50	94.5

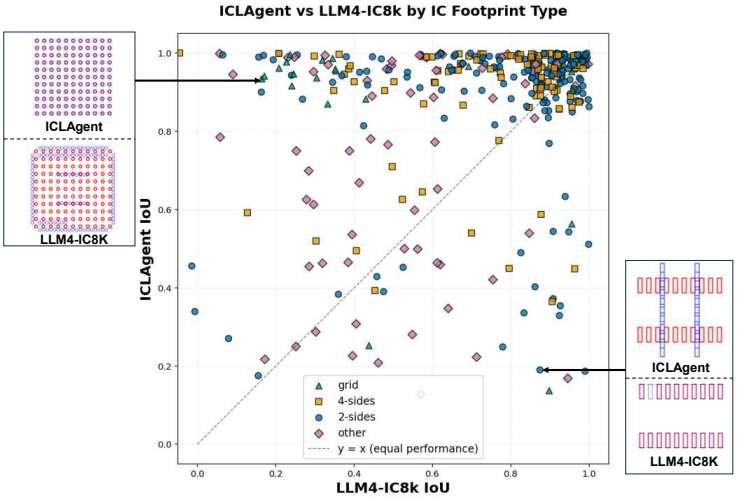

Fig. 4: Sample-level Comparison on ICLAgent vs. LLM4-IC8K.

on standardized footprints, "other" ICs show a performance dip, particularly in Stage 1 and Task 2, reflecting the challenge of irregular layouts and highlighting the necessity of the expert-guided task decomposition.

5.6 Sample-level Study and Failure Analysis

We conduct a sample-level qualitative evaluation comparing ICLAgent with the former specialized baseline LLM4-IC8K (Fig. 4). Most test samples sit above the equivalence line, showing ICLAgent's superiority. ICLAgent excels in high-pin-count, uniform layouts where LLM4-IC8K fails due to its reliance on isolated pin reasoning (upper-left case). Conversely, ICLAgent may introduce larger errors if key parameters are mispredicted or exchanged (lower-right case).

ICLAgent errors primarily categorize into: (1) Semantic misalignments, where the model confuses geometric properties of annotations, such as swapping dx and dy (*e.g.*, the failure shown in Fig. 4) or misidentifying dimension labels as positions; (2) Counting discrepancies, which often occur in dense or irregular layouts, as shown in Tab. 4 "grid" Task 1; (3) Localization inaccuracies, involving incomplete or incorrect detection of diagram regions, evidenced by lower Stage 1 $IoU_{diagram}$ scores for "4-sides" (58.8%) and "other" (45.7%) types in Tab. 4; and (4) Reasoning failures in "other" footprints due to the lack of stage-wise task decomposition, resulting in overall worse performance. To mitigate error propagation, we employ checklists to constrain LMM outputs and

inter-stage consistency checks. Given the industry-wide scarcity of IC-specific correction rules, expert-level fine-tuning that ensures high-fidelity agent outputs is essential to prevent error propagation fundamentally.

6 Limitation and Discussion

We recognize that Reinforcement Learning (RL) and feedback refinements (*e.g.*, ReAct [55]) hold considerable potential for further improving ICLAgent's performance. Furthermore, larger and more advanced base models could enhance the framework's overall effectiveness. Nevertheless, our framework demonstrates the great potential of geometric reasoning in IC footprint labeling by introducing an engineer-like thinking process.

Geometric diagram understanding is a fundamental challenge across mechanical and architectural engineering. We acknowledge the heavy reliance on hard-coded domain priors for complex IC footprint geometry understanding. While IC footprints feature domain-specific symbols, they share universal commonalities with other technical drawings, such as standardized dimensioning and annotation styles. Our human-interpretable reasoning workflow can be effectively extended to various technical domains by incorporating various domain-specific rule sets.

7 Conclusion

We introduced ICLAgent, a multi-agent framework for fully automated IC footprint geometry labeling. ICLAgent mimics the workflow of expert PCB engineers and comprises three task-specific agents for diagram region detection, type-specific parameter planning, and dynamic parameter reasoning. To support the dynamic reasoning capabilities of ICLAgent, we developed a novel dataset, ICAgent-Instruct, which encapsulates expert-level planning processes and reasoning strategies tailored for IC footprint labeling. We proposed a multi-stage training pipeline to fine-tune the LMM agents on their respective tasks. ICLAgent achieves a labeling IoU of 79.0%, surpassing the previous SOTA method by 10.3% and the manual baseline by 79.5%, proving its effectiveness.

Acknowledgements

We are grateful to all anonymous reviewers for their constructive comments. This work is supported in part by the United States National Science Foundation (NSF) (Grant No. CAREER-2046972), the National Natural Science Foundation of China (NSFC) (Grant No. 62502303), and the Natural Science Foundation of Shanghai (Grant No. 24ZR1439100, 25ZR1401171).

References

1. Abraham, R.: High-quality pcb libraries matter in pcb design (Mar 2025), https://blindburiedcircuits.com/why-high-quality-pcb-libraries-are-essential-for-effective-design/

2. Anthropic: Introducing claude 4 (2025), https://www.anthropic.com/news/claude-4, [Accessed 31-07-2025]
3. ASSEMBLY, A.: Inside the minds of pcb designers and buyers: The state of electronic anufacturing in the age of covid recovery (2021), https://f.hubspotusercontent20.net/hubfs/3932677/2021%20Engineer%20Survey%20-%20Combined%20-%20Final.pdf, [Accessed 31-07-2025]
4. Ayodele, A.: Ic package types: A comprehensive guide (2024), https://www.wevolver.com/article/ic-packages
5. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
6. bel: Cui inc (2025), https://www.belfuse.com/company/brands/cui-inc
7. Chang, K., Wang, Y., Ren, H., Wang, M., Liang, S., Han, Y., Li, H., Li, X.: Chipgpt: How far are we from natural language hardware design. arXiv preprint arXiv:2305.14019 (2023)
8. Chen, G., Dong, S., Shu, Y., Zhang, G., Sesay, J., Karlsson, B., Fu, J., Shi, Y.: Autoagents: a framework for automatic agent generation. In: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence. IJCAI '24 (2024). https://doi.org/10.24963/ijcai.2024/3, https://doi.org/10.24963/ijcai.2024/3
9. Chen, L., Chen, R., Hu, S., Yao, X., Tang, Z., Kai, S., Xu, S., Yuan, M., Hao, J., Yu, B., et al.: Pcbagent: An agent-based framework for high-density printed circuit board placement. In: Proceedings of the 30th Asia and South Pacific Design Automation Conference. pp. 781–787 (2025)
10. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
11. Du, Y., Chen, Z., Xie, H., Jia, C., Jiang, Y.G.: Svtrv2: Ctc beats encoder-decoder models in scene text recognition. arXiv preprint arXiv:2411.15858 (2024)
12. Duan, C., Jiang, Q., Fu, P., Chen, J., Li, S., Wang, Z., Guo, S., Luo, J.: Instructocr: Instruction boosting scene text spotting. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 2807–2815 (2025)
13. Electronics, D.: Digikey electronics - electronic components distributor. https://www.digikey.com (2025), accessed: 2025-05-14
14. Engineer: Things to learn before you learn pcb designing (Apr 2021), https://www.pcbway.com/blog/PCB_Basic_Information/Things_to_Learn_Before_You_Learn_PCB_Designing_1.html
15. Gao, J., Pi, R., Zhang, J., Ye, J., Zhong, W., Wang, Y., Hong, L., Han, J., Xu, H., Li, Z., Kong, L.: G-llava: Solving geometric problem with multi-modal large language model (2023), https://arxiv.org/abs/2312.11370
16. Google: A new era of intelligence with gemini 3 (2025), https://blog.google/products-and-platforms/products/gemini/gemini-3/, [Accessed 31-07-2025]
17. Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)
18. Guo, Z., Xu, R., Yao, Y., Cui, J., Ni, Z., Ge, C., Chua, T.S., Liu, Z., Huang, G.: Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. In: European Conference on Computer Vision. pp. 390–406. Springer (2024)
19. Han, B., Wang, X., Wang, Y., Yan, J., Tian, Y.: New interaction paradigm for complex eda software leveraging gpt. arXiv preprint arXiv:2307.14740 (2023)

20. Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S.K.S., Lin, Z., et al.: Metagpt: Meta programming for a multi-agent collaborative framework. In: The Twelfth International Conference on Learning Representations (2023)
21. Hu, X., Zhang, L., Liu, J., Fan, J., You, Y., Wu, Y.: Gptr: gestalt-perception transformer for diagram object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 899–907 (2023)
22. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
23. Ikezogwo, W., Seyfioglu, S., Ghezloo, F., Geva, D., Sheikh Mohammed, F., Anand, P.K., Krishna, R., Shapiro, L.: Quilt-1m: One million image-text pairs for histopathology. Advances in neural information processing systems **36**, 37995–38017 (2023)
24. Jocher, G.: Yolov5 by ultralytics (2020). https://doi.org/10.5281/zenodo.3908559, https://github.com/ultralytics/yolov5
25. Liu, M., Ene, T.D., Kirby, R., Cheng, C., Pinckney, N., Liang, R., Alben, J., Anand, H., Banerjee, S., Bayraktaroglu, I., et al.: Chipnemo: Domain-adapted llms for chip design. arXiv preprint arXiv:2311.00176 (2023)
26. Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.W., Galley, M., Gao, J.: Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. arXiv preprint arXiv:2310.02255 (2023)
27. Luo, J., Zhang, W., Yuan, Y., Zhao, Y., Yang, J., Gu, Y., Wu, B., Chen, B., Qiao, Z., Long, Q., et al.: Large language model agent: A survey on methodology, applications and challenges. arXiv preprint arXiv:2503.21460 (2025)
28. Martens, J.: Electronic part selection and library creation made simple (Aug 2022), https://blogs.sw.siemens.com/electronic-systems-design/2022/08/11/electronic-part-selection-and-library-creation-made-simple/
29. Masry, A., Kavehzadeh, P., Do, X.L., Hoque, E., Joty, S.: Unichart: A universal vision-language pretrained model for chart comprehension and reasoning. arXiv preprint arXiv:2305.14761 (2023)
30. Ni, Y.J., Wang, Y.J., Ho, T.Y.: Footprint classification of electric components on printed circuit boards. In: Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD. pp. 169–174 (2020)
31. Nurminen, J.K., Rainio, K., Numminen, J.P., Syrjänen, T., Paganus, N., Honkoila, K.: Object detection in design diagrams with machine learning. In: Progress in Computer Recognition Systems 11. pp. 27–36. Springer (2020)
32. OpenAI: Introducing gpt-5 (2025), https://openai.com/index/introducing-gpt-5/
33. Orenes-Vera, M., Manocha, A., Wentzlaff, D., Martonosi, M.: Autosva: Democratizing formal verification of rtl module interactions. In: 2021 58th ACM/IEEE Design Automation Conference (DAC). pp. 535–540. IEEE (2021)
34. Park, J.S., O'Brien, J., Cai, C.J., Morris, M.R., Liang, P., Bernstein, M.S.: Generative agents: Interactive simulacra of human behavior. In: Proceedings of the 36th annual acm symposium on user interface software and technology. pp. 1–22 (2023)
35. Qian, C., Liu, W., Liu, H., Chen, N., Dang, Y., Li, J., Yang, C., Chen, W., Su, Y., Cong, X., et al.: Chatdev: Communicative agents for software development. arXiv preprint arXiv:2307.07924 (2023)
36. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., Scialom, T.: Toolformer: Language models can teach

themselves to use tools. Advances in Neural Information Processing Systems **36**, 68539–68551 (2023)
37. Schumann, R., Zhu, W., Feng, W., Fu, T.J., Riezler, S., Wang, W.Y.: Velma: Verbalization embodiment of llm agents for vision and language navigation in street view. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 18924–18933 (2024)
38. SnapMagic Search: Snapmagic search - free pcb footprints and schematic symbols (2025), https://www.snapeda.com/
39. Sun, Y., Si, Y., Zhu, C., Zhang, K., Shui, Z., Ding, B., Lin, T., Yang, L.: Cpathagent: An agent-based foundation model for interpretable high-resolution pathology image analysis mimicking pathologists' diagnostic logic. arXiv preprint arXiv:2505.20510 (2025)
40. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781–10790 (2020)
41. Teel, J.: Pcb design: The top 5 mistakes made on printed circuit board layout (Sept 2023), https://predictabledesigns.com/pcb-design-top-5-mistakes-on-printed-circuit-boards-layout/
42. Texas Instruments: Texas instruments (2025), https://www.ti.com/
43. Tian, Y., Ye, Q., Doermann, D.: Yolov12: Attention-centric real-time object detectors. arXiv preprint arXiv:2502.12524 (2025)
44. Tkachenko, M., Malyuk, M., Holmanyuk, A., Liubimov, N.: Label Studio: Data labeling software (2020-2025), https://github.com/HumanSignal/label-studio, open source software available from https://github.com/HumanSignal/label-studio
45. Tzelepi, M., Mezaris, V.: Exploiting lmm-based knowledge for image classification tasks. In: International Conference on Engineering Applications of Neural Networks. pp. 166–177. Springer (2024)
46. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
47. 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)
48. Wang, Z., Cai, S., Chen, G., Liu, A., Ma, X., Liang, Y.: Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. arXiv preprint arXiv:2302.01560 (2023)
49. Wang, Z., Wu, J., Cai, L., Low, C.H., Yang, X., Li, Q., Jin, Y.: Medagent-pro: Towards evidence-based multi-modal medical diagnosis via reasoning agentic workflow. arXiv preprint arXiv:2503.18968 (2025)
50. Wu, H., He, Z., Zhang, X., Yao, X., Zheng, S., Zheng, H., Yu, B.: Chateda: A large language model powered autonomous agent for eda. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **43**(10), 3184–3197 (2024)
51. Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E.E., Jiang, L., Zhang, X., Zhang, S., Awadallah, A., White, R.W., Burger, D., Wang, C.: Autogen: Enabling next-gen llm applications via multi-agent conversation. In: COLM 2024 (August 2024)
52. Yang, K., Tian, Y., Peng, N., Klein, D.: Re3: Generating longer stories with recursive reprompting and revision. arXiv preprint arXiv:2210.06774 (2022)

53. Yang, Z., Liang, K., Shi, Z., Han, Q., Zhao, W., Zeng, Z.: Circuit object detection and standard cell recognition in ic images based on semantic segmentation techniques. In: 2024 IEEE 2nd International Conference on Image Processing and Computer Applications (ICIPCA). pp. 300–303. IEEE (2024)
54. Yang, Z., Li, L., Wang, J., Lin, K., Azarnasab, E., Ahmed, F., Liu, Z., Liu, C., Zeng, M., Wang, L.: Mm-react: Prompting chatgpt for multimodal reasoning and action. arXiv preprint arXiv:2303.11381 (2023)
55. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R., Cao, Y.: React: Synergizing reasoning and acting in language models. In: The eleventh international conference on learning representations (2022)