UC San Diego

IoTGen: Towards LLM-Driven IoT Hardware PCB Generation

Qinpei Luo[1,2], Ruichun Ma[2], Xinyu Zhang[1], Lili Qiu[2]

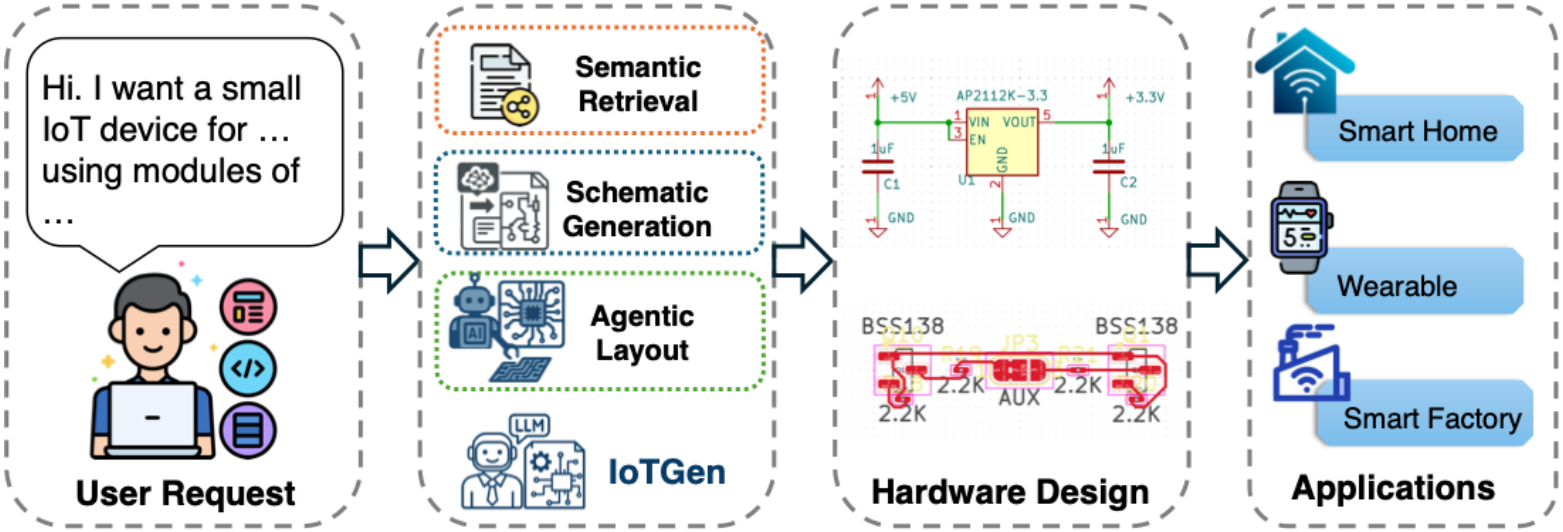
UC San Diego

Microsoft Research

Mobisys 2026 | Jun 24, 2026

UC San Diego

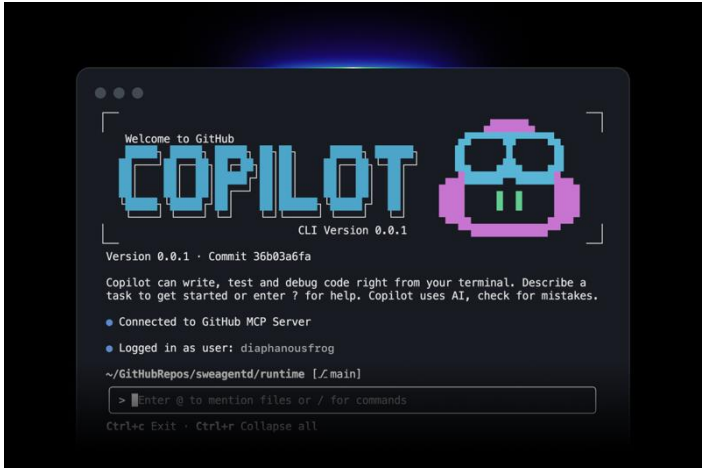
Introduction



- With the growing demand for Internet of Things devices, the need for customized hardware solutions is increasing rapidly
- IoT devices in particular require frequent hardware customization.

UC San Diego

Challenges

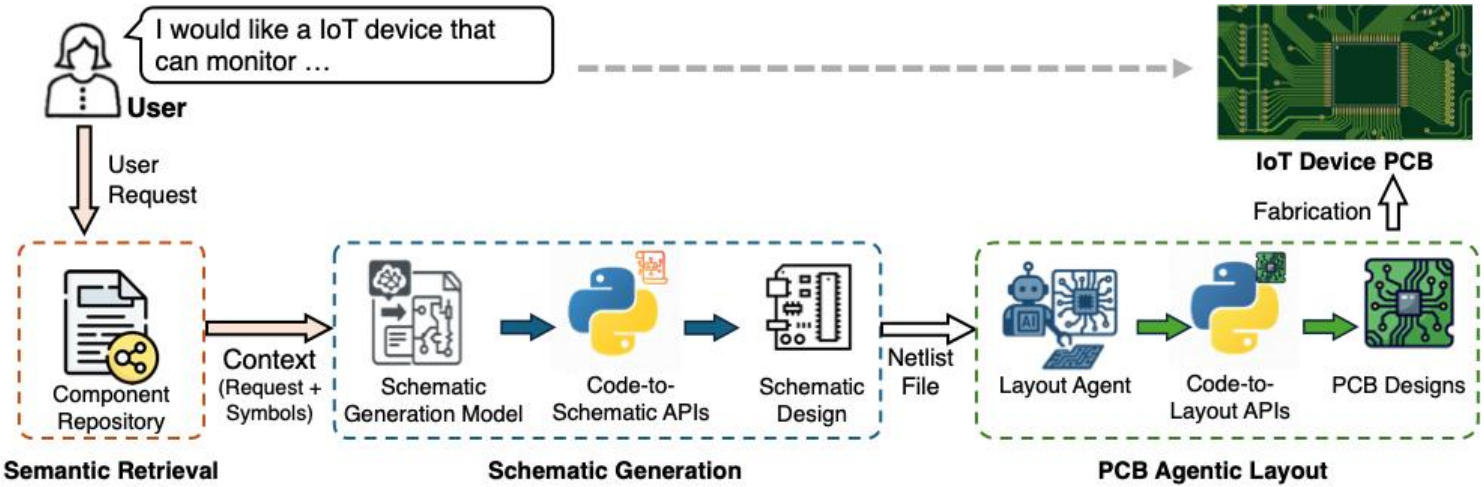


1. A significant gap between a design requirement and fabrication-ready PCB designs.
2. Today's IoT PCB design workflow is still GUI-centric and manual, demanding substantial domain expertise.

UC San Diego

System Design

Overview

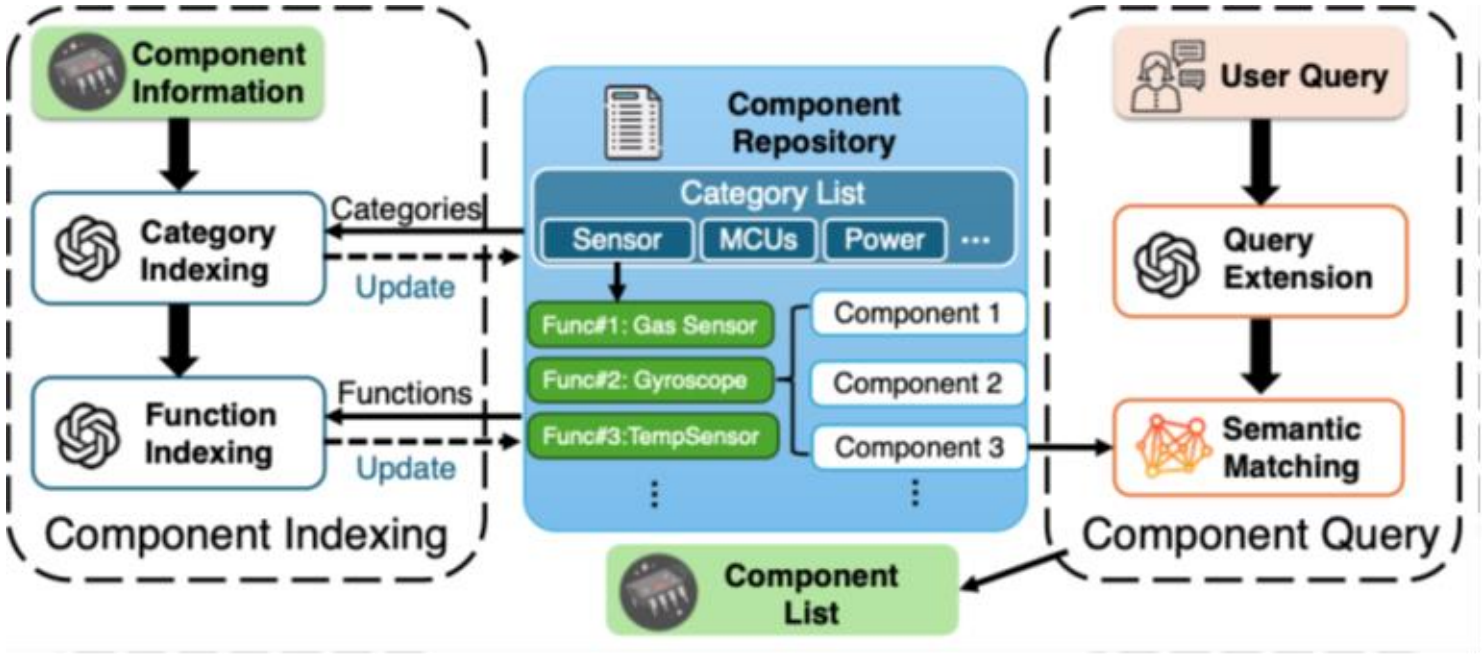


- General PCB design pipeline: Schematic→ Netlist→ Layout/Wiring→ Fab
- Schematic File: Component + Power + Net Label + Connection
- PCB Design File: Footprint + Metal Wires + Via holes

UC San Diego

System Design

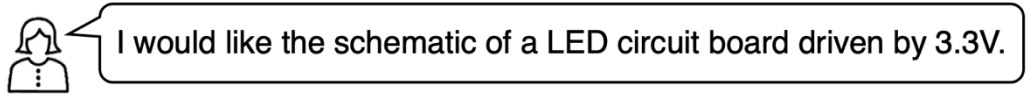
Semantic Retrieval



UC San Diego

System Design

Code Representation for schematic generation



LLM

```
(kicad_sch
(version 20231120)
(generator "eeschema")
(generator_version "8.0")
(paper "A4")
(lib_symbols
(symbol "Device:R"
(pin_numbers hide)
(pin_names …
```

Text Representation

❌ Uninterpretable
❌ Format Error

LLM + Image Generation

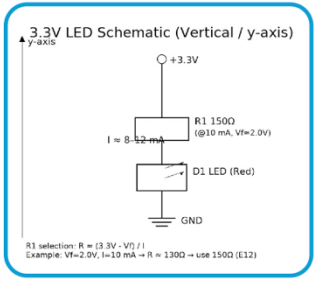
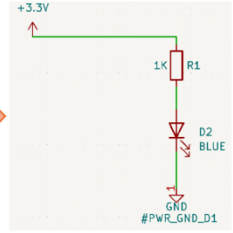


Image Representation

❌ Unsupported Format
❌ Prone to Error

```
add_schematic_symbol(
symbol_lib="Device",
symbol_name="LED",…)
…
connect_pins("D2", "K",
"#PWR_GND_D1", "")
```



Code Representation

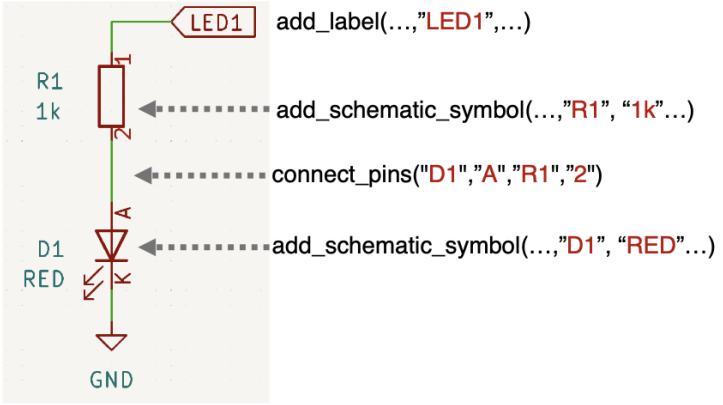
✅ Readable and Robust
✅ Suitable for LLM-based generation

UC San Diego

System Design

Code Representation for schematic generation

- Inspired by the observation of human engineers, we represent the schematic with functions called.
- Design Logic:
  - First, put the central symbol.
  - Second, other symbols and labels
  - Finally, connections
- Key Observations:
  - A series of editing operations
  - Placement regarding a local reference/coordinates

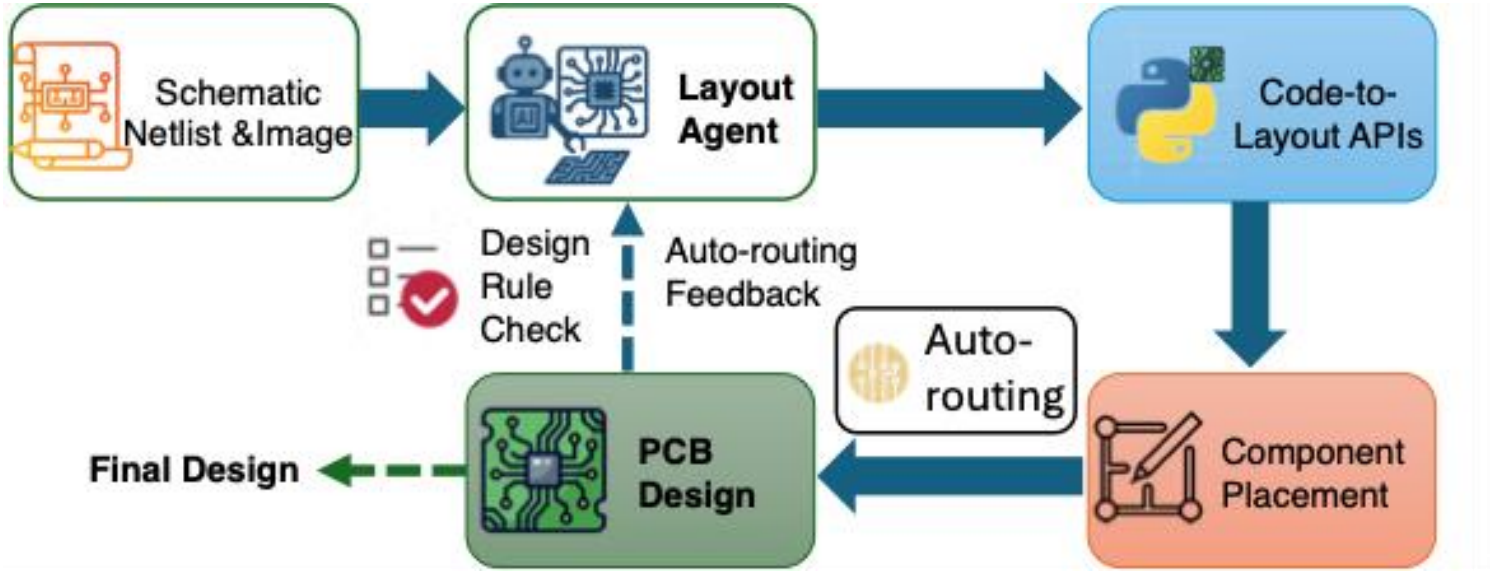



```
def add_schematic_symbol(symbol_lib, symbol_name, x, y, ref, value, rotation, mirror):
def add_label(label_pos, label_text, label_ref, label_type, text_orient):
def get_pin_location(symbol_reference, pin_name):
def connect_pins(symbol_ref_a, pin_name_a, symbol_ref_b, pin_name_b):
def write_out_all_wires():
```


UC San Diego

System Design

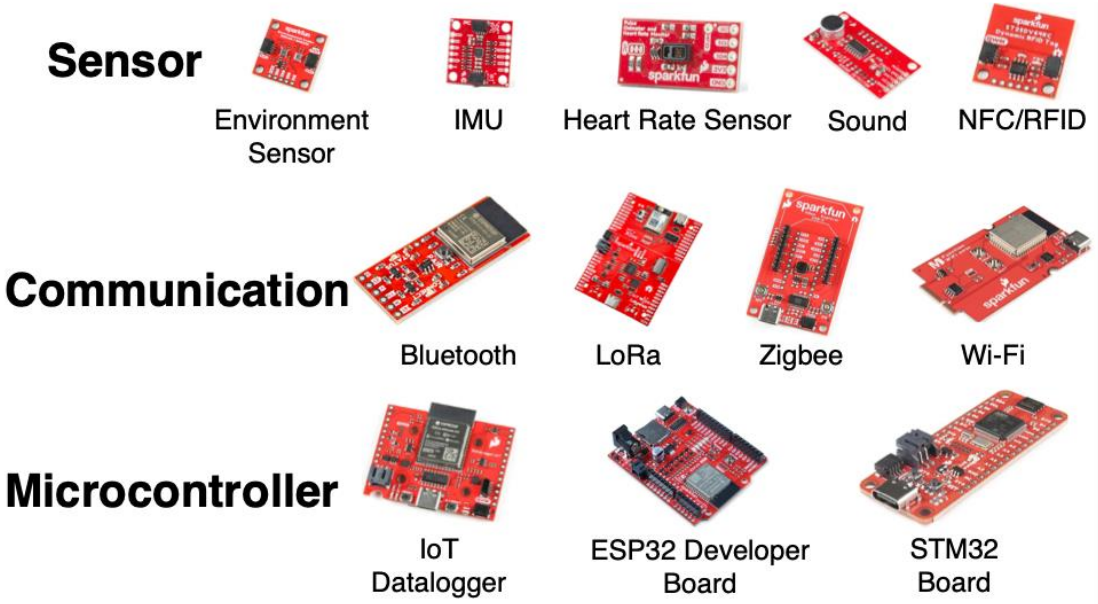
Agentic Layout



UC San Diego

Evaluation

Dataset and Setup

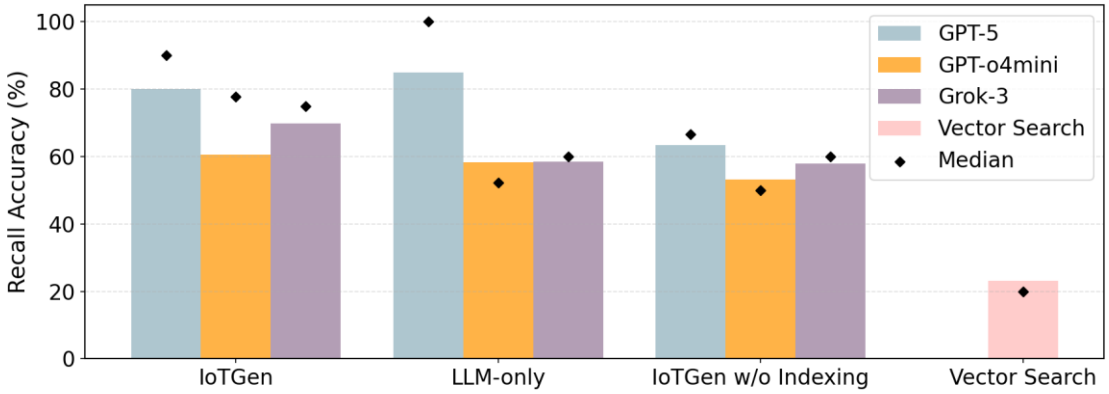


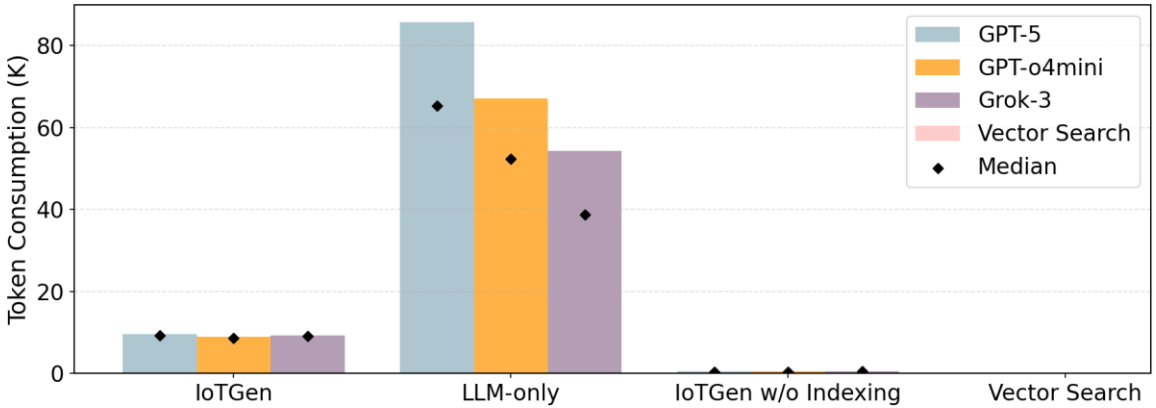
- Dataset: 196 KiCAD full-device schematic designs, spanning various types of functionality, covering microcontroller, analog modules, LED, power, storage, battery, antenna, USB, etc.
- We finetune the base model of *GPT-oss-20b* with LoRA on a single 80 GB Nvidia A100 GPU for schematic generation.
- For the semantic symbol retrieval and agentic PCB layout, we adopt commercial LLM access.

UC San Diego

Evaluation

Semantic Retrieval





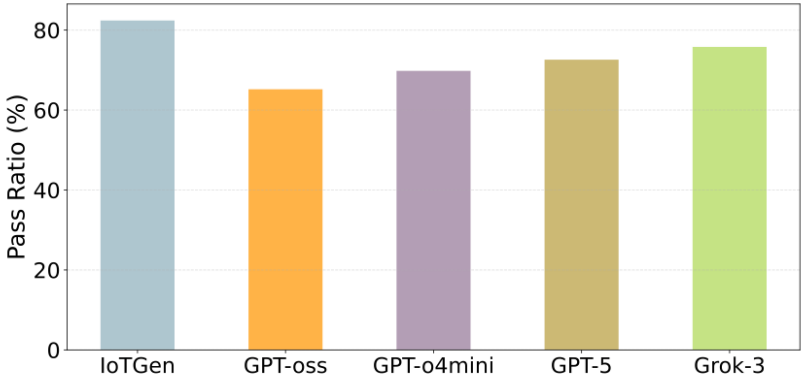
90% accuracy with only 10% token consumption compared to LLM-only

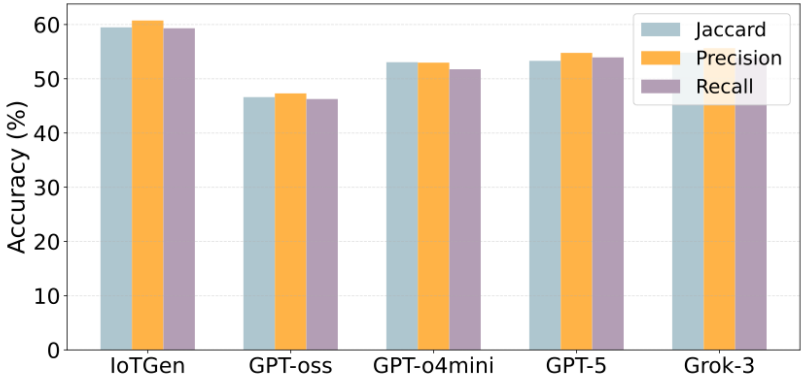
UC San Diego

Evaluation

Schematic Generation

- IoTGen outperforms other models regarding ratio of valid circuits and netlist accuracy.
- IoTGen contains a much smaller model compared to those by commercial services.

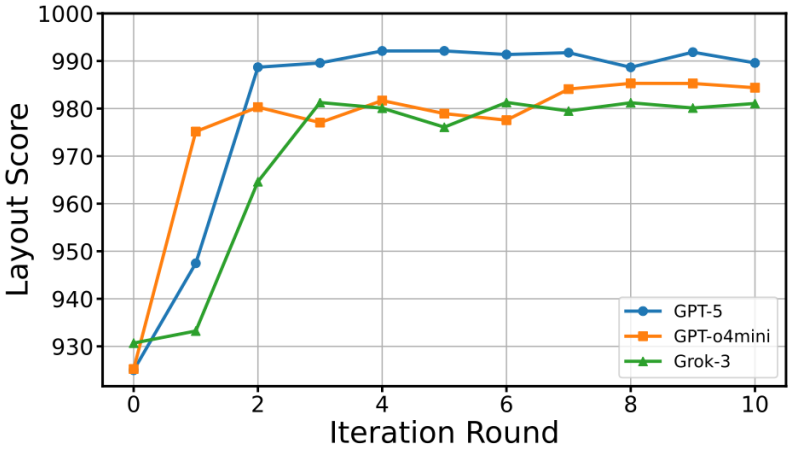


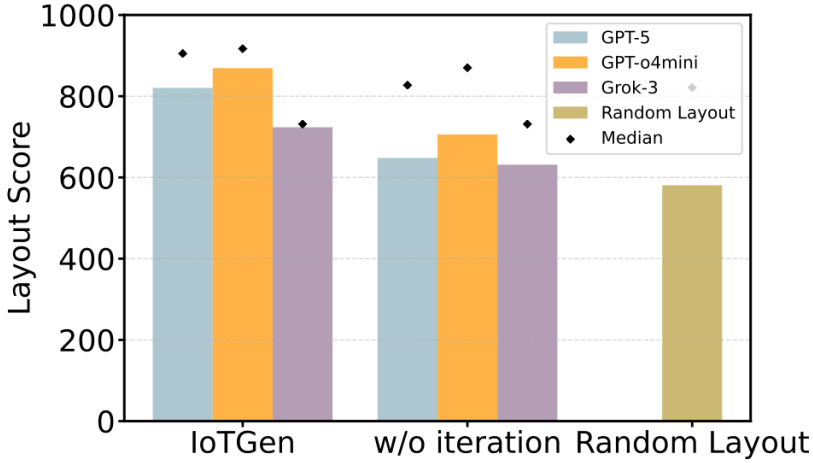


UC San Diego

Evaluation

Agentic Layout





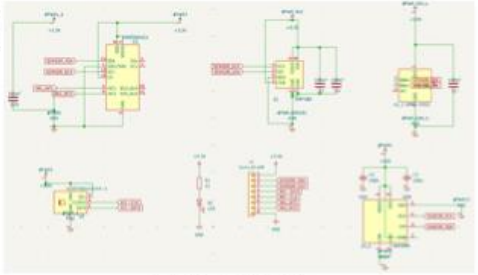
- GPT-5 is the best model for iterative agentic layout.
- IoTGen outperforms other two microbenchmarks.

UC San Diego

Evaluation

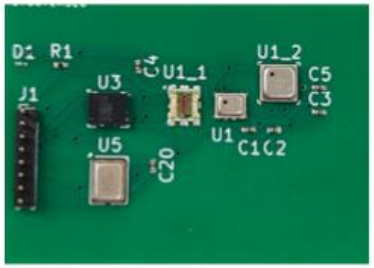
Case Study

(i)



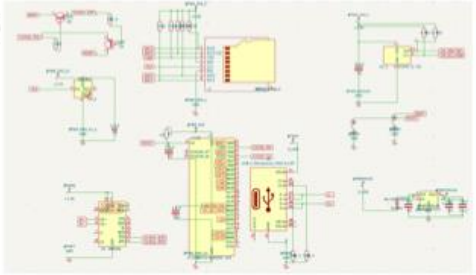
(a) Schematic

(b) PCB Layout

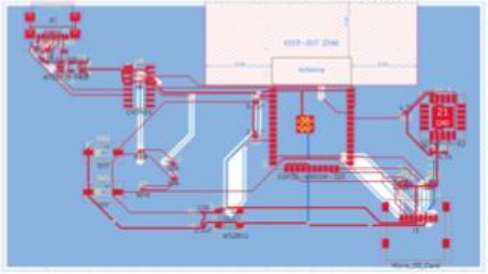


(c) Fabricated PCB

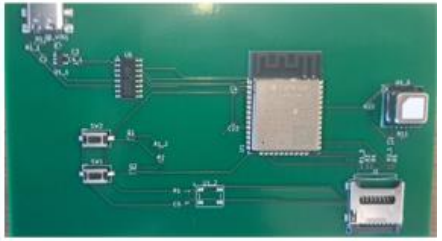
(ii)



(d) Schematic



(e) PCB Layout



(f) Fabricated PCB

Top: Sensor Module
Bottom: CO2 Monitor

UC San Diego

Conclusion

- IoTGen automates the end-to-end PCB design workflow, translating natural-language intent into fabrication-ready IoT hardware.
- It integrates semantic component retrieval, schematic generation, and agentic PCB layout in a unified framework.
- For more details, please check our paper. Our codebase, model and dataset are available at: https://github.com/luoqinpei/IoTGen.git

IoTGen

UC San Diego