arXiv:2508.03725v1 [cs.CV] 30 Jul 2025

A Large Language Model Powered Integrated Circuit Footprint Geometry Understanding

Yida Wang[1]* **Taiting Lu**[2]* **Runze Liu**[2] **Lanqing Yang**[1]† **Yifan Yang**[3]‡
Zhe Chen[1] **Yuehai Wang**[1] **Yixin Liu**[1] **Kaiyuan Lin**[2] **Xiaomeng Chen**[1]

Dian Ding[1] **Yijie Li**[4] **Yi -Chao Chen**[1] **Yincheng Jin**[5] **Mahanth Gowda**[2]

[1]Shanghai Jiao Tong University [2]Pennsylvania State University
[3]Microsoft Research [4]National University of Singapore [5]Binghamton University

{yidawang, yanglanqing, chenzhe2003, yuehaiw, panh09,
dingdian94, yichao, sjtu_chenxm}@sjtu.edu.cn
{txl5518, rml6043, kjl6302, mahanth.gowda}@psu.edu
yifanyang@microsoft.com louise_l@alumni.sjtu.edu.cn
yijieli@nus.edu.sg yjin5@binghamton.edu

Abstract

Printed-Circuit-board (PCB) footprint geometry labeling of integrated circuits (IC) is essential in defining the physical interface between components and the PCB layout, requiring exceptional visual perception proficiency. However, due to the unstructured footprint drawing and abstract diagram annotations, automated parsing and accurate footprint geometry modeling remain highly challenging. Despite its importance, no methods currently exist for automated package geometry labeling directly from IC mechanical drawings. In this paper, we first investigate the visual perception performance of Large Multimodal Models (LMMs) when solving IC footprint geometry understanding. Our findings reveal that current LMMs severely suffer from inaccurate geometric perception, which hinders their performance in solving the footprint geometry labeling problem. To address these limitations, we propose **LLM4-IC8K**, a novel framework that treats IC mechanical drawings as images and leverages LLMs for structured geometric interpretation. To mimic the step-by-step reasoning approach used by human engineers, LLM4-IC8K addresses three sub-tasks: perceiving the number of pins, computing the center coordinates of each pin, and estimating the dimensions of individual pins. We present a two-stage framework that first trains LMMs on synthetically generated IC footprint diagrams to learn fundamental geometric reasoning and then fine-tunes them on real-world datasheet drawings to enhance robustness and accuracy in practical scenarios. To support this, we introduce ICGEO8K, a multi-modal dataset with 8,608 labeled samples, including 4138 hand-crafted IC footprint samples and 4470 synthetically generated samples. Extensive experiments demonstrate that our model outperforms state-of-the-art LMMs on the proposed benchmark. The accurate translation of footprint diagrams enabled by LLM4-IC8K contributes to advancing standardization within the PCB industry.

*Both authors contributed equally.
†Corresponding author.
‡Corresponding author.

Preprint. Under review.

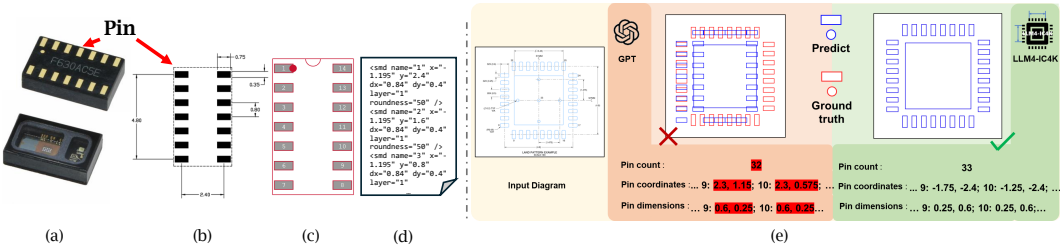


Figure 1: Illustration of IC geometry understanding problem. (a) Example IC packages with visible pins, serving as electrical and mechanical interfaces. (b) Corresponding datasheet footprint diagram showing pin positions and dimensions for PCB design. (c) Visualization of EDA footprint based on the datasheet diagram. (d) Raw EDA footprint source file. (e) Visualization of an IC footprint parsing task: GPT and LLM4-IC8K are given the same input diagram and tasked with extracting pin count, pin center coordinates, and pin dimensions. GPT produces incorrect answers, with errors highlighted in red, such as miscounted pin numbers or incorrect coordinate values.

1 Introduction

Large Language Models (LLMs) have shown promising performance in geometry and spatial reasoning [17][8], proof and logic reasoning [44], and Graph and Set Theory [40]. However, existing researches mainly focus on abstract geometric reasoning within textual and synthetic visual domains, whereas complex, real-world engineering tasks, such as understanding footprint geometry of Printed-Circuit-board (PCB) and integrated circuit (IC) package drawings, remain underexplored. Since general LLMs are not designed to handle dense IC footprint diagrams that include many annotations and many engineering drawing labels (as shown in Figure 2 and Appendix A.1), LLMs struggle to perform accurately on such visually complex tasks. Furthermore, collecting trainable diagram-description pairs for LLM involves massive labor efforts and expert experiences, and there are no publicly available datasets containing IC diagrams that can be directly utilized for training LLMs in IC geometry understanding.

PCB footprint geometry understanding has significant value in industrial production and PCB design because it ensures accurate component placement and reliable electrical connections on the PCB. In the PCB footprint geometry understanding, pins of the footprint define where and how the IC connects to the PCB board, determining both the physical placement and the electrical pathways needed for the circuit to function properly (as depicted in Figure 1). Inaccuracies in pin size or placement can lead to mismatched impedance or unintended parasitic effects. These issues can significantly degrade the performance and reliability of high-speed or sensitive circuits [9, 11]. In industrial production, a precise footprint ensures accurate component placement by automated assembly machines. Misaligned or incorrect footprint causes soldering defects, such as tomb-stoning or short circuits, resulting in production delays, and higher costs [34]. There have been numerous industrial electronic design automation (EDA), such as IPC compliant footprint wizard by Altium [4], Package Generator by Autodesk EAGLE [5], and Footprint Editor by KiCad [32], etc. While these tools provide manual footprint creation, they often require significant user input and lack automation in interpreting datasheet drawings. Motivated by the promising capabilities of LLMs in geometric and spatial reasoning, we first investigate their visual interpretation abilities to automate the perception of IC footprint information from datasheet drawings.

Understanding a precise IC footprint involves three core tasks: (1) Identifying **the number of pins** correctly to ensure accurate electrical connectivity. (2) Accurately determining **the spatial arrangement of all pins** relative to a clearly defined reference point (origin), such as the component's center or a specific pin, to ensure correct placement and alignment of the component on the PCB. (3) Determining **the exact size of each pin** is important to make sure the electronic parts are firmly attached and work properly. If the pin sizes in a PCB design are wrong, the electronic parts might not fit properly or connect correctly, resulting in weak electrical connections and short circuits. However, existing large language models typically struggle to interpret IC footprint geometry accurately. They lack the spatial reasoning capabilities and the domain-specific alignment of visual and textual information necessary to reliably comprehend technical drawings and translate them into precise geometric layouts.

In this paper, we propose LLM4-IC8K (**L**arge **L**anguage **M**odel for (**4**) **I**ntegrated **C**ircuit - 8K), a two-stage training framework designed to enable large language models to accurately understand

IC footprint geometry and automate the footprint generation process. by mimicking the step-by-step reasoning approach used by human engineers. In the first stage, we employ a synthetically generated dataset comprising clean and structured diagrams to address the challenges of collecting and annotating real-world training data, which is both labor-intensive and requires domain expertise. In the second stage, the model is further trained on real-world footprint drawings from datasheets (shown in Appendix A.1), which contain dense annotations and varied geometric labeling. This stage is essential for adapting the model to the complexity and noise present in practical applications, enabling it to generalize effectively to diverse visual styles and layout conventions encountered in real-world IC design. To further improve performance of the model, we structure the training prompt in a chain-of-thought (CoT) manner, to mimic the human reasoning process during fine-tuning.

To rigorously evaluate the performance of existing LLMs and our approach in IC footprint labeling, we construct a systematic benchmark that not only captures real-world complexities, such as varying pin counts and package styles, but also aligns closely with the actual distribution of IC footprint types observed in real-world datasets, covering all major package categories. The benchmark is detailed in Section 5.1 and shown in Appendix A.3. In addition, to assess the practical utility of our model, we compare its performance against manual footprint creation by experienced electrical engineers using standard EDA tools, focusing on both accuracy and time efficiency. Our main contributions of this article are threefold:

- To the best of our knowledge, we are the first to develop a novel benchmark on IC footprint geometry understanding, ICGEOQA, and investigate the capability of state-of-the-art general-purpose LLMs (GPT-4o [21], Gemini 2.0 [25], DeepSeek-VL2 [42], and Qwen2-VL [41]) on this problem. As shown in Table 1, these models struggle with precise geometric reasoning, achieving low overall layout accuracy (e.g., IoU_{IC}: GPT-4o 11.1%, Qwen2-VL 1.7%) and exhibiting high errors in pin localization and dimension estimation. In contrast, our proposed LLM4-IC8K achieves significantly superior performance with 71.6% IoU_{IC}, demonstrating its strong capability in accurate and fine-grained IC footprint understanding.
- We introduce a new multi-modal geometric reasoning dataset, ICGEO8K, containing a total of 8,608 labeled samples—4,138 collected from real-world IC footprint diagrams and 4,470 synthetically generated samples. The dataset contains three sub-tasks, each targeting a key aspect of geometric reasoning: (i) counting the number of pins, (ii) computing the center coordinates of individual pins, and (iii) estimating the dimensions of each pin. To construct the synthetic portion, we propose a novel data augmentation tool that transforms publicly available EDA footprint designs into datasheet-style diagrams with aligned annotations.
- We propose a two-stage training framework that enables LLMs to accurately understand IC footprint drawings and automatically generate corresponding footprint designs. Our approach achieves SOTA results on the real-world benchmark compared with existing LLMs. By emulating the step-by-step reasoning process of human engineers, our method achieves substantial efficiency gains, generating a complete IC footprint in 0.26 min/sample, representing up to a 58× reduction in time compared to traditional EDA tools, while maintaining comparable accuracy.

2 Related Work

Manual PCB Component Footprint Generation. Traditional PCB component geometry generation is labor-intensive and time-consuming, requiring manual interpretation of datasheets, footprint creation, symbol generation, and signal mapping [28, 1]. With hundreds of components in modern designs, this manual process becomes a bottleneck [26], prone to inconsistencies, human error, and outdated libraries due to frequent spec updates [35, 2]. The iterative nature of PCB design further complicates manual updates, delaying time-to-market. These limitations underscore the need for automated and data-driven solutions to improve efficiency and reliability.

Automated IC Footprint Geometry Understanding. Existing PCB labeling methods focus on the segmentation or classification of IC footprints [28, 43]. However, these methods do not attempt to understand the geometric information of IC pins, leaving the automated labeling process limited to a level above the individual pins. Although object detection methods [23, 33, 46] may help with counting the number and computing the relative size of the IC pins, they cannot handle footprint

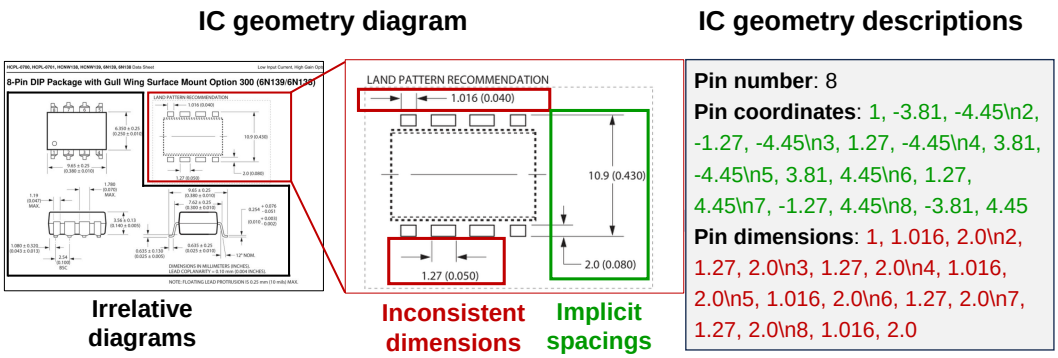


Figure 2: An example of IC geometry understanding. This starts with locating the correct diagram (upper-right corner). Then the dimensions (framed in red) and location (framed in green) of each pin are extracted and computed. Note that datasheet page images often contain irrelevant and misleading information, such as the 3-view diagrams, inconsistent dimension labels, and implicit spacing labels, adding to the complexity.

diagrams with implicit information omitted, as shown in Figure 1(d). Optical Character Recognition (OCR) methods [13, 14] can be used to extract diagram texts from datasheet images, but they cannot understand the physical and geometrical meaning of the numerical labels. There are works that utilize object detection and OCR for diagram object detection [20] or converting legacy schematic diagrams [29]. However, these methods fail to bridge the gap between the extracted annotations and the implicit geometric knowledge. Therefore, a method capable of performing logical reasoning by interpreting both the footprint diagrams and annotation information is needed for fully automated IC footprint geometry labeling.

LLM for Mathematic. Generative general-purpose models such as ChatGPT, DeepSeek, and Qwen2 have demonstrated strong generalization abilities across a wide range of tasks without the need for task-specific fine-tuning [7, 6, 19]. To address the problem of visual mathematical geometry reasoning, vision-language models like LLaVA, UniChart, and MathVista use large-scale image-text datasets to develop broad visual reasoning capabilities [24, 27, 16]. However, no existing work has explored the capabilities of large language models in understanding IC footprint geometry.

3 Data Collection

Building an IC footprint dataset is challenging due to the lack of a complete index of IC models and the absence of a unified source that provides both datasheets and corresponding EDA footprint files. This fragmentation requires collecting and aligning data from multiple platforms, making large-scale data acquisition labor-intensive, error-prone, and difficult to automate. To build a comprehensive IC footprint dataset, we follow a four-step pipeline by integrating resources from multiple platforms, as no single source provides all the necessary data. Detailed data collection pipeline is discussed following. (I) **Datasheet and EDA Collection.** We compile a comprehensive list of IC models from Digi-Key [15] and collect their associated datasheets, using IC part numbers as unique identifiers. To supplement missing EDA files, we retrieve standardized footprint and CAD models from UltraLibrarian [38], aligning them with the datasheets based on these part numbers. (II) **AI-Aided Diagram Page Finding.** Because IC footprint diagrams in datasheets are often embedded within extensive technical content, we leverage general-purpose LLMs such as Gemini 2.0 [25] to identify the page numbers containing relevant IC footprint diagrams. Manual verification is subsequently performed to ensure the accuracy and relevance of the extracted images prior to their inclusion in the dataset. (III) **Image and Label Processing.** The IC footprint diagrams extracted from datasheets are aligned with the corresponding geometric descriptions from EDA source files using IC part numbers as identifiers. From this alignment, we extract key properties, including the number of pins, their center coordinates, and their dimensions, to construct fully labeled examples for the dataset. (IV) **Expert Data Correction.** Expert engineers are engaged to review and correct mismatches between EDA-derived labels and datasheet diagrams, addressing inconsistencies due to design variability. An example illustrating IC geometry understanding is presented in Figure 2 and Appendix A.1, where

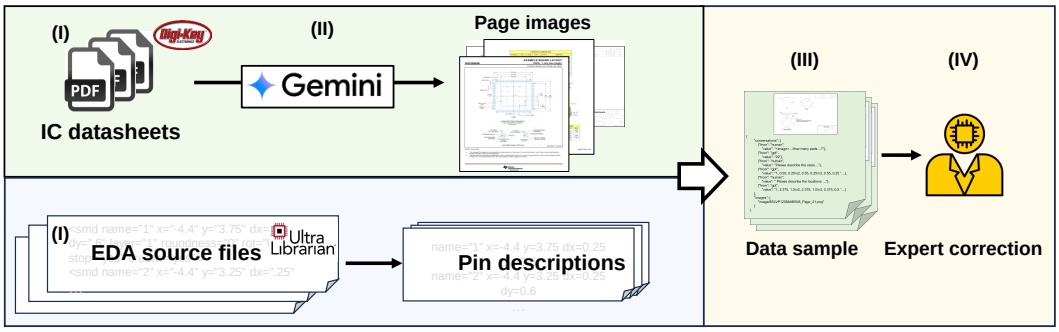


Figure 3: Dataset building process. A valid sample in ICGEO8K is developed in four steps: (I) datasheets and EDA files collection, (II) AI-aided diagram page finding, (III) image and label processing, and (IV) label correction by PCB experts.

the IC geometry diagram is embedded within a datasheet page, and the corresponding geometry descriptions are extracted from the associated EDA file and organized in JSON format.

3.1 Dataset Building

Phase I: Datasheet and EDA collecting. Our dataset consists of paired samples, comprising IC footprint diagrams as input data and corresponding IC geometry descriptions as structured labels (see Figure 2). To construct this dataset, we first extract a broad list of IC models from Digi-Key [15], a global distributor offering over 13 million electronic products from more than 2,000 manufacturers. IC part numbers are used as unique identifiers to ensure consistent alignment across different data sources. We then collect the corresponding datasheets from Digi-Key, which provide essential design information such as footprint diagrams, mechanical package types, and reference application circuits. Due to the limited availability of EDA resources on Digi-Key, we augment the dataset by retrieving standardized footprint source files, based on the part number from UltraLibrarian [38], a widely used platform for manufacturer-approved EDA content. The IC geometry descriptions, including properties such as pin count, center coordinates, and physical dimensions, are extracted from the EDA source files and aligned with the visual footprint diagrams.

Phase II: AI-aided diagram page finding. Because datasheets use different formats, terms, and visual styles, and often include multiple footprint diagrams for different versions of the same chip, it is difficult to directly use them as input for training without first carefully processing and selecting the correct IC footprint images. To address this challenge, we utilize Gemini 2.0[25] to assist with identifying the datasheet pages that contain IC footprint diagrams. Given that the model's predictions are not always precise, manual verification and correction by expert engineers are conducted to ensure accurate localization. The verified pages containing suggested pad layout diagrams are then extracted and used as image inputs for downstream IC geometry understanding tasks.

Phase III: Image and label processing. After identifying and verifying the page number containing the IC footprint diagrams in the datasheet, we extract that pages as images to serve as the input data. In parallel, we preprocess the corresponding EDA source file to extract structured pin-level information, as they often include extra, unrelated design details such as general component information, wiring data, and board layout elements. To focus only on the footprint geometry, we extract and organize three key properties: (i) the total number of pins, (ii) the center coordinates of each pin, and (iii) the size of each pin. These are then formatted to match the label structure shown in Figure 2.

Phase IV: Expert data correction. Although the number of pins and their relative positions in the EDA file are generally consistent with the IC footprint diagrams in the datasheet, discrepancies often arise in the exact pin dimensions and precise locations. These inconsistencies, caused by variations in EDA file formatting and design conventions, can prevent LLMs from accurately understanding or inferring the geometric details found in EDA files solely from the visual information in datasheet diagrams. Therefore, we engage human engineering experts to review and correct any inaccurate labels in the EDA files, ensuring that they precisely match the specifications outlined in the IC geometry diagrams. All PCB engineers involved are paid 0.61$ per processed sample. After manual

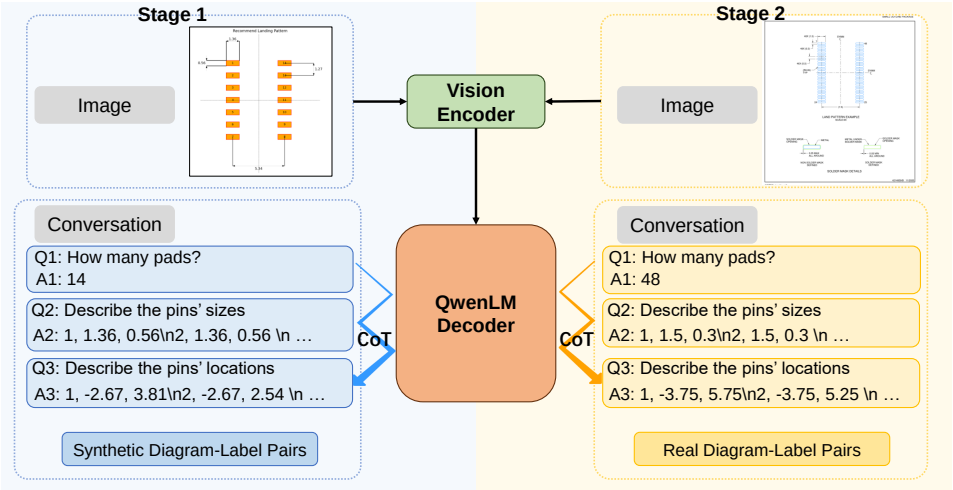


Figure 4: Two-stage training overview. We apply a two-stage supervised fine-tuning on Qwen2-VL-7B. The questions in a conversation are ordered in a chain-of-thought (CoT) manner.

correction and filtering, we carefully selected $4,138$ IC footprint entries, spanning 10 distinct IC package types, each consisting of datasheet-oriented IC footprint diagrams and their corresponding geometry description labels. We name our real-world collected dataset ICGEO8K.

3.2 Synthetic Diagram Augmentation

Due to the time- and labor-intensive nature of manual labeling, collecting a large-scale real-world dataset is challenging, which in turn limits the effectiveness of model fine-tuning. To overcome this limitation, we developed a footprint diagram generation toolkit that synthesizes clean, datasheet-style footprint images from imprecise EDA geometry descriptions. This approach enables the construction of a synthetic dataset, wherein the synthetic images serve as inputs and the corresponding EDA footprint descriptions provide accurate ground truth annotations for training. Comparisons of synthetic and real-world diagrams are shown in Figure 8 in Appendix A.2. The synthetic dataset contains 4,470 data samples synthesized from EDA files collected from **Phase I**. Combined with the real-world dataset with $4,138$ manually labeled real-world IC footprint diagram-label pairs, our final dataset ICGEO8K comprises a total of 8,608 entries. The difference between the synthetic part and the real-world part in entry numbers arises from manual filtering in real-world samples, where real-world diagrams are removed if their pins are vague or irregular.

4 Training Pipeline

To realize the geometric reasoning ability for understanding IC footprint diagrams, this section introduces our proposed LLM4-IC8K in detail. The main intuition is to leverage the capabilities of LLMs to perceive the relationships between image patterns and annotations, and to derive effective information progressively, similar to how humans process and reason through such information. To achieve this goal, we propose a two-stage, end-to-end IC geometry labeling framework (Figure 4) that progressively extracts pin geometries through a chain-of-thought manner.

4.1 Supervised Fine-tuning with Chain-of-thought

When human engineers draw EDA footprint based on the IC footprint diagrams in IC datasheets, several critical pieces of information need to be extracted from the diagrams, including the type of IC, the total number of pins, the locations of the pins, and the dimensions of the pins. The thinking process in this procedure follows a progressive approach from the whole to the details: first, the engineer identifies the suggested pin layout diagram in the datasheet and determines the type of IC. Next, the engineer counts the number of pins and identifies their indices. Then, the engineer identifies and reads the spacings between the pins for positioning, followed by determining the pins' widths and lengths to define their dimensions. Drawing inspiration from this reasoning approach, LLM4-IC8K should also mimic this thinking process to derive the geometric information in a step-by-step manner.

To imitate human reasoning logic, we structure each query conversation for an IC geometry labeling task in three progressive questions: (1) *the number of pins in the IC footprint diagram*, (2) *the coordinates of each pin relative to the center of the diagram*, and (3) *the dimensions of each pin in millimeters*. **Question 1** directs the LLM to locate the correct diagram and identify the pins within the diagram. **Question 2** guides the LLM to identify each pin in index order and learn how to locate and interpret the corresponding annotations that reflect pin spacings. **Question 3** guides the LLM to learn how to identify and interpret the corresponding annotations that reflect pin dimensions. Example query conversations are shown in Figure 4.

4.2 Two-stage Training

To fully fine-tune the model for IC geometry understanding, we propose a two-stage, end-to-end training process. The LLM is initially trained on the synthetic samples of ICGEO8K, introduced in Section 3.2. Since the IC diagrams in the synthetic part are generated from EDA descriptions using a plotting toolkit, the annotation rules and graphic patterns are relatively simple and uniform. This allows the LLM to acquire basic geometric reasoning capabilities without interference from complex real-world distractions. After the first round of training, the LLM undergoes further training on the real-world samples of ICGEO8K, where the target diagrams are embedded within datasheet pages. These real-world diagrams vary in image resolution, annotation rules, label text fonts, and presentation styles (examples shown in Figure 9 in Appendix A.3). Additionally, multiple diagrams may appear on the same datasheet page, increasing the complexity of identifying the desired diagram. During the second training stage, the model adapts to complex real-world scenarios, enhancing its ability to understand geometric information in IC footprint diagrams.

5 Experiments

5.1 Dataset Analysis

We utilize proposed ICGEO8K dataset to train the LLM. The package type of ICs can be categorized into 10 categories based on their pin types and placements, based on established standards and guidelines provided in [22, 36]. Our dataset covers all package categories, reflecting the real-world distribution of common IC footprints. ICGEO8K possesses IC footprints with pin counts ranging from 1 to 800, implying the complexity in scales for IC pin labeling.

As no prior work addresses pin-level IC geometry labeling, we introduce a novel benchmark, **ICGEOQA**, to systematically evaluate the performance of LLMs in understanding IC footprint geometry. ICGEOQA consists of 400 carefully curated real-world IC entries sampled from ICGEO8K. To ensure that the benchmark reliably reflects real-world IC diversity, we analyzed the package type distribution across $200,000$ IC entries crawled from Digi-Key. The analysis confirms that ICGEOQA exhibits an identical package category distribution to the full 200K dataset, and it shares the same pin count distribution as ICGEO8K. These validations demonstrate that ICGEOQA can reflect the distribution of real-world IC geometries and satisfy the needs of PCB engineers. Note that the IC entries in ICGEOQA are excluded from ICGEO8K during the training process. Please refer to Appendix A.3 and the Supplementary Material for further detailed descriptions of our benchmark and datasets.

5.2 Experiment Setup

Implementation Details. We fine-tune a footprint geometry understanding model based on Qwen2-VL [41], a SOTA LLM in image understanding with Naive Dynamic Resolution mapping and Multi-modal Rotary Position Embedding (M-ROPE). As a balance between performance and computational cost, we choose its 7B version (Qwen2-VL-7B) as our base model. We utilize LLaMA-Factory [45] to fine-tune our model. As stated in Section 4.2, we first implement SFT on the LLM with the synthetic training samples for 3 epochs, and then fine-tune the LLM with real-world samples for 3 epochs. We use Low-Rank Adaptation (LoRA) for model training and set the cut-off length to 4096 and the learning rate to $5e^{-5}$. During each training stage, we randomly split 10% training samples as a validation set. All experiments are conducted on 2 NVIDIA A100-40G GPUs, and the batch size is set to 2 per GPU at both training stages.

Evaluation Metric. As no existing work focuses on pin-level IC geometry labeling, we develop our own metric, IoU_{IC}, defined as:

$$IoU_{IC} = \frac{Area_{pred} \cap Area_{label}}{Area_{pred} \cup Area_{label}} \tag{1}$$

where $Area_{pred}$ is the area of the pin layout reconstructed from the predicted pin geometry, and $Area_{label}$ is the area of the pin layout reconstructed from ground truth pin geometry. IoU_{IC} ranges from 0 to 1, where a value of 1 indicates the predicted pin layout completely overlaps with the ground truth pin layout, meaning the LLMs provides the correct answer precisely. A value of 0 indicates no overlap between predicted and ground truth layouts, denoting a completely incorrect result.

As stated in Section 4.1, the IC pin geometry is described by three sub-questions: the number of pins, the position of each pin, and the dimension of each pin. Therefore, we also evaluate model performance for each sub-task. For **task 1 (pin counting)**, we use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), where lower values indicate a more accurate counting outcome. For **task 2 (pin positions)**, we calculate the Euclidean distance between each predicted pin's center coordinates and its corresponding ground truth. Then, the distances of all pins are averaged as d_{pin}, where a lower value indicates a more accurate position prediction. For **task 3 (pin dimensions)**, we use the average IoU between the predicted and ground truth pattern of each pin, IoU_{pin}, as the evaluation metric. A more accurate dimension prediction achieves an IoU_{pin} closer to 1.

5.3 Comparison with General LLMs

To evaluate the IC geometric reasoning capability of LLM4-IC8K, we make comprehensive comparisons between current SOTA general LLMs, namely GPT-4o [21], Gemini 2.0 [25], DeepSeek-VL2 [42], and Qwen2-VL-7B [41] with LLM4-IC8K on ICGEOQA. All general LLMs are called via API. To ensure the outputs of general LLMs are compatible and have a consistent format as that of LLM4-IC8K, we utilize single-shot prompt engineering to inform the LLMs of the output formats.

Table 1: Comparison of QA performance with general LLMs on 3 tasks in mean + std format

Methods	**Overall** (IoU_{IC} **%**)	**Task 1**		**Task 2** (d_{pin})	**Task 3** (IoU_{pin} **%**)
		MAE	RMSE		
GPT-4o	11.1 ± 0.4	8.21 ± 0.47	23.04 ± 0.22	4.01 ± 0.02	45.6 ± 0.3
Gemini 2.0	4.5 ± 0.1	1.84 ± 0.40	7.87 ± 0.85	18.27 ± 0.45	57.3 ± 0.4
DeepSeek-VL	1.5 ± 0.8	21.97 ± 0.35	41.70 ± 2.15	4.22 ± 0.23	20.1 ± 0.2
Qwen2-VL-7B	1.7 ± 0.1	19.17 ± 0.37	43.76 ± 0.84	3.63 ± 0.54	41.1 ± 0.5
LLM4-IC8K	**71.6±0.5**	**0.35±0.07**	**2.81±0.08**	**1.11±0.02**	**88.0±0.3**

As shown in Table 1, our model demonstrates superior performance among existing SOTA general LLMs. LLM4-IC8K achieves a IoU_{IC} of 71.6%, indicating its ability to label IC geometries accurately, while all general LLMs receive IoU_{IC} below 20%, implying failures in labeling IC geometries. Since general LLMs do not have prior knowledge of IC footprint geometry labeling, they fail to understand the meaning of the numerical annotations and graphic symbols in the diagrams, such as mistaking pin spacing annotations as dimensions and miscounting pin numbers with omission symbols. Moreover, general LLMs struggle in accurately identifying numbers, resulting in great misinterpretations of geometric information (Figure 12 in Appendix A.4). On the other hand, LLM4-IC8K acquires prior knowledge on IC footprint diagrams, gaining the capability of labeling IC footprint geometry accurately.

5.4 Ablation Study

5.4.1 Different Dialogue Training Strategies

As described in Section 4.1, the problem of IC footprint geometry labeling can be divided into three distinct yet logically connected tasks: **pin number counting (task 1)**, **pin position understanding (task 2)**, and **pin dimension understanding (task 3)**. These tasks differ in complexity and interdependency. For instance, accurately counting pin numbers may be relatively straightforward, yet errors made at this initial stage will prevent correct solutions to the subsequent tasks. Considering these logical relationships, we evaluate the QA performance across various training strategies, reflecting different dialogue sequencing approaches.

Table 2: Summary of dialogue and dataset training strategies

Dialogue Training Strategy			Dataset Training Strategy		
Strategy	**Rounds**	**Task Order**	**Strategy**	**Stages**	**Data Order**
S1	1	{123}	T1	1	{real-world}
S2	1	{1,2,3}	T2	1	{real-world, synthetic}
S3	2	{1}{23}	T3	2	{real-world}{synthetic}
S4	2	{12}{13}	T4	2	{synthetic}{real-world}
S5	3	{1}{2}{3}	–	–	–

Table 3: Comparison of performance with existing EDA tools

Methods	**Overall (IoU_{IC} %)**	**Time Cost (min/sample)**
Altium	95 ± 0.4	7
EAGLE	80 ± 1.5	15
KiCAD	83 ± 1.7	15
LLM4-IC8K	71.6 ± 0.5	**0.26**

We organize our training strategies in each training stage as shown in Table 2. Each training round using different training samples is presented in curly braces ({ }), the digits 1, 2, and 3 denote the QAs of the corresponding task, and the order of the digits represents the order of the tasks in the conversation. The commas denote that the tasks are separated into independent samples.

Table 4: Comparison of QA performance with different training strategies

Factor	**Strategy**	**Metrics**				
		Overall (IoU_{IC} %)	**Task 1**		**Task 2 (d_{pin})**	**Task 3 (IoU_{pin} %)**
			MAE	RMSE		
Dialogues	**S1 (LLM4-IC8K)**	**71.6±0.5**	0.35 ± 0.07	2.81 ± 0.08	**1.11±0.02**	**88.0±0.3**
	S2	63.5 ± 0.4	**0.09±0.05**	**0.39±0.17**	1.20 ± 0.10	82.5 ± 0.1
	S3	62.6 ± 0.4	5.60 ± 0.04	0.63 ± 0.14	1.26 ± 0.12	85.6 ± 0.1
	S4	31.3 ± 0.5	0.63 ± 0.07	5.67 ± 0.04	2.23 ± 0.14	75.6 ± 0.1
	S5	25.4 ± 0.3	10.27 ± 0.06	1.09 ± 0.21	2.94 ± 0.16	74.6 ± 0.2
Training Stages	T1	65.1 ± 0.1	0.59 ± 0.06	3.30 ± 0.11	1.17 ± 0.03	81.8 ± 0.4
	T2	68.2 ± 0.3	0.41 ± 0.05	2.91 ± 0.03	**1.03±0.06**	85.3 ± 0.3
	T3	24.7 ± 0.5	3.17 ± 0.06	36.99 ± 0.33	3.69 ± 0.20	64.9 ± 0.2
	T4 (LLM4-IC8K)	**71.6±0.5**	**0.35±0.07**	**2.81±0.08**	1.11 ± 0.02	**88.0±0.3**

As shown in Table 4, the training strategy S1 achieves the highest IoU_{IC} of 71.6%. Compared with independent training in S2, the IoU_{IC} of S1 increases by 12.8%. This observation suggests that building a chain of thoughts among labeling tasks significantly enhances IC geometry understanding. The sub-optimality of multi-round training leads to biased learning, where the model retains more recent task knowledge more effectively, while earlier knowledge gradually fades.

5.4.2 The Benefits of the Synthetic Image Dataset

As stated in Section 4.2, LLM4-IC8K applies a two-stage training where it is trained under the synthetic part of ICGEO8K in the first stage and the real-world part in the second stage. To explore the role of synthetic data in enhancing model performance, we evaluate the QA performance under various training strategies under different dataset part combinations as shown in Table 4. T4 significantly improves performance, increasing IoU_{IC} by 10%, 5%, and 189.9% compared to T1, T2, and T3, respectively. LLM4-IC8K gains a deeper understanding of real-world IC geometry by first learning general IC footprint patterns from synthetic data, and subsequently adapting to more complex real-world diagrams.

5.5 Comparison with Manual Labeling

To emphasize LLM4-IC8K's efficiency on IC footprint geometry labeling, we compare the effort of automated labeling using LLM4-IC8K with traditional manual labeling using EDA tools. We conduct labeling experiments on ICGEOQA with professional IC engineers using three common EDA software: Altium [4], Autodesk EAGLE [5], and KiCAD [32]. The labeling times are recorded and averaged by the total sample number. We also record the time of automated labeling using LLM4-IC8K. As shown in Table 3, Autodesk EAGLE and KiCAD require tens of minutes to process an IC diagram. While Altium provides IC templates to accelerate manual labeling, it still costs 7 minutes on average. On the other hand, automated labeling via LLM4-IC8K costs only 15 seconds per image with an accuracy of over 70% in IoU, indicating the groundbreaking evolution our model brings about.

6 Conclusion and Discussion

We present the first systematic investigation of general-purpose LLMs in understanding IC footprint geometry, revealing their limitations in precise spatial reasoning tasks. To address this, we intro-

duce ICGEO8K, a multi-modal dataset targeting IC geometric understanding, and augment it with ICGEO8K-SYN for broader coverage. We develop LLM4-IC8K, enabling LLMs to accurately interpret footprint diagrams and generate designs with SOTA performance.

While we use a two-stage supervised training to fine-tune the LLM, we are aware that Reinforcement Learning (RL) methods such as Group Relative Policy Optimization (GRPO) [30] or systematic designs such as the use of agents may further improve model performance. We plan to fully exploit the potential of RL methods and the use of agents in our future work. We plan to continue scaling up the real-world dataset to establish a more comprehensive and representative benchmark. Additionally, by continuously collecting practical data and feedback, we seek to gain deeper insights into the actual needs of IC engineers. Nevertheless, our work leverages the capability of general LLMs to comprehend complex IC diagram geometry, thereby paving the way toward fully automated PCB engineering and the advancement of geometry-aware LLM development.

References

[1] Things to learn before you learn pcb designing, Apr 2021.

[2] Ronald Abraham. High-quality pcb libraries matter in pcb design, Mar 2025.

[3] Ronald Abraham. Pcb documentation standards for reliable production, Mar 2025.

[4] Altium Limited. *IPC Compliant Footprint Wizard*, 2022.

[5] Autodesk, Inc. Library basics part 1: Creating your first package in autodesk eagle, 2020.

[6] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[7] Sébastien Bubeck, Varun Chadrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

[8] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision language models, 2024.

[9] Sierra Circuits. How to reduce parasitic capacitance in pcb layout, 2023.

[10] Component Search Engine. Component search engine: Free symbols, footprints, & 3d models, 2025.

[11] Intel Corporation. An530: Optimizing impedance discontinuity caused by surface mount pads for high-speed channel designs, 2023.

[12] DigiKey. Digikey - electronic components distributor, 2025.

[13] Yongkun Du, Zhineng Chen, Hongtao Xie, Caiyan Jia, and Yu-Gang Jiang. Svtrv2: Ctc beats encoder-decoder models in scene text recognition. *arXiv preprint arXiv:2411.15858*, 2024.

[14] Chen Duan, Qianyi Jiang, Pei Fu, Jiamin Chen, Shengxi Li, Zining Wang, Shan Guo, and Junfeng Luo. Instructocr: Instruction boosting scene text spotting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 2807–2815, 2025.

[15] DigiKey Electronics. Digikey electronics - electronic components distributor. `https://www.digikey.com`, 2025. Accessed: 2025-05-14.

[16] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*, 2023.

[17] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, and Lingpeng Kong. G-llava: Solving geometric problem with multi-modal large language model, 2023.

[18] Poulomi Ghosh. Design mistakes that cause pcb assembly errors, Sept 2023.

[19] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[20] Xin Hu, Lingling Zhang, Jun Liu, Jinfu Fan, Yang You, and Yaqiang Wu. Gptr: gestalt-perception transformer for diagram object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 899–907, 2023.

[21] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[22] IPC. Ipc-7351b: Generic requirements for surface mount design and land pattern standard. `https://www.ipc.org/TOC/IPC-7351B.pdf`, 2010. Available from IPC Organization.

[23] Glenn Jocher. Yolov5 by ultralytics, 2020.

[24] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

[25] Shrestha Basu Mallick and Logan Kilpatrick. Gemini 2.0: Flash, flash-lite and pro, February 2025. Accessed: 2025-05-15.

[26] Jim Martens. Electronic part selection and library creation made simple, Aug 2022.

[27] Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. Unichart: A universal vision-language pretrained model for chart comprehension and reasoning. *arXiv preprint arXiv:2305.14761*, 2023.

[28] Yun-Jie Ni, Yan-Jhih Wang, and Tsung-Yi Ho. Footprint classification of electric components on printed circuit boards. In *Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD*, pages 169–174, 2020.

[29] Jukka K Nurminen, Kari Rainio, Jukka-Pekka Numminen, Timo Syrjänen, Niklas Paganus, and Karri Honkoila. Object detection in design diagrams with machine learning. In *Progress in Computer Recognition Systems 11*, pages 27–36. Springer, 2020.

[30] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[31] SnapMagic Search. Snapmagic search - free pcb footprints and schematic symbols, 2025.

[32] SparkFun Electronics. Creating a custom kicad footprint library, n.d.

[33] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[34] Altium Designer Team. Diabolical footprint mistakes and how to avoid them, 2022.

[35] John Teel. Pcb design: The top 5 mistakes made on printed circuit board layout, Sept 2023.

[36] Texas Instruments. Texas instruments packaging information. `https://www.ti.com/packaging`, 2023.

[37] Ultra Librarian. Ipc-7351: Explaining the standard for pcb footprints, 2023.

[38] Ultra Librarian. Ultra librarian: The world's largest free pcb cad library, 2024.

[39] Ultra Librarian. Ultra librarian: Free online pcb cad library, 2025.

[40] Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. Llms as zero-shot graph learners: Alignment of gnn representations with llm token embeddings. *Advances in Neural Information Processing Systems*, 37:5950–5973, 2024.

[41] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[42] Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024.

[43] Zhen Yang, Kun Liang, Zuozheng Shi, Qing Han, Wei Zhao, and Zeng Zeng. 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)*, pages 300–303. IEEE, 2024.

[44] Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 5168–5191. Curran Associates, Inc., 2023.

[45] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*, 2024.

[46] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A Appendices

A.1 Data sample formation example

This section shows an example of the data processing pipeline for a dataset sample. Figure 5 shows the real-world datasheet pages of the target IC entry. In this example, the information about the IC footprint diagram is distributed over two pages. The two pages are concatenated into one image. The target diagram is located inside the red frame, while the associated parameter labels about the coordinates and dimensions of pins are framed in green and the corresponding values are framed in blue.

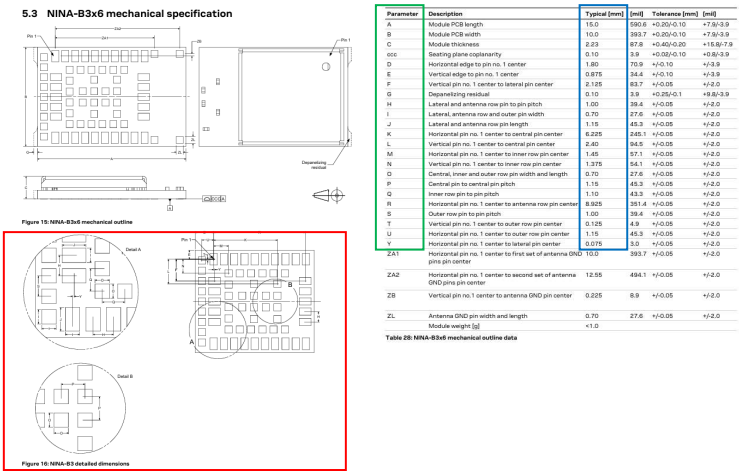


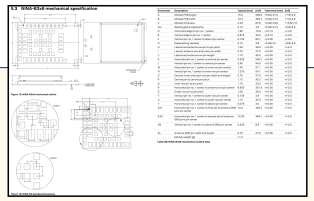
Figure 5: Real-world datasheet example.

Figure 6 shows the EDA description of pins for the same IC entry. The EDA descriptions are in XML format and contain the location and dimension information of each pin. To visualize clearly, the pin indices are colored in blue, the coordinates of pins are colored in orange, and the dimensions of pins are colored in red.

```
<packages>
<package name="XCVR_NINA-B306-00B">
…
<smd name="1" x="-4.125" y="5.7" dx="1.15" dy="0.7" layer="1"/>
<smd name="2" x="-4.125" y="4.7" dx="1.15" dy="0.7" layer="1"/>
<smd name="3" x="-4.125" y="3.7" dx="1.15" dy="0.7" layer="1"/>
<smd name="4" x="-4.125" y="2.7" dx="1.15" dy="0.7" layer="1"/>
<smd name="5" x="-4.125" y="1.7" dx="1.15" dy="0.7" layer="1"/>
<smd name="6" x="-4.125" y="0.7" dx="1.15" dy="0.7" layer="1"/>
<smd name="7" x="-4.125" y="-0.3" dx="1.15" dy="0.7" layer="1"/>
<smd name="8" x="-4.125" y="-1.3" dx="1.15" dy="0.7" layer="1"/>
<smd name="9" x="-4.125" y="-2.3" dx="1.15" dy="0.7" layer="1"/>
<smd name="10" x="-4.125" y="-3.3" dx="1.15" dy="0.7" layer="1"/>
…
</package>
```

Figure 6: EDA description example.

Figure 7 shows the final data sample of the IC entry. The data sample comprises the datasheet page image and the JSON format conversation QAs. Note how the values in the three QAs are matched with those in Figure 6.




```
{
   "conversations": [
     {"from": "human",
       "value": "<image>…How many pins are there in the package image?"},
     {"from": "gpt",
       "value": "8"},
     {"from": "human",
       "value": "let the coordinate of the Suggest Pad's center be 'x=0, y=0'. Please describe the center locations of the smd pads in
number pairs organized as \"name, x, y\\n\""},
     {"from": "gpt",
       "value": "1, -4.125, 5.7\n2, -4.125, 4.7\n3, -4.125, 3.7 \n4, -4.125, 2.7 \n5, -4.125, 1.7 \n … "}
     {"from": "human",
       "value": "Please describe the sizes of the pins in number pairs organized as \"name, dx, dy\\n\" for a square pad or \"name,
diameter\\n\" for a circle pad."},
     {"from": "gpt",
       "value": "1, 1.15, 0.7 \n2, 1.15, 0.7 \n3, 1.15, 0.7 \n4, 1.15, 0.7 \n5, 1.15, 0.7 … "},
],
   "images": [
     "image/ XCVR_NINA-B306-00B _Page_43.png"
   ]
 }
```


Figure 7: Dataset sample example.

A.2 Comparison between synthetic diagrams and real-world datasheets

Sample 1 and 2 are two distinct ICs. The left-half figure shows the real presentations of the two samples (all real-world diagrams in the following sections are the zoomed-in targeted diagrams), and the right-half figure shows their synthetic counterparts. The synthetic diagrams are simple and unified in colors and label formats, while the real pages contain disturbances and are organized in various formats.

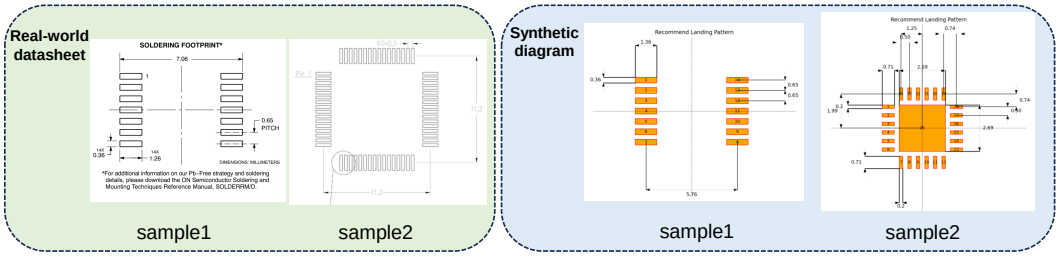


Figure 8: Examples of synthetic diagrams and real-world datasheets.

A.3 Benchmark data distribution

We downloaded $200K$ IC information entries, of which $50K$ have available datasheets. **IC EDA files:** Ultra Librarian is a comprehensive electronic component library offering access to 16 million verified components described in EDA design files. These files provide real-world examples of IC landing patterns, allowing extraction of numerical descriptions of IC parts (*i.e.*, pin positions and sizes). We download EDA design files corresponding to the collected IC datasheets. However, since EDA files are individually provided by manufacturers or engineers, over half of the datasheets in the Digi-Key library lack matching EDA files, reducing the pool of valid image-label pairs to fewer than $25K$ entries.

Figure 9 shows a variety of IC layout patterns for different package types. Different IC package types are largely distinct from each other in pin types, pin numbers, pin shapes, and placements, indicating the complexity of the IC geometry understanding problem.

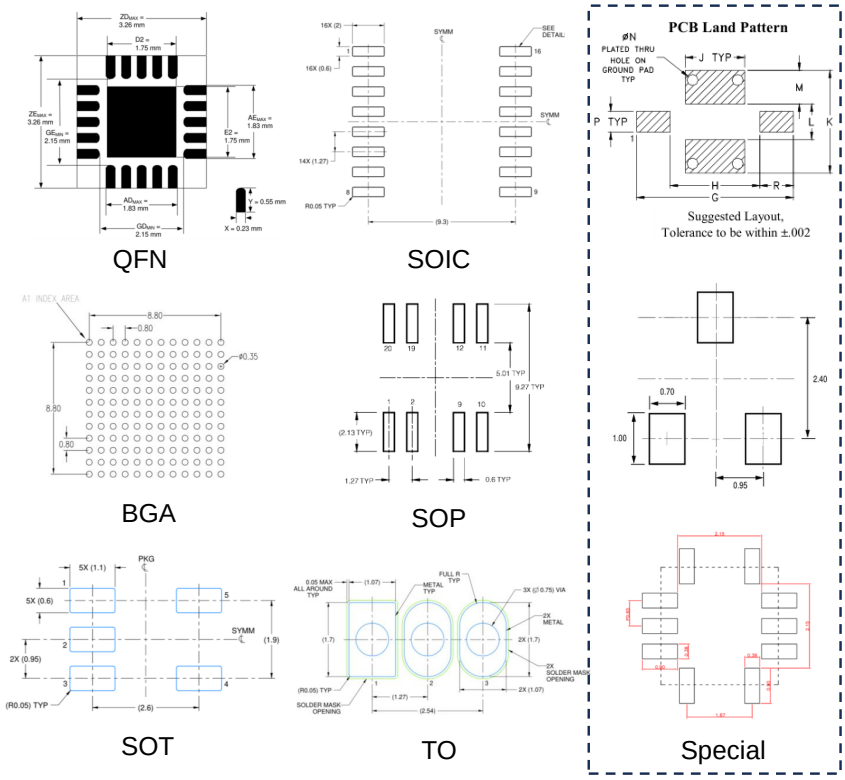


Figure 9: Examples of different diagram patterns for distinct package types.

Figure 10 illustrates the package type distribution of ICGEOQA alongside the $200K$ IC footprint entries collected from Digi-Key. The distribution of ICGEOQA is deliberately designed to mirror that of the $200K$ IC footprint dataset, ensuring that our benchmark accurately reflects the practical requirements of PCB engineers in their daily design tasks.

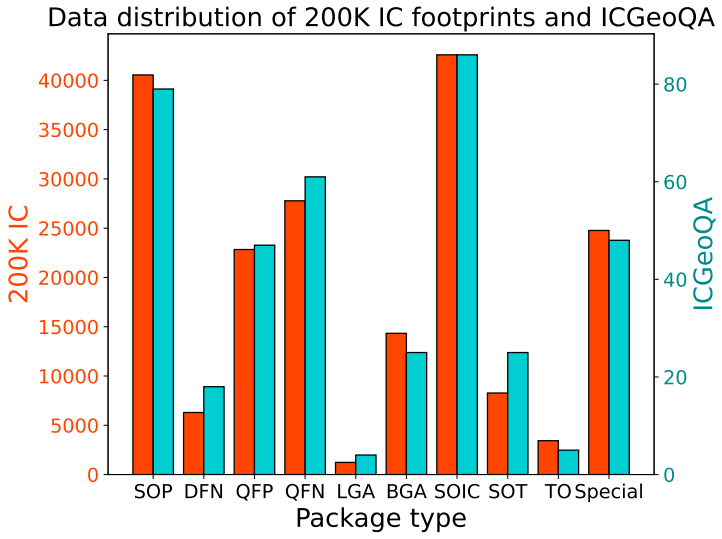


Figure 10: Examples of synthetic diagrams and real-world datasheets.

Figure 11 shows the distribution of pin numbers of our benchmark, indicating the diversity in pin count cases. As over 80% of the samples contain less than 100 pins, it's reasonable to set the cut-off length of our model to 4096 to avoid massive token truncation.

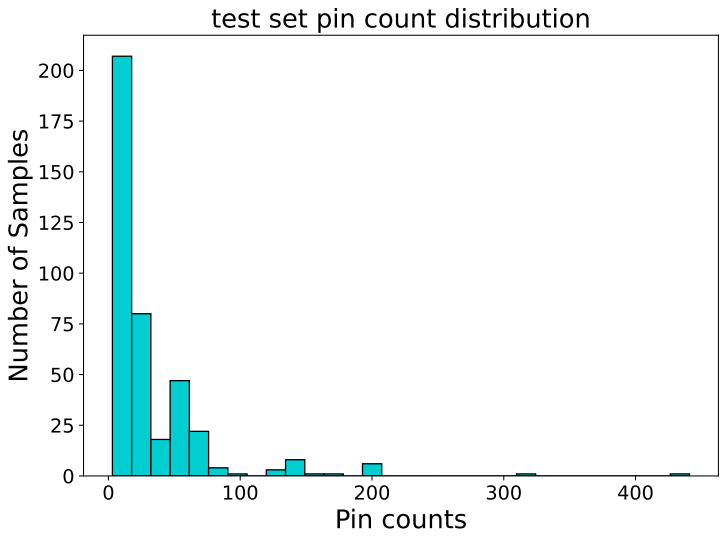


Figure 11: The pin count distribution of ICGEOQA.

A.4 Qualitative results

Figure 12 shows the qualitative results. In the experiment, LLM4-IC8K and GPT-4o are both employed to predict the pin dimensions and locations given the same IC diagrams. We then visualized the pin patterns of both ground truth and predicted descriptions. The squares and circles in the images are visualized IC pins according to label descriptions, the predicted results by LLMs are colored in blue, and the ground truth images are colored in red. Note that the results predicted by LLM4-IC8K overlaps perfectly with the ground truth images, while GPT-4o performances poorly in all 3 sub-tasks. The results suggest that LLM4-IC8K significantly outperforms the SOTA general LLM (GPT-4o) in predicting IC pin quantities, positions, and dimensions, achieving near-perfect alignment with the ground truth, whereas GPT-4o exhibits significant inconsistencies.

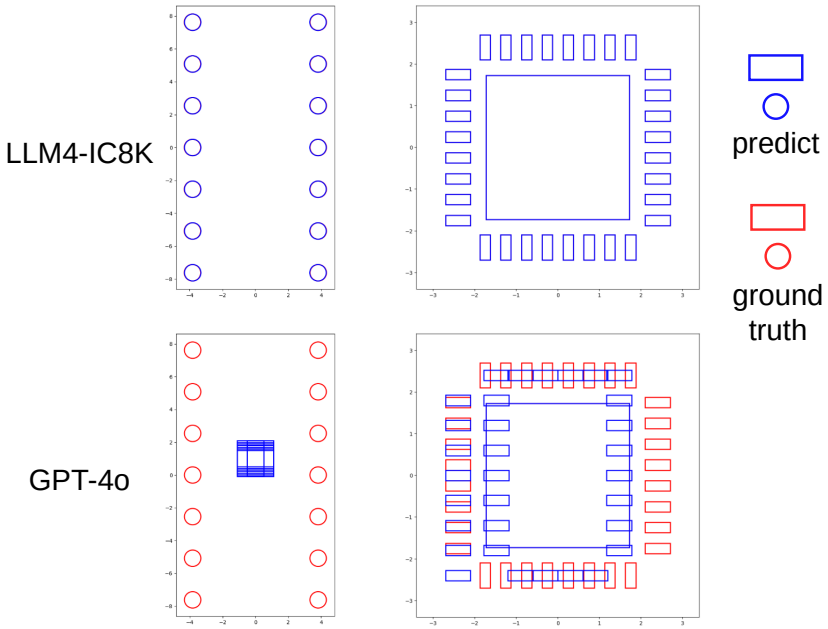


Figure 12: Visualization of pin dimensions and locations of ground truth and generated results.