



GPIoT: Tailoring Small Language Models for IoT Program Synthesis and Development

Leming Shen, Qiang Yang, Xinyu Huang, Zijing Ma, **Yuanqing Zheng**

♠ The Hong Kong Polytechnic University,

♠ University of Cambridge

SENSYS '25

Presented by Lili Pan

作者团队

研究方向

- Mobile/Edge Computing
- Wireless Networking,
- Ubiquitous Computing
- AIoT Applications
- Embedded AI



沈乐明

香港理工大学在读博士



郑元庆

香港理工大学教授

1. [MOBICOM'25] **Leming Shen**, Qiang Yang, **Yuanqing Zheng**, Mo Li, “**AutoIoT**: LLM-Driven Automated Natural Language Programming for AIoT Applications” , In ACM MobiCom, 2025, (PDF).
2. [SENSYS'25] **Leming Shen**, Qiang Yang, Xinyu Huang, Zijing Ma, **Yuanqing Zheng**, “**GPIoT**: Tailoring Small Language Models for IoT Program Synthesis” , In ACM SenSys, 2025, (PDF).
3. [MOBISYS'24] Leming Shen, Qiang Yang, Kaiyan Cui, Yuanqing Zheng, Xiao-Yong Wei, Jianwei Liu, Jinsong Han, “FedConv: A Learning-on-Model Paradigm for Heterogeneous Federated Clients” , In ACM MobiSys, 2024, (PDF).

Outline

1

Background

2

Related work

3

Design

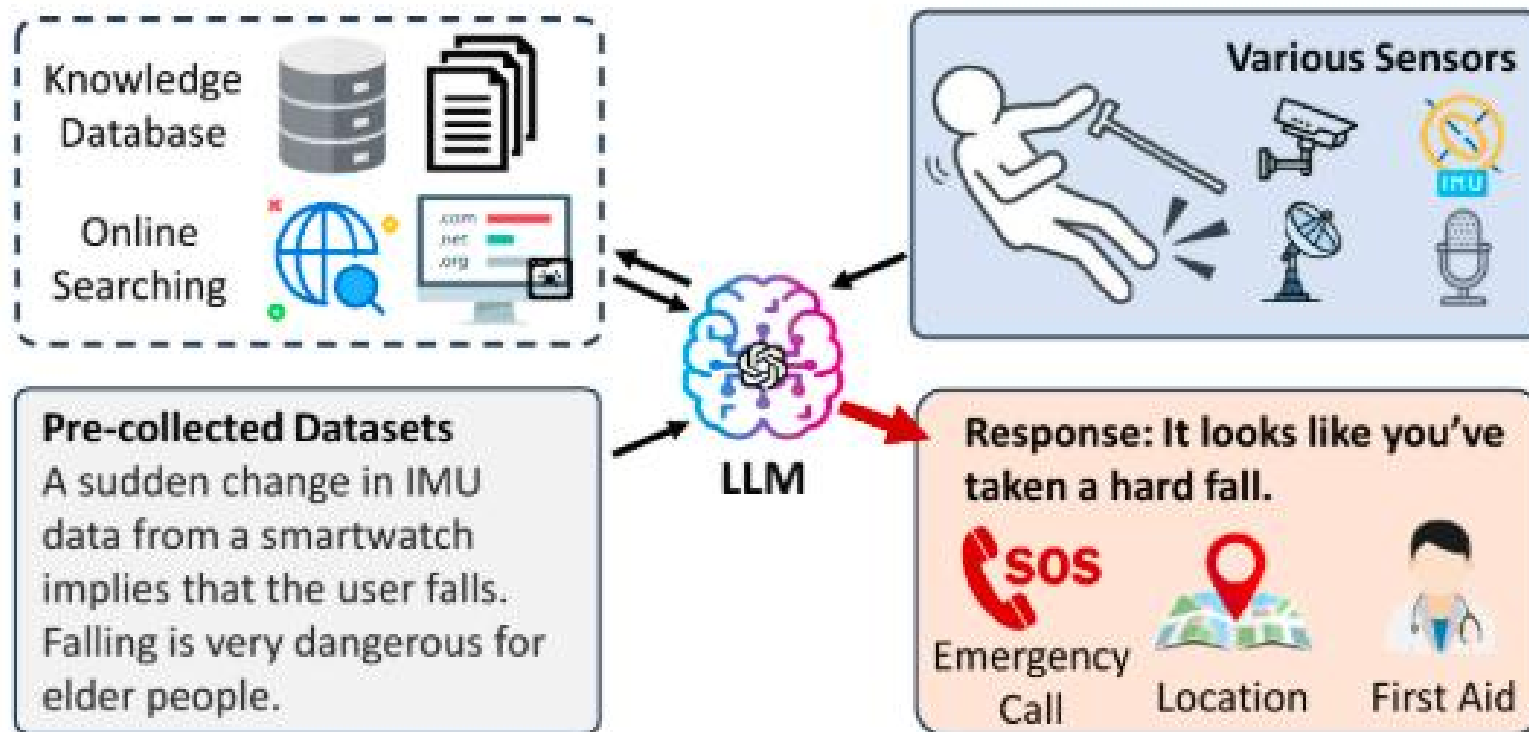
4

Experiments

5

Conclusion

AIoT场景



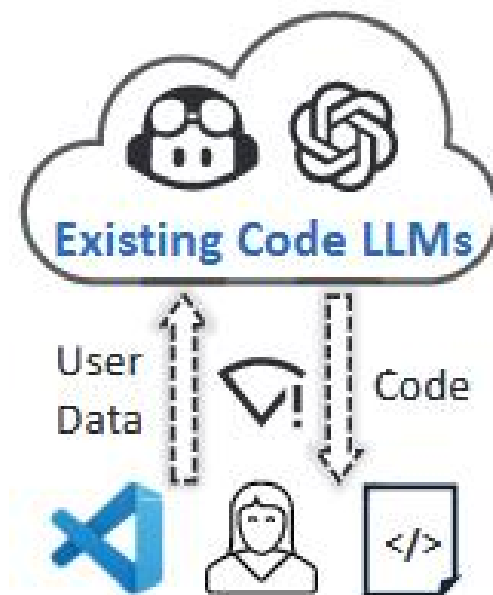
LLM 如何在 AIoT 应用中感知物理世界并与之交互的图示

Penetrative AI--渗透性AI

LLMs用于代码生成

Large Language Models (LLMs)

- LLMs 改变了我们与 AI 的互动方式
- 卓越的自然语言理解能力
- 在代码生成有广泛的应用前景



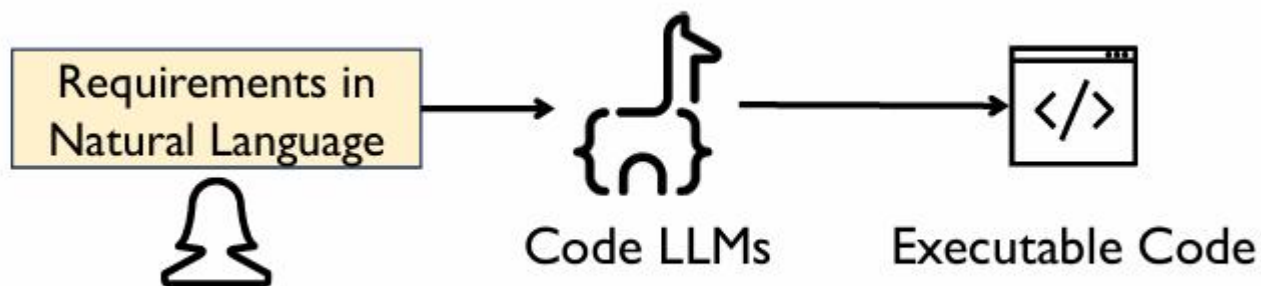
- 代码补全
- 代码生成
- 缺陷检测
- 文档编写



ChatGPT
(OpenAI)



DeepSeek-
Coder



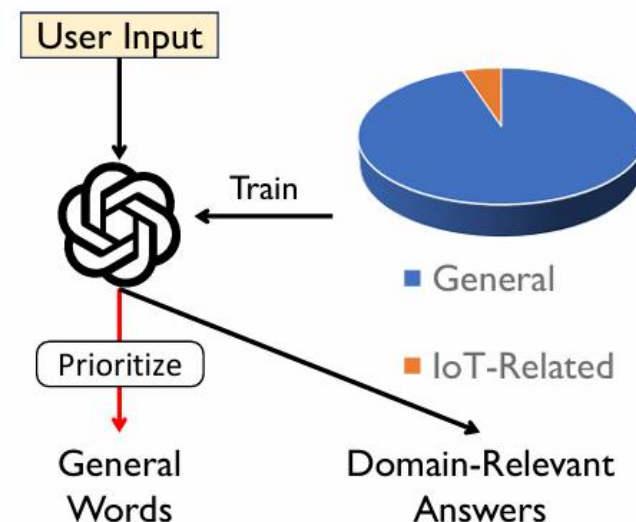
通用LLMs在AIoT场景的局限性?

Question?

当面对需要**特殊领域知识**的物联网应用时，
现有的代码LLMs只能提供**通用但不一定适合**的解决方案。

Why?

- 传统LLMs专注于**通用编程**任务，而不是针对任何特定领域进行训练。
- 物联网相关的知识和程序在编码LLMs的数据集上只占**很小**的比例。
- 在推理过程中，**物联网术语**将被赋予较低的优先级，生成的代码将更专注于通用领域。



Our goals:

- ✓能否构建专为 IoT 应用程序代码生成任务量身定制的代码 LLM?

Outline

1

Background

2

Related work

3

Design

4

Experiments

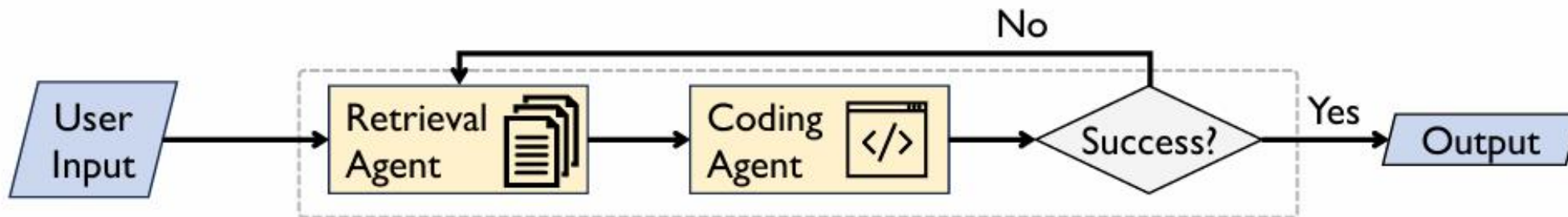
5

Conclusion

相关工作

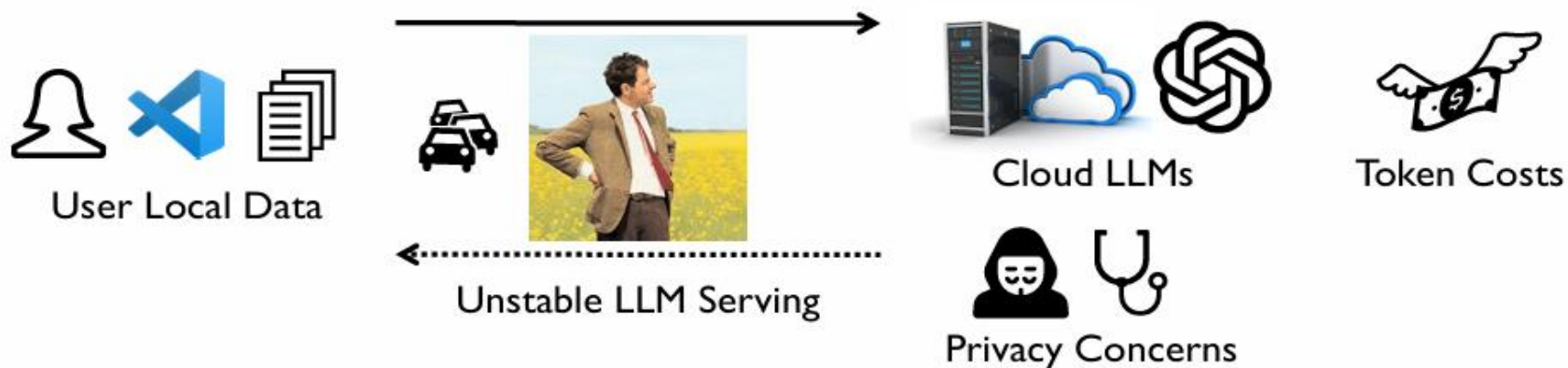
检索增强生成 (RAG) LLM+RAG

- 检索增强生成 (RAG) 可以为 LLMs 提供检索到的领域知识，以增强上下文相关性



L1:

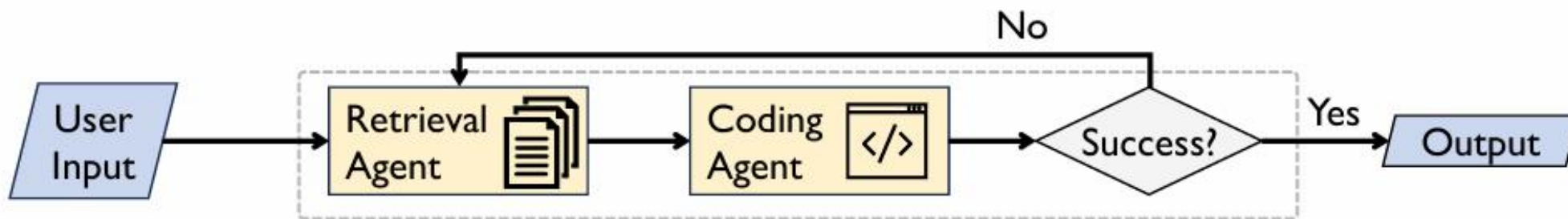
- 需要一个具有**强大语言理解能力**的 LLM 来从检索到的知识中学习。
- 云端LLM（例如 GPT-4）可能会遇到**网络条件差、成本高和隐私问题**，而本地 LLM（例如 Llama2-70b）对**系统资源**（例如内存和网络）有苛刻的要求。



相关工作

检索增强生成 (RAG) LLM+RAG

- 检索增强生成 (RAG) 可以为 LLMs 提供检索到的领域知识，以增强上下文相关性



L2:

- 复杂的 RAG 设计是必不可少的，需确保检索到的知识的正确性和高度相关性。
- 否则，LLM 可能无法专注于 IoT 上下文，并且仍然提供通用解决方案。

L3:

- 需要精心设计的提示，以确保输出必须严格遵循预定义的格式。
- 由于 LLM 的幻觉和不可靠性，这是极具挑战性的。

Outline

1

Background

2

Related work

3

Design

4

Experiments

5

Conclusion

Why SLM?

- SLM 体积更小，可以本地部署
- 在 IoT 专用数据集上优化 SLM
- 优化数据集，增强稳定性避免幻觉

3个SLMs (Small Language Models)

- 任务分解 SLM (TDSLM)
- 需求转换 SLM (RTSLM)
- 代码生成 SLM (CGSLM)

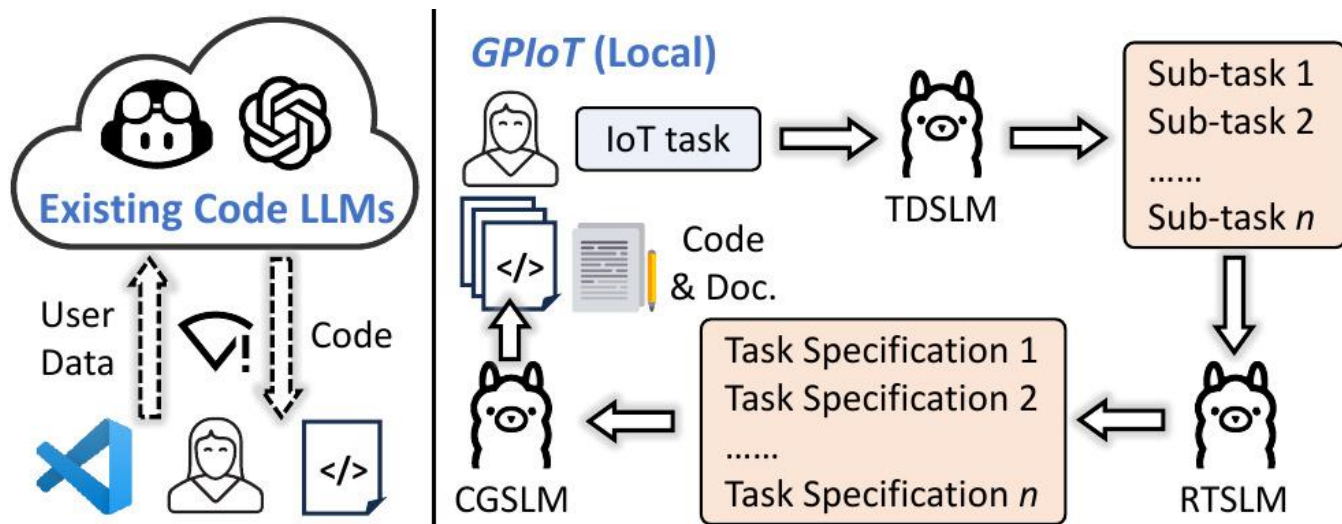


图 1：现有代码 LLM 需要将敏感数据传输到远程服务器
GPIoT 具有三个本地 SLM 来保护用户隐私并降低查询成本

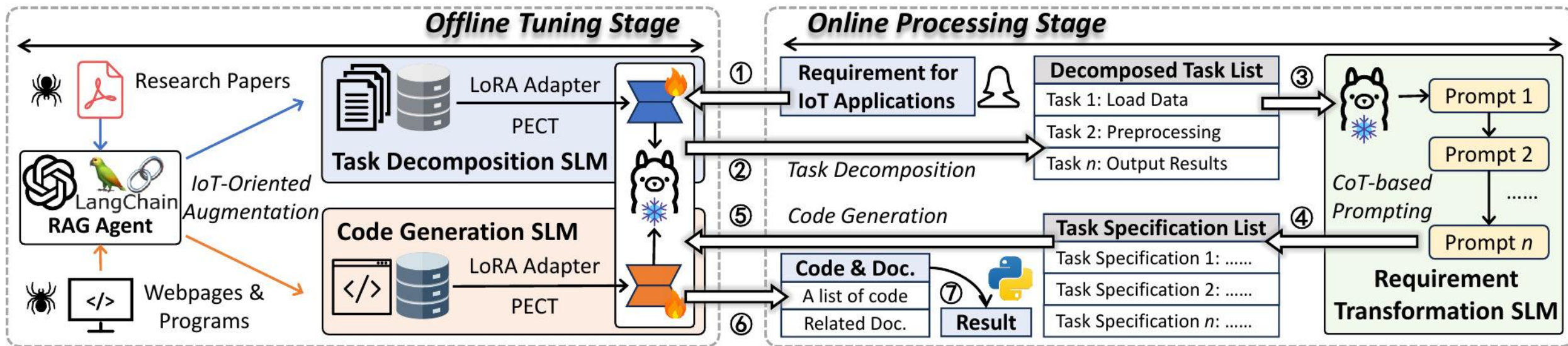


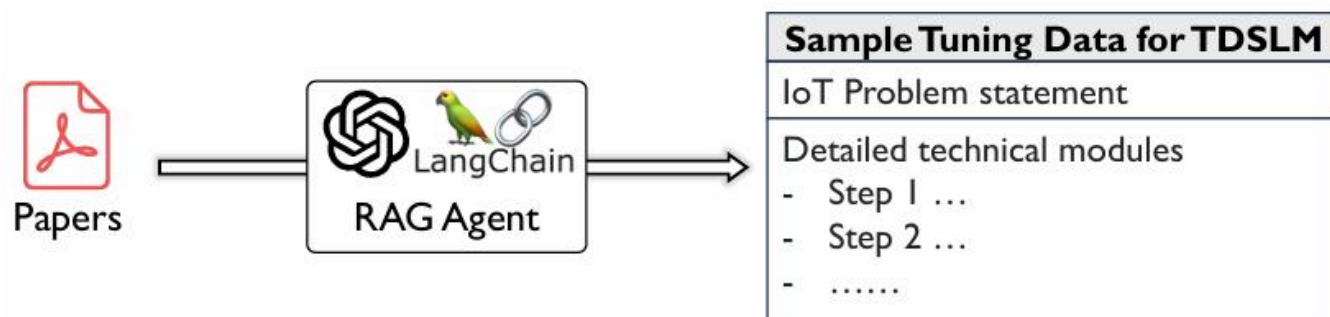
图 4: GPIoT 的系统概述和工作流程

- 离线阶段：离线调优阶段构建了两个物联网专用数据集，微调TDSLM和CGSLM
- 在线阶段：在线阶段根据物联网应用开发的用户需求，合成物联网专用程序

C1: 缺乏高质量数据

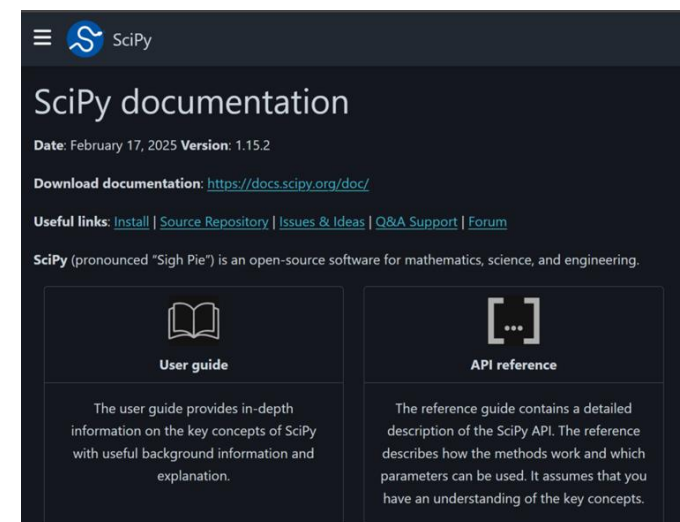
- 据我们所知，目前没有面向物联网的文本生成数据集：

- TDSLM: 用户需求 → 子任务
- CGSLM: 子任务 → 程序



- 解决方案

- 任务分解数据集 (TDD)
 - TDD 包含36,098 个 “问题描述→分解任务” 的成对数据
 - 数据来源：与物联网相关的论文
- 代码生成数据集 (CGD)
 - CGD 包含35,419 个 “任务规范→代码与文档” 的配对
 - 数据来源：与物联网相关的 Python 包 (e.g. SciPy)



- 数据集构建
- 任务分解数据集 (TDD)
 - 数据收集、数据格式化和数据增强
- 代码生成数据集 (CGD)
 - 数据收集、目标多样性感知增强

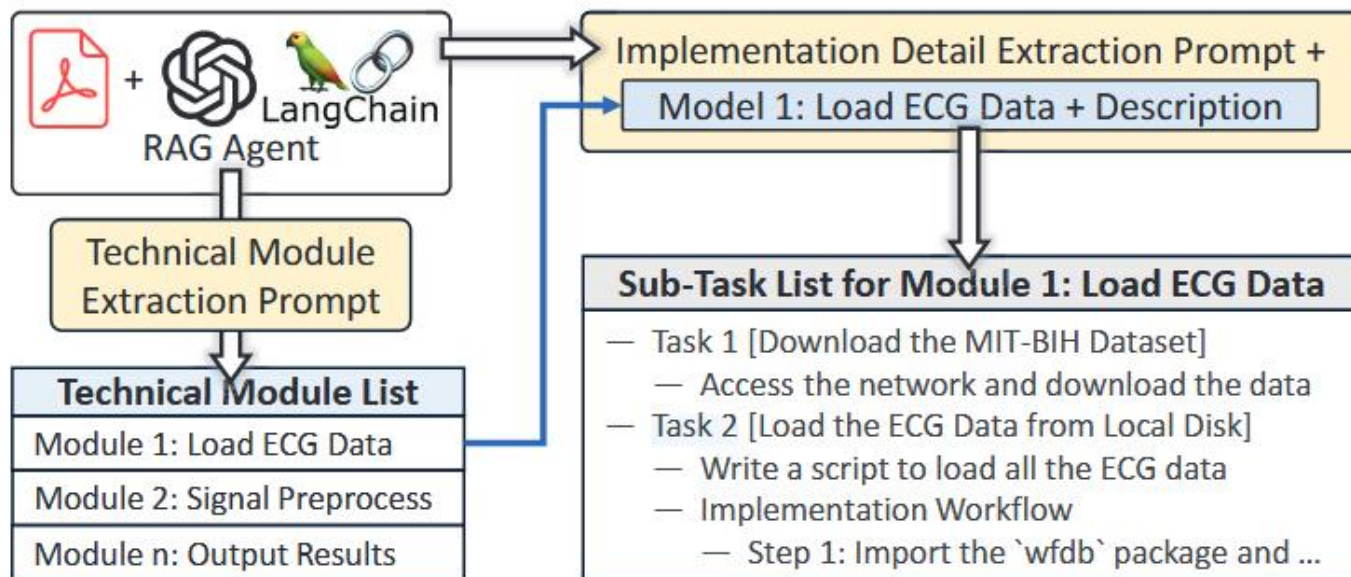






图 5：任务分解数据集构建

Technical Module Extraction Prompt	
Context Document: PDF File	 
System Message	
Based on the document, split the system proposed in the paper into multiple modules with detailed descriptions. You must search through the document repeatedly for detailed information. The output must be in the following Markdown format:	
<ul style="list-style-type: none"> - Module 1: module name + description - Module 2: module name + description 	

(a) Technical module extraction

Implementation Detail Extraction Prompt	
Context Document: PDF File	 
System Message	
Based on the document, summarize a problem statement of the given technical module and split it into a sub-task list with implementation details. You must search through the entire document thoroughly. The output must be in the format of:	
<ul style="list-style-type: none"> - Sub-Task 1 [Task Name] - [Problem Statement] - Implementation step 1 	

(b) Implementation detail extraction

图 6：论文信息提取的prompt Q-A对

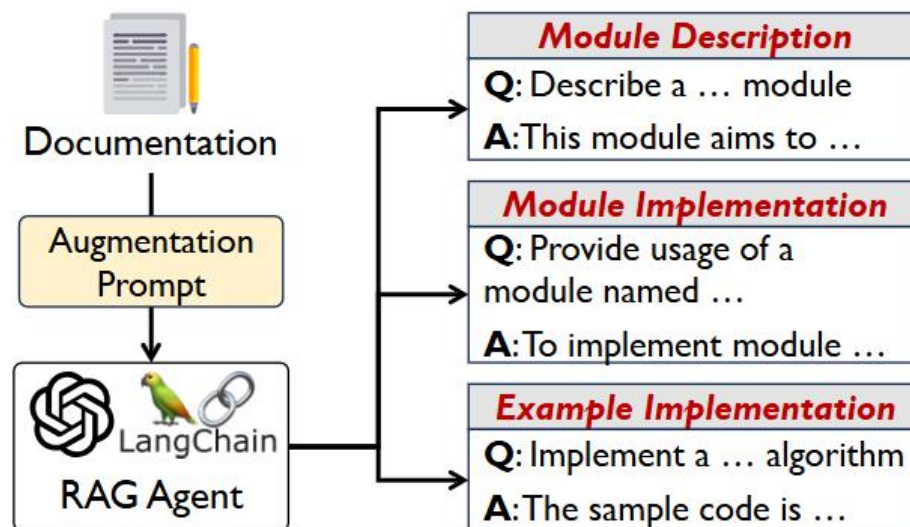
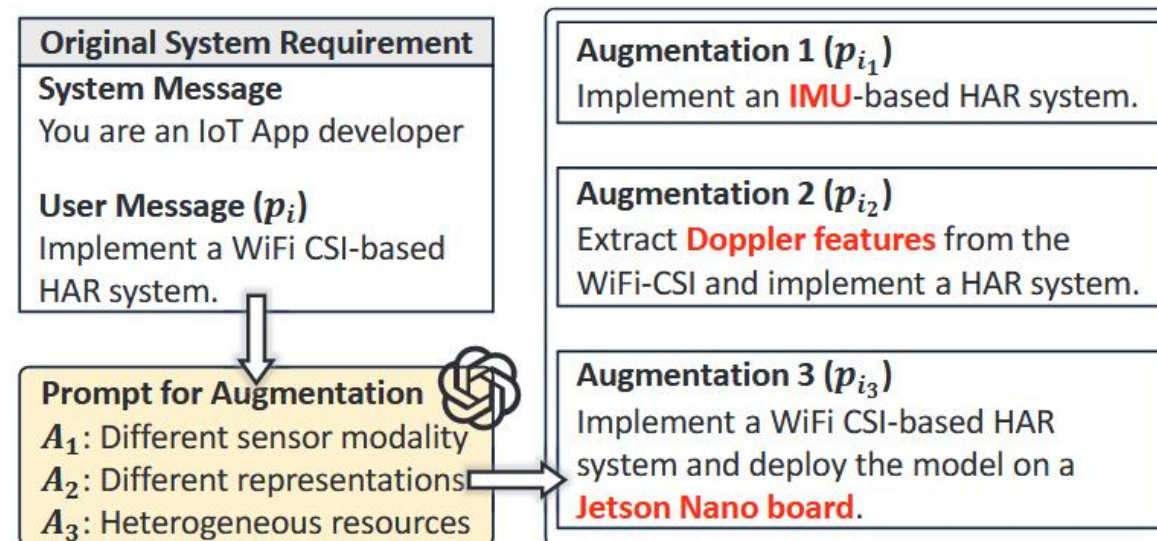
C1: 缺乏高质量数据

➤ 现有的数据增强方法在物联网领域无效?

• 面向 IoT 的**数据增强**用于 TDD

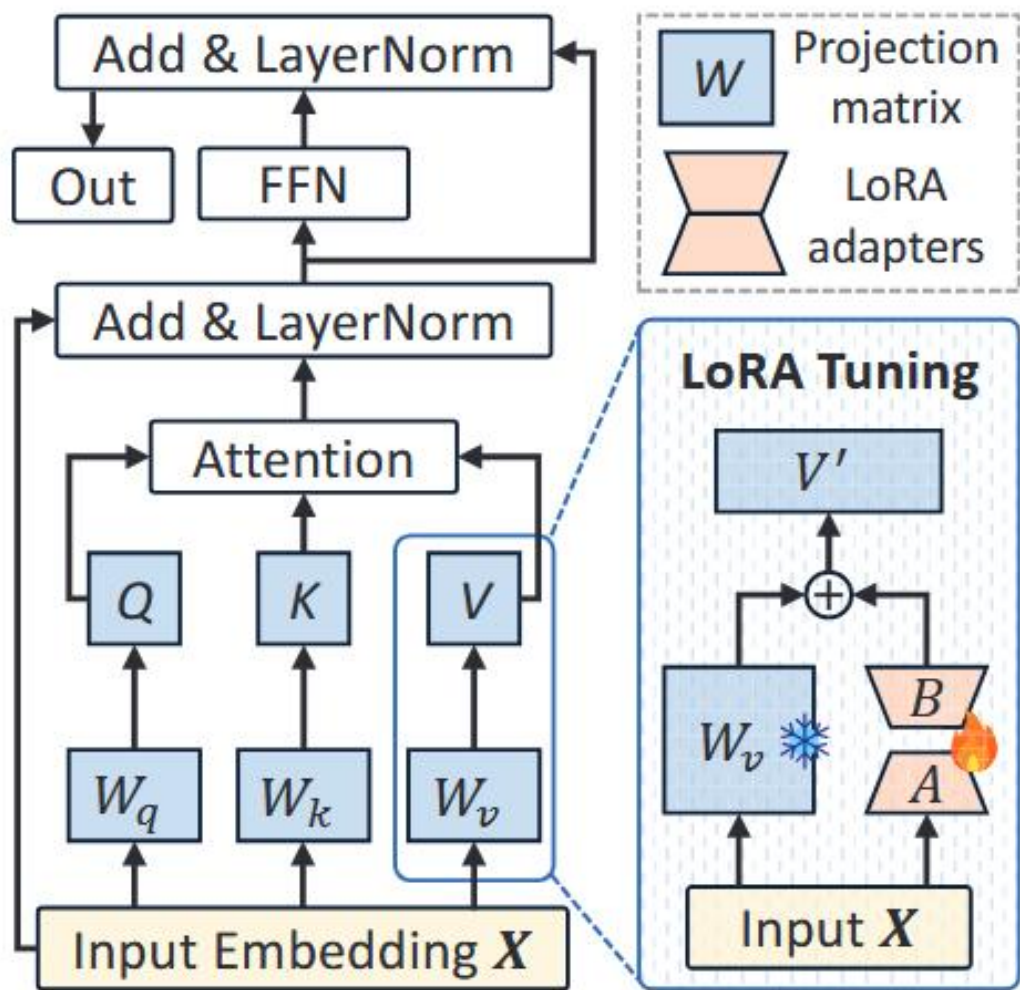
- 传感器模态
- 数据表示
- 系统资源

• 目标多样性**感知增强**用于 CGD

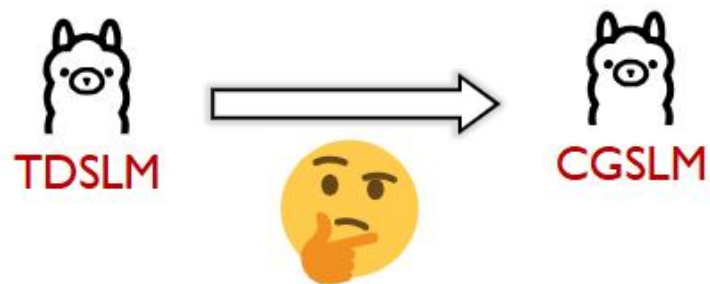


$$\mathcal{D}'_c = \mathcal{D}_1\{t_i \rightarrow m_{i_1}\} \cup \mathcal{D}_2\{t_i \rightarrow (c_i, d_i)\} \cup \mathcal{D}_3\{t_j \rightarrow (c_j, d_j)\}$$

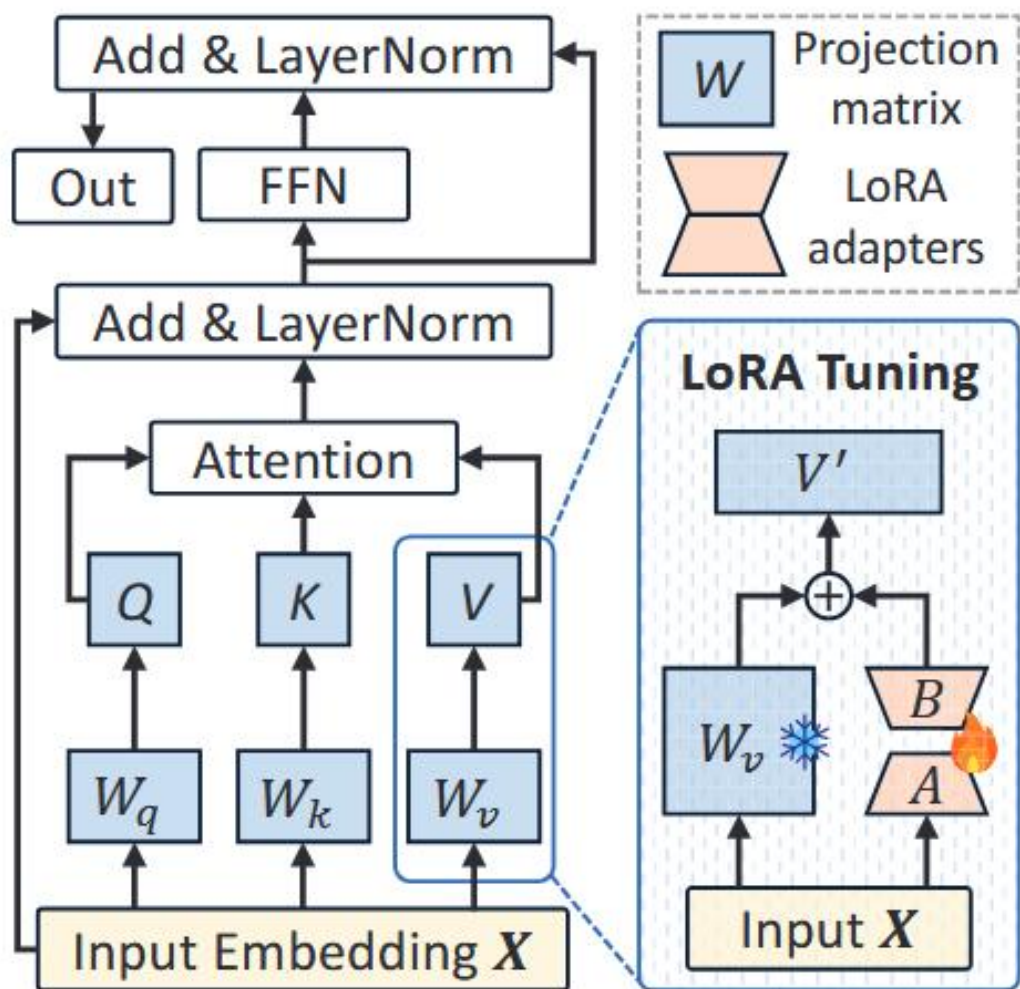
微调小模型TDSLM和CGSLM



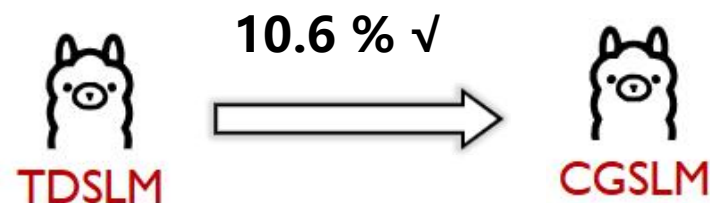
$$V' = (W_v + BA) \cdot X$$



微调小模型TDSLML和CGSLM

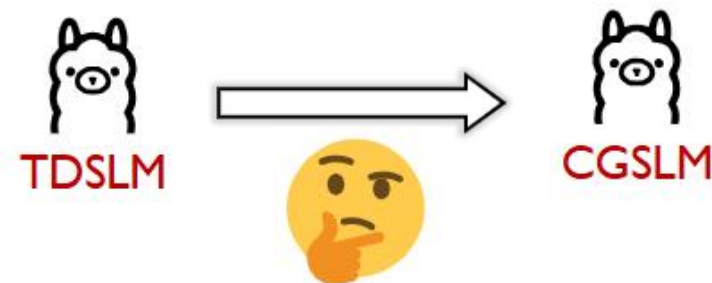


- 任务分解数据集 (TDD)
 - TDD 包含36,098 个 “问题描述→分解任务” 的成对数据
 - 数据来源：与物联网相关的论文
- 代码生成数据集 (CGD)
 - CGD 包含35,419 个 “任务规范→代码与文档” 的配对
 - 数据来源：与物联网相关的 Python 包 (e.g. SciPy)

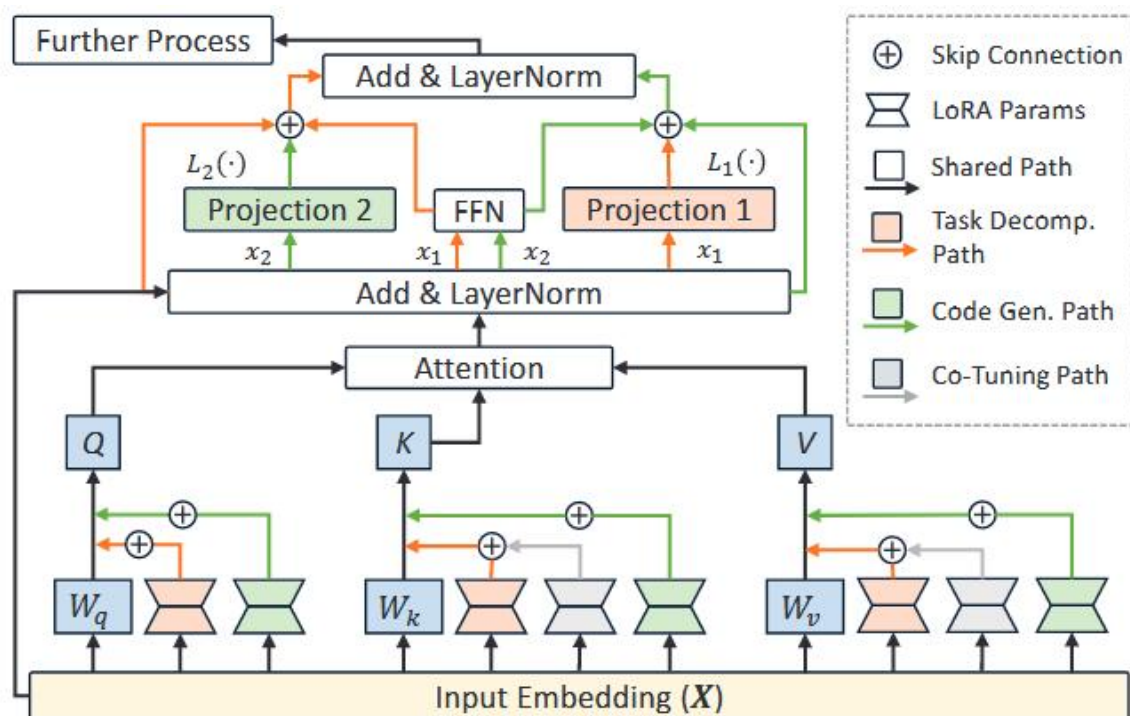


C2: 领域错位

- 两种SLM在微调过程中发挥了不同领域的特长，产生的知识并不一致
- TDSLM的输出可能落在CGSLM所能处理的范围之外



M2: PECT 参数高效微调



C3: 格式不兼容

- TDSLM 的输出（分解任务）用自然语言描述
- CGSLM 的输入（任务规范）应该有固定结构

M3: 需求转换（RTSLM小模型）

- 通过 CoT提示链 构建结构化任务规范
- 将 TDSLM 的输出转换为规范格式

- 用户目标提取
- I/O 规范提取

Well-Structured Task Specification (Prompt)

****Target****

Write some code and documentation to create a universal non-uniform random number generator to sample random variates from a wide variety of univariate continuous and discrete distributions.

****Input Specification****

- numbers (arr): a stream of uniform random numbers.

****Output Specification****

- output (arr): random variates sampled from the specified distribution.

TDSLM Output

We need to first preprocess the raw ECG data by
Next, to enhance the QRS complex and suppress noises, we should adopt a set of filtering steps
Finally, the output should be ...



CGSLM Input

****Target****

Preprocess the ECG data

****Input****

- signal (array)

****Output****

- processed_signal (array)

Outline

1

Background

2

Related work

3

Design

4

Experiments

5

Conclusion

实验设置&物联网应用

- **硬件**

- 微调: NVIDIA A100 GPU
(80 GB)
- 推理: RTX 4090 (24 GB)

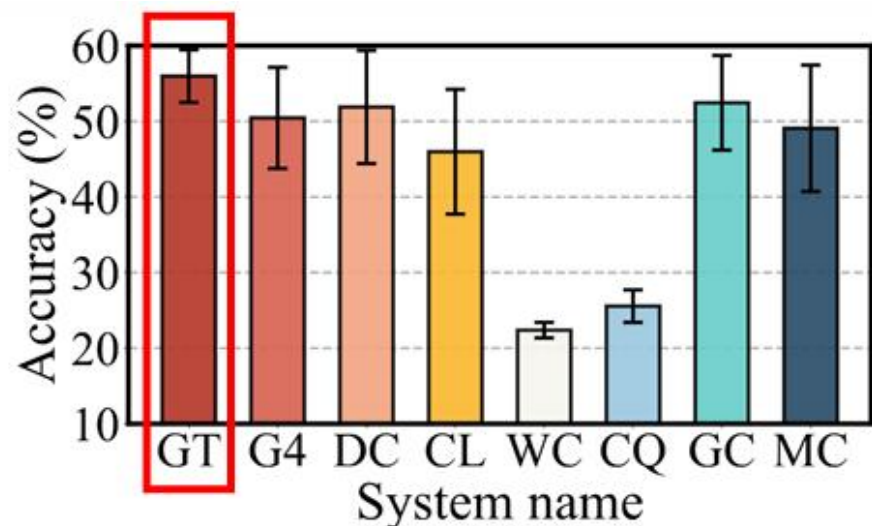
- **软件**

- 基础模型: Llama2-13b
- Agent: LangChain

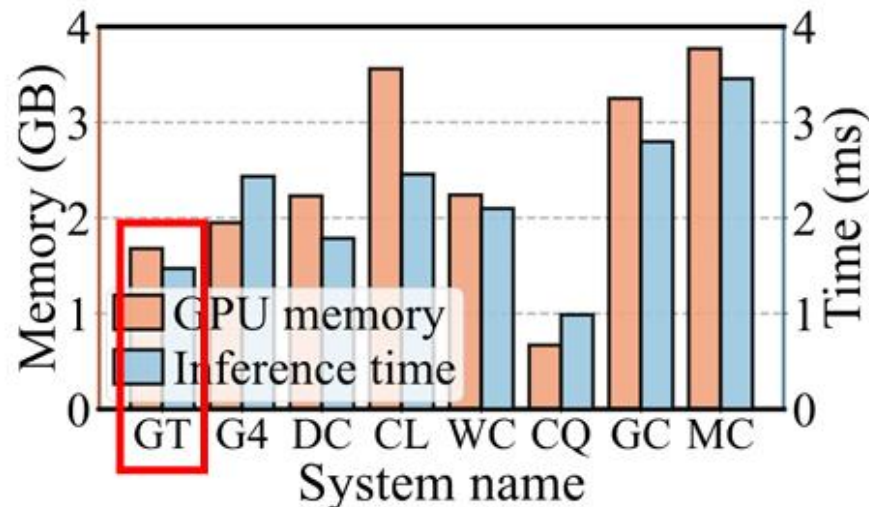
构建了一个基准测试 IoT Bench

- 心跳检测 (HD)
- 数据集: MIT-BIH
- 人类活动识别 (HAR)
- 数据集: WiAR
- 多模态 HAR
- 数据集: Harmony

多模态 HAR 实验评估



(a) Classification accuracy



(b) GPU memory & inference time

GT – GPIoT

G4 – GPT-4o

DC – DeepSeek-Coder

CL – CodeLlama

WC – WizardCoder

CQ – CodeQwen

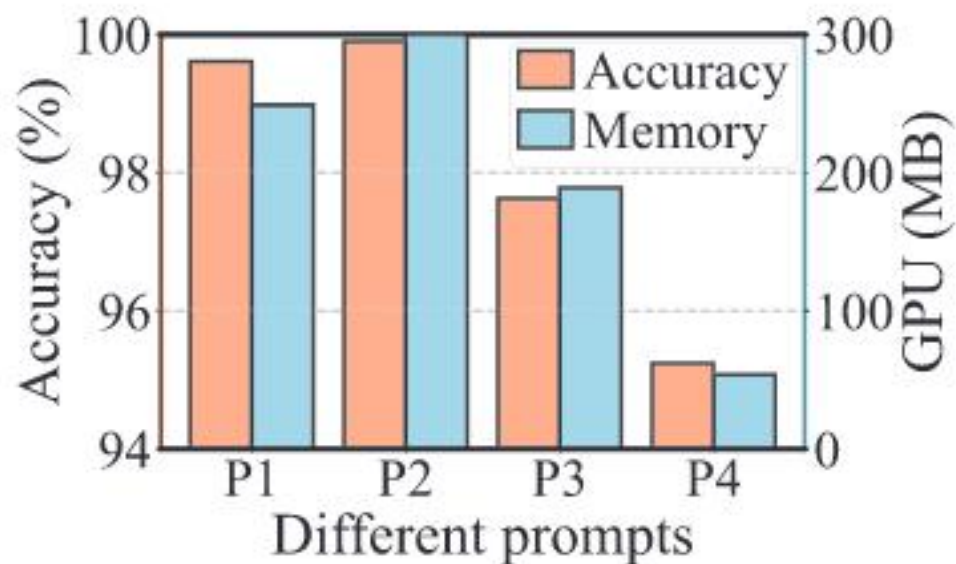
GC – GitHub Copilot

MC – MapCoder

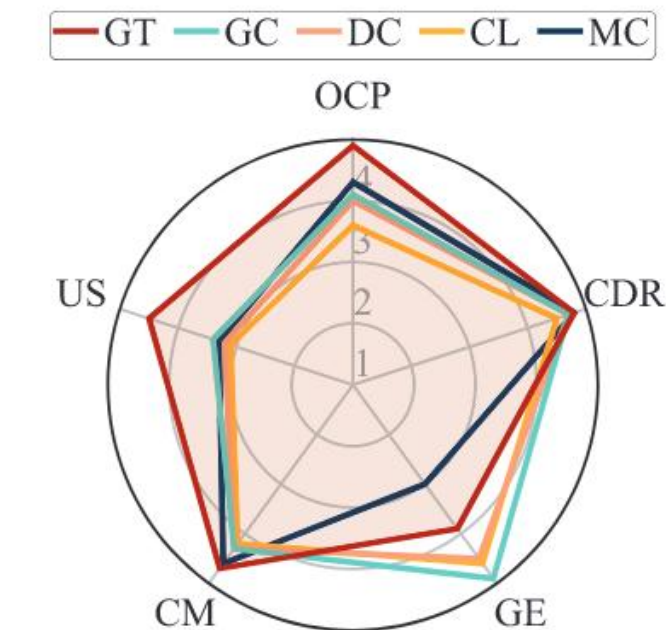
- 分类准确率13.44% ↑
- 适度的GPU内存和推理时间，鲁棒性 ↑ 波动 ↓
- 针对物联网数据的模型优化

GPIoT 可在调优过程中利用嵌入式物联网领域知识，整合更多物联网特定数据处理和模型优化算法。

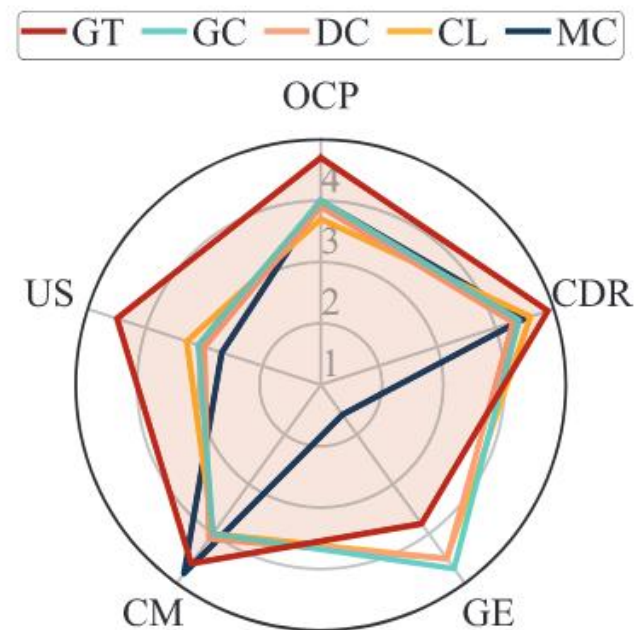
性能评估&用户研究



(a) Specifying resource constraints



(a) Signal processing-related tasks



(b) Machine learning-related tasks

(左图)：使用具有不同资源约束的提示来评估GPIoT的性能

(右图)：对于不同案例任务的用户研究

Outline

1

Background

2

Related work

3

Design

4

Experiments

5

Conclusion

Conclusion

- **Conclusion**

- 提出了GPIoT, 一个为物联网应用量身定制的本地代码生成系统
- 面向物联网的文本数据增强、PECT范式

- **Inspiration**

- 能不能进一步提高?

- 引入持续学习, 持续对本地 SLMs 进行微调
- 自动更新物联网知识数据库, 静态数据库--动态数据库

- 能不能用到我们的场景?

- 可以, 结合模拟仿真形成闭环, 对物联网代码生成系统进行持续更新

- 泛化性?



Q&A

2025.07.04