THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

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

Leming Shen[1], Qiang Yang[2], Xinyu Huang[1], Zijing Ma[1], Yuanqing Zheng[1]

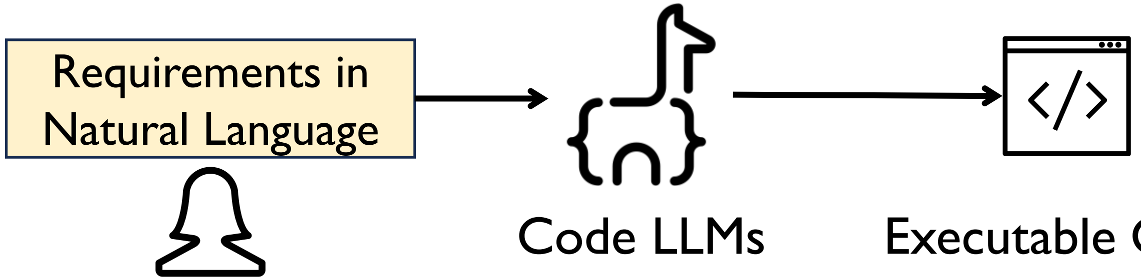
[1]The Hong Kong Polytechnic University, [2]University of Cambridge

Large Language Models (LLMs)

- LLMs revolutionize our interactions with AI
- Remarkable natural language understanding capabilities
- Promising applications: code generation

ChatGPT (OpenAI)

DeepSeek-Coder



Code LLMs

Executable Code

Gemini
2.0 Flash

Try Gemini Advanced

Please write some Python code to implement a R-peak detection algorithm using the MIT-BIH Arrhythmia dataset.

Python

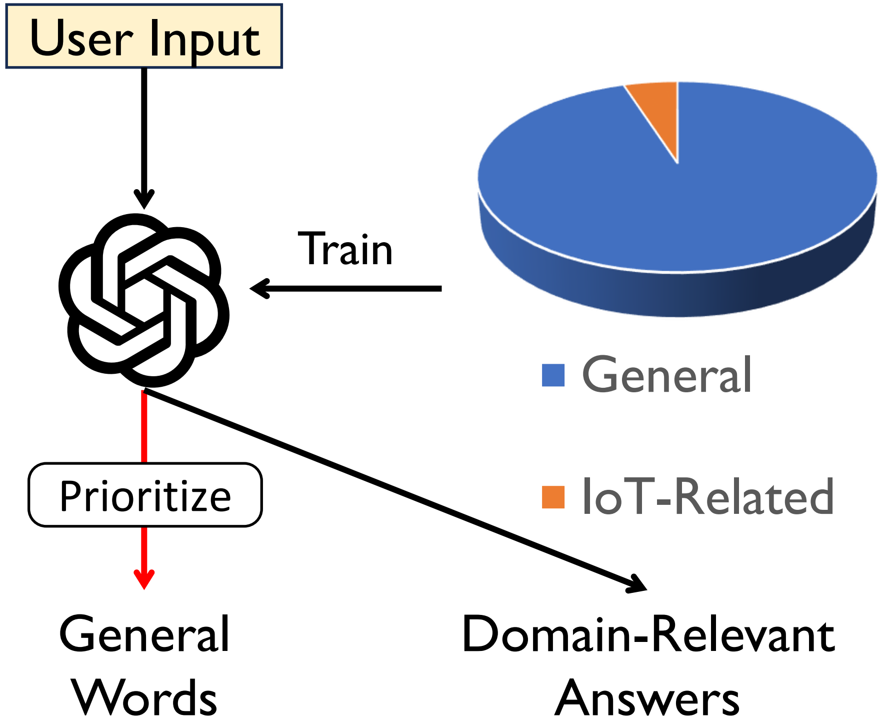
```
import numpy as np
import matplotlib.pyplot as plt
import wfdb
from scipy.signal import find_peaks

def r_peak_detection(ecg_signal, sampling_rate, window_size_sec=0.2, prominence_fac
    """
    Detects R-peaks in an ECG signal using a sliding window approach and prominence

    Args:
        ecg_signal (np.ndarray): The 1D ECG signal.
        sampling_rate (int): The sampling rate of the ECG signal in Hz.
        window_size_sec (float): The size of the sliding window in seconds.
        prominence_factor (float): Factor to adjust the prominence threshold.
                                   A higher value requires more prominent peaks.

    Returns:
        np.ndarray: Indices of the detected R-peaks.
    """
```

→ Sub-optimal Performance

Gemini can make mistakes, so double-check it

Why?

- Existing code LLMs are tasked with general-purpose code generation.
- IoT-related programs only occupy a small proportion of their training data.
- LLMs tend to prioritize and respond with more general words

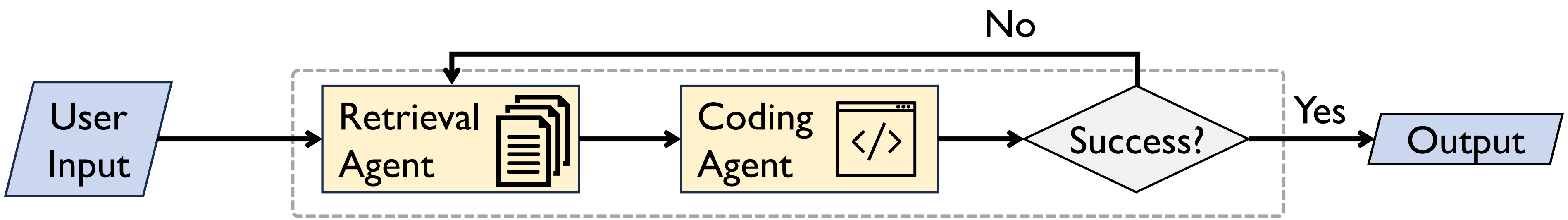


Research Question

Can we build a code LLM specially tailored for code generation tasks in the IoT domain?

A Promising Solution: LLM + RAG

- Retrieval-Augmented Generation (RAG) provides LLMs with retrieved domain knowledge to enhance context relevance

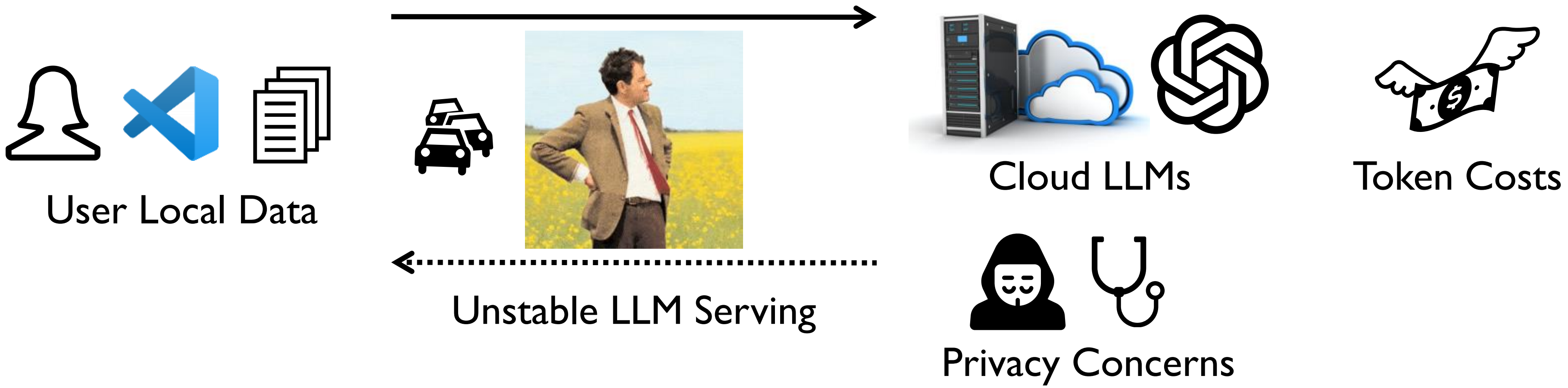


AutoIOT Agent [1]

[1] Shen, Leming, et al. "AutoIOT: LLM-Driven Automated Natural Language Programming for AIoT Applications." ACM MobiCom 2025

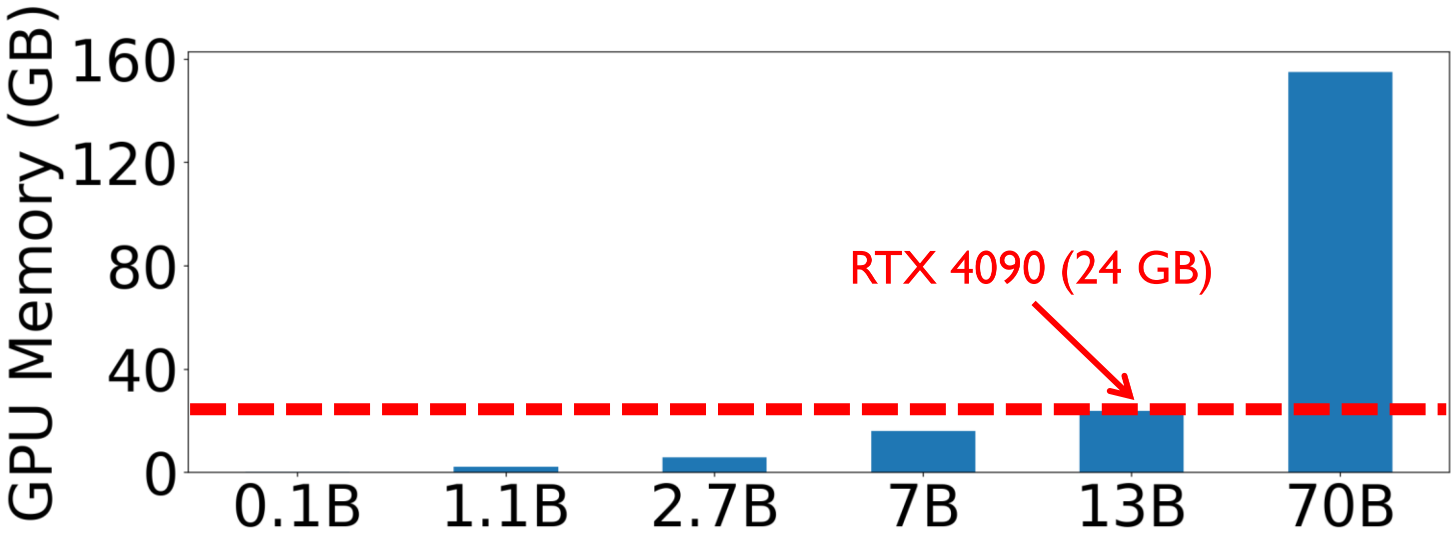
Limitations of LLM + RAG

- A powerful LLM with strong language understanding capability is required to learn from the retrieved knowledge.
  - Cloud LLMs (e.g., GPT-4)



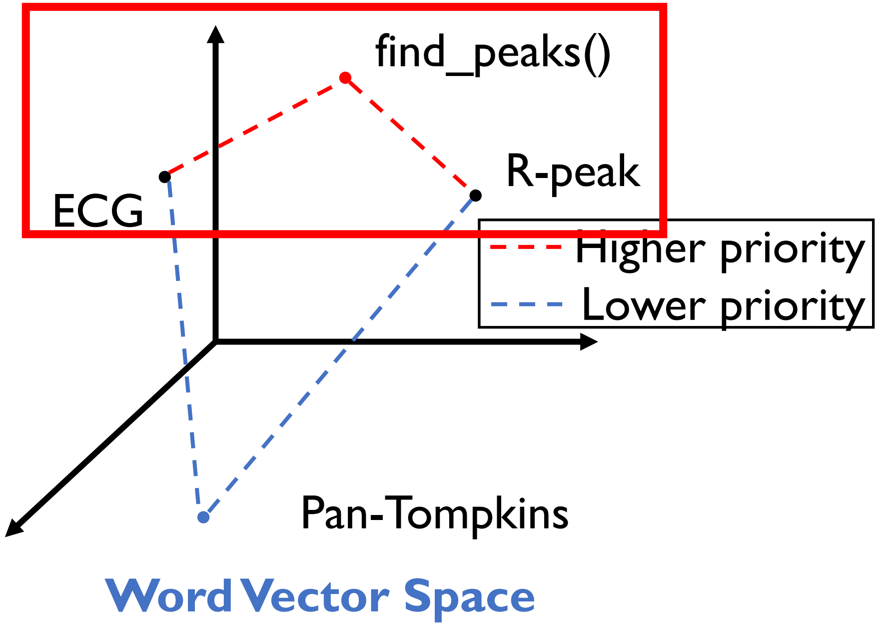
Limitations of LLM + RAG

- A powerful LLM with strong language comprehension capability is required to learn from the retrieved knowledge.
  - Cloud LLMs (e.g., GPT-4)
  - Large local LLMs (e.g., Llama2-70b requires around 150 GB GPU memory)



Limitations of LLM + RAG (Cont.)

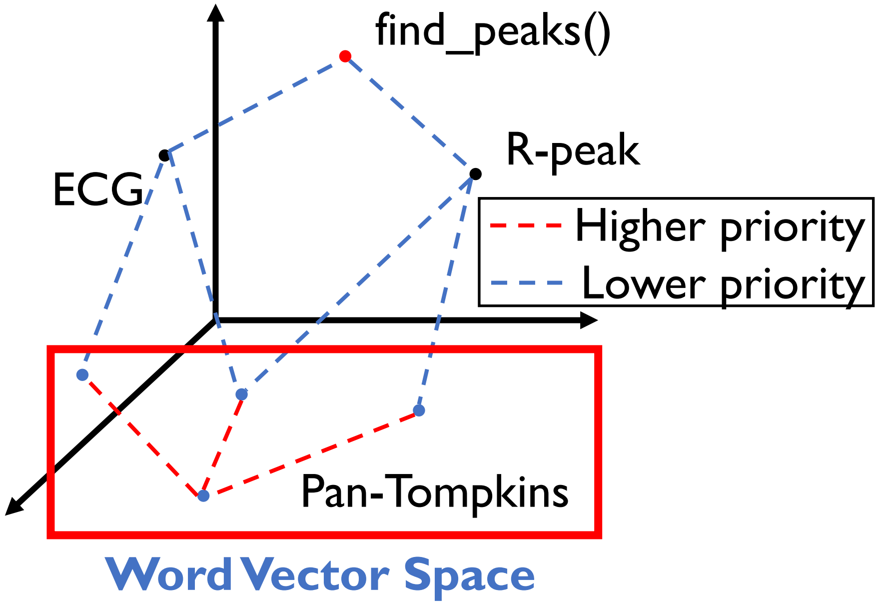
- Complicated RAG designs are mandatory to ensure the correctness and high relevance of the retrieved knowledge



User Input
Given ECG data, please implement a R-peak detection algorithm.

Limitations of LLM + RAG (Cont.)

- Complicated RAG designs are mandatory to ensure the correctness and high relevance of the retrieved knowledge



User Input
Given ECG data, please implement a R-peak detection algorithm.

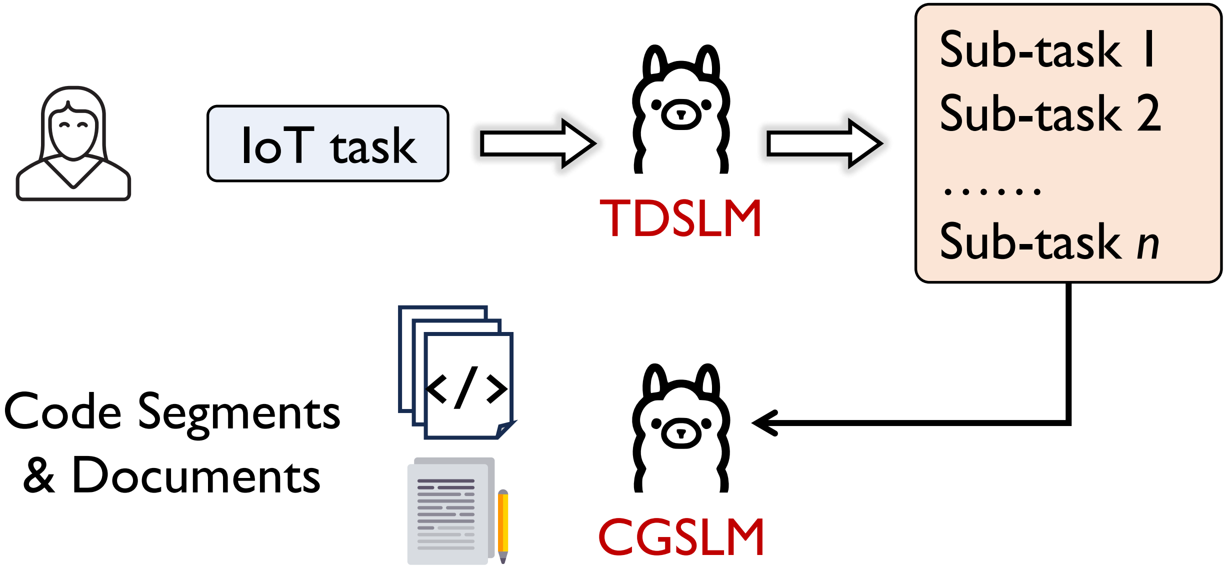
The retrieved knowledge should be highly relevant to the IoT context

GPIoT – Tuning Local SLM for IoT

	LLM + RAG	GPIoT (Tuned Local SLM)
System Overhead	Cloud: privacy, unstable network Local: high overhead	Low (local SLM) No privacy and serving issues
Response Relevance	High with dedicated and complex RAG prompts design	High relevance (tuned on IoT-specialized corpus datasets)
Stability	Unstable with accumulated errors	Structed responses

GPIoT – A Potential System

- Task Decomposition SLM (TDSLM)
- Code Generation SLM (CGSLM)



Technical Challenges

- **C1 – Lack of High-Quality Data**
- **S1 –** IoT Specialized Text-Generation Datasets

- **C2 – Domain Misalignment between SLMs**
- **S2 –** Parameter-Efficient Co-Tuning (**PECT**)

- **C3 – Format Incompatibility**
- **S3 –** Requirement Transformation

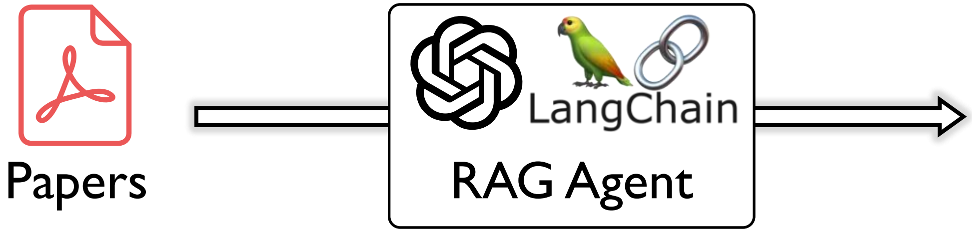
C1: Lack of High-Quality Data

- To our best, there are no IoT-oriented text generation datasets:
  - TDSLM: User requirement → Sub-task
  - CGSLM: Sub-task → Program

- **Solution**:
  - Task Decomposition Dataset (TDD)
  - Code Generation Dataset (CGD)

Task Decomposition Dataset (TDD)

- TDD contains pairs of "problem statement → decomposed tasks"
- Data source: IoT-related papers



Sample Tuning Data for TDSLM
IoT Problem statement
Detailed technical modules - Step 1 … - Step 2 … - ……

Limited quantity, quality, and diversity

IoT-Oriented Data Augmentation for TDD

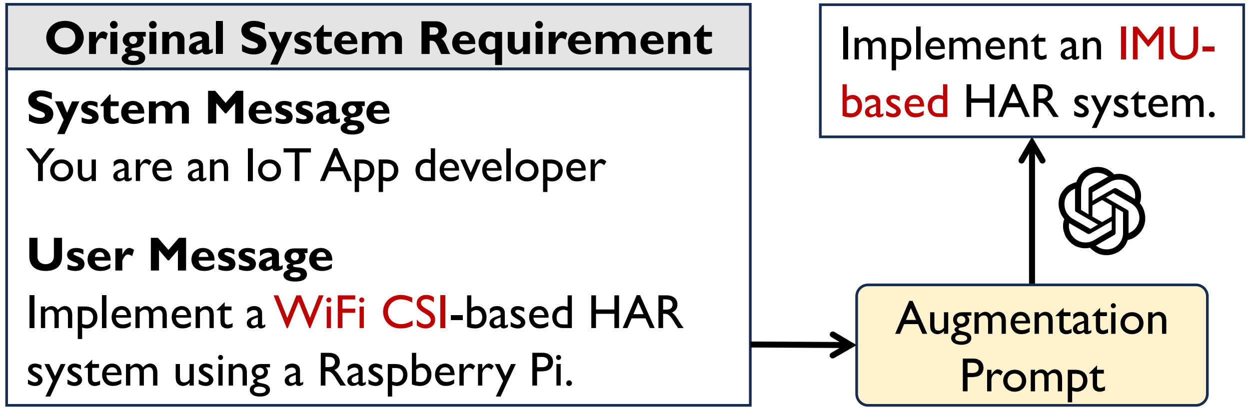
- Existing text augmentation methods focus on language features

- IoT-Oriented Data Augmentation:
  - Sensor Modality
  - Data Representation
  - System Resource

IoT-Oriented Data Augmentation for TDD

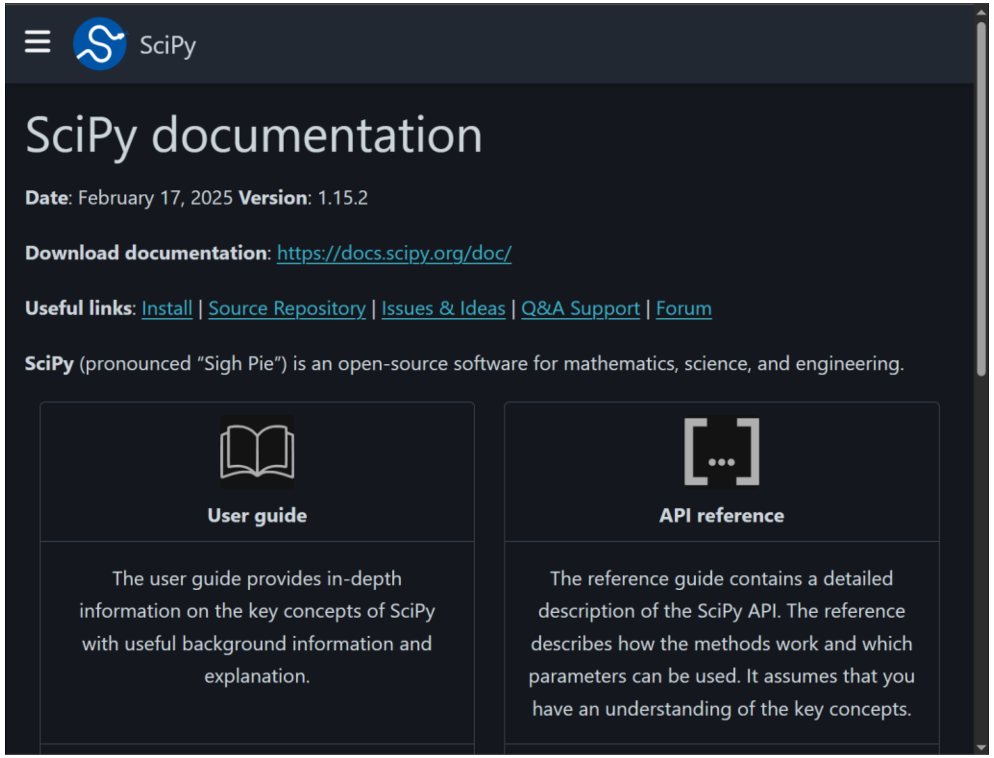
- Existing text augmentation methods focus on language features

- We consider:
  - Sensor Modality
  - Data Representation
  - System Resource



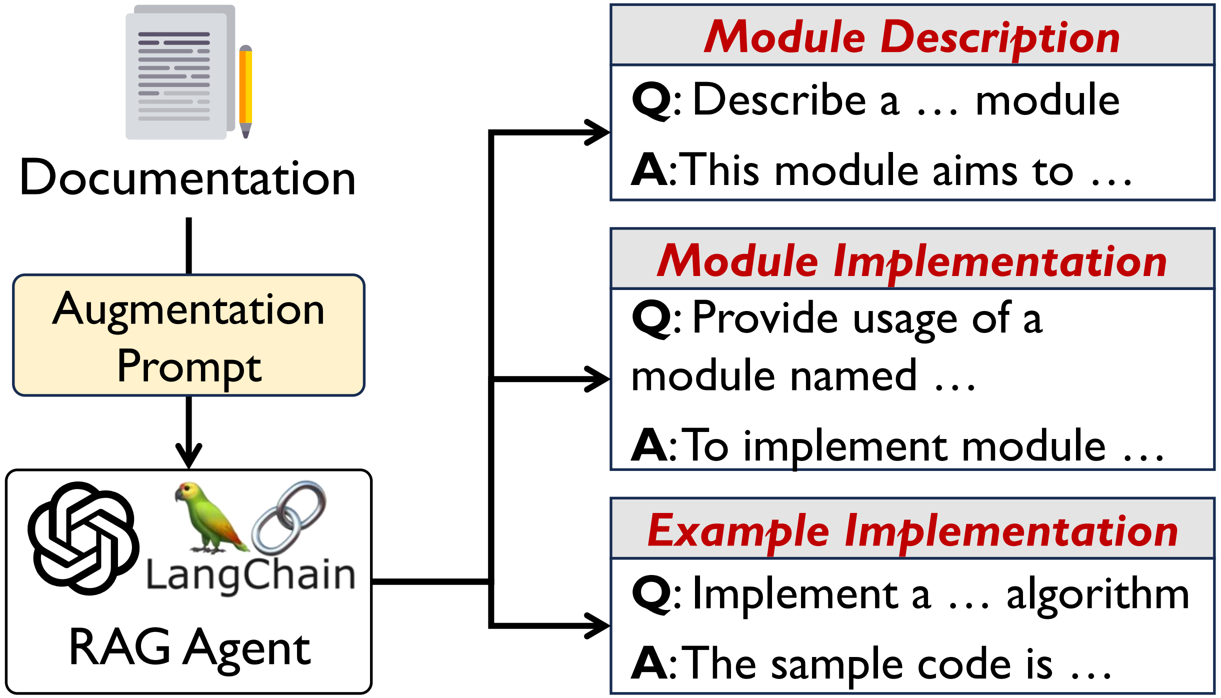
Code Generation Dataset (CGD)

- CGD contains pairs of "task specification → code & documentation"
- Data source: IoT-related Python packages (e.g., SciPy)



Code Generation Dataset (CGD)

- Target Diversity-Aware Augmentation



C1: Lack of High-Quality Data

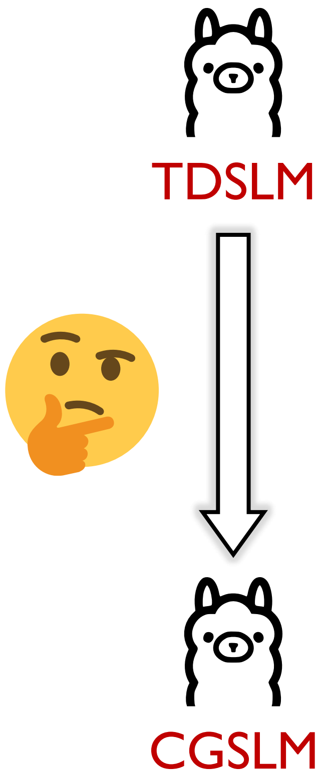
- We construct two IoT-specialized text-generation datasets
- We propose IoT-oriented augmentation methods

- TDD contains 36,098 "problem statement → decomposed tasks"
- CGD contains 35,419 "task specification → code & documentation"

C2: Domain Misalignment

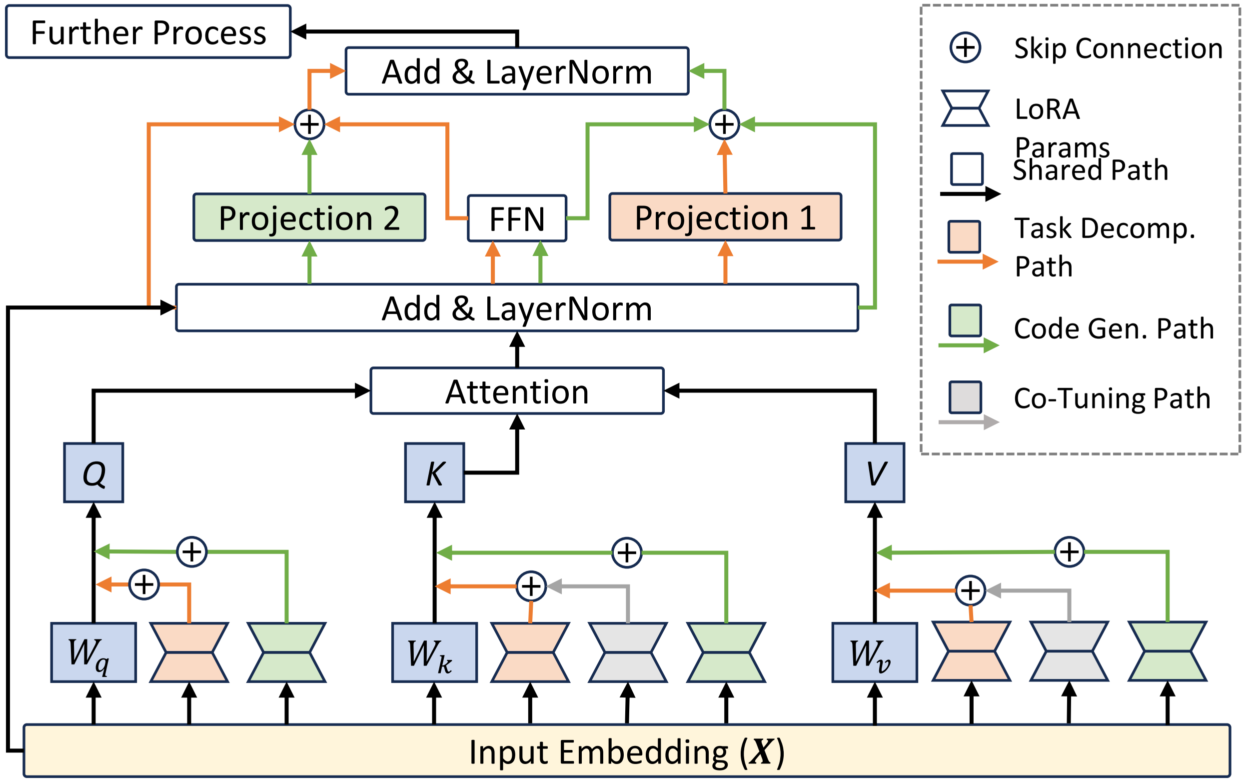
- The two SLMs develop expertise in different domains with inconsistent knowledge during tuning

- TDSLM's outputs may fall outside of the scope that CGSLM can handle



S2: Parameter-Efficient Co-Tuning (PECT)

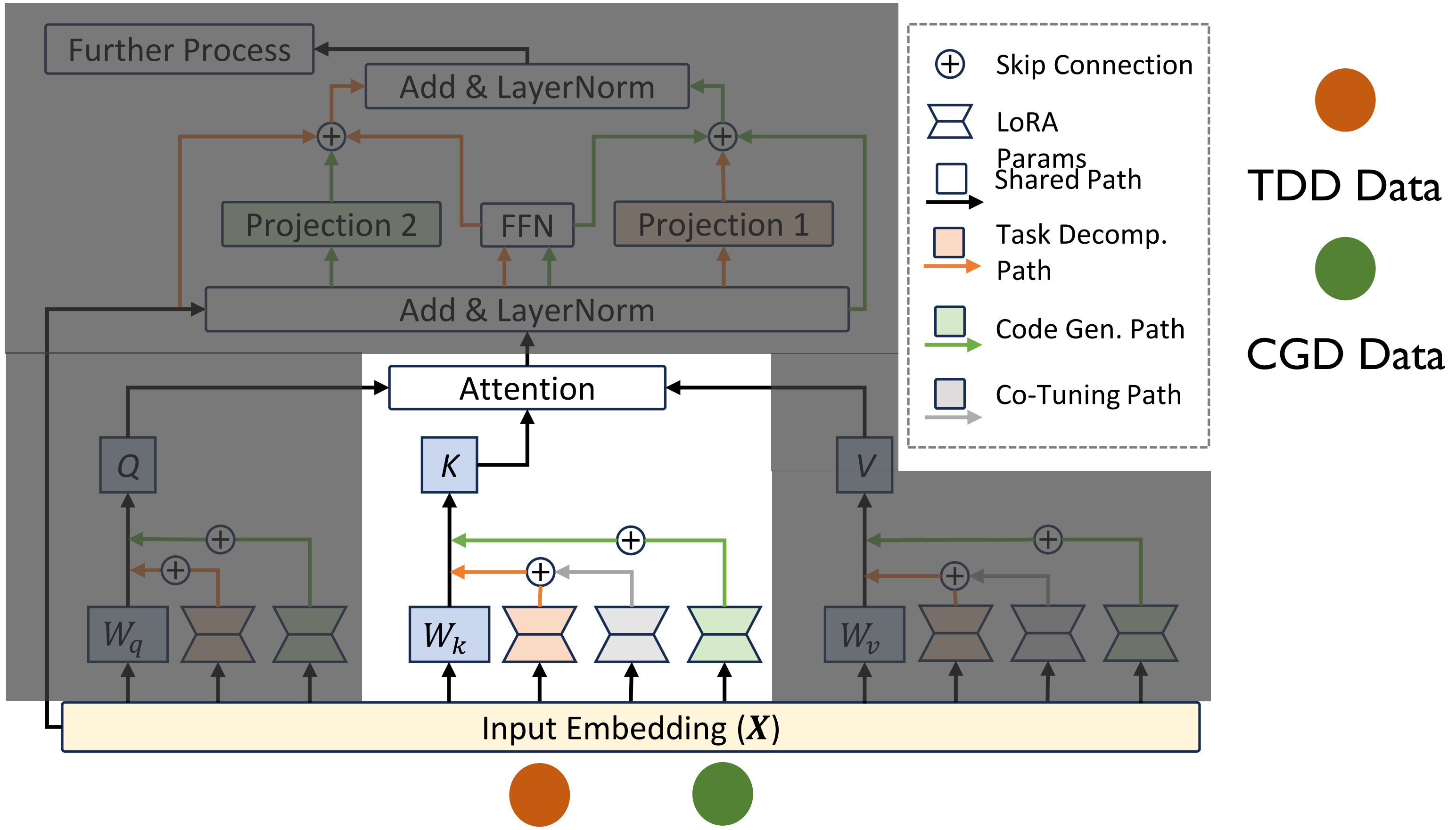
- PECT enables collaborative tuning of multiple SLMs with a shared base model but with different LoRA adapters.



THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

S2: Parameter-Efficient Co-Tuning (**PECT**)

- PECT enables collaborative tuning of multiple SLMs with a shared base model but with different LoRA adapters.



C3: Format Incompatibility

- Decomposed tasks (TDSLM output) are described in natural language
- The input (task specification) of CGSLM should be well-structured

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)

S3: Requirement Transformation

- We build a requirement transformation SLM (RTSLM)

- We prompt RTSLM to transform the TDSLM's output into well-structured task specifications via Chain-of-Thought prompts

CGSLM Input
Target Preprocess the ECG data ……
Input - signal (array)
Output - processed_signal (array)

S3: Requirement Transformation

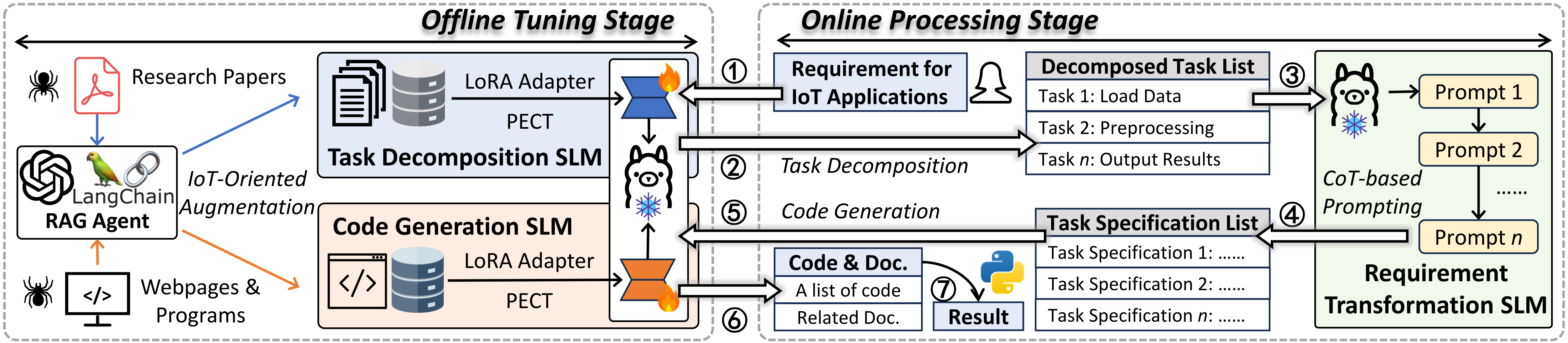
- We build a requirement transformation SLM (RTSLM)

- We prompt RTSLM to transform the TDSLM's output into well-structured task specifications via Chain-of-Thought prompts

  - User target extraction
  - I/O specification extraction

CGSLM Input
Target Preprocess the ECG data ……
Input - signal (array)
Output - processed_signal (array)

GPIoT – Put Things Together



Experiment Setup & IoT Applications

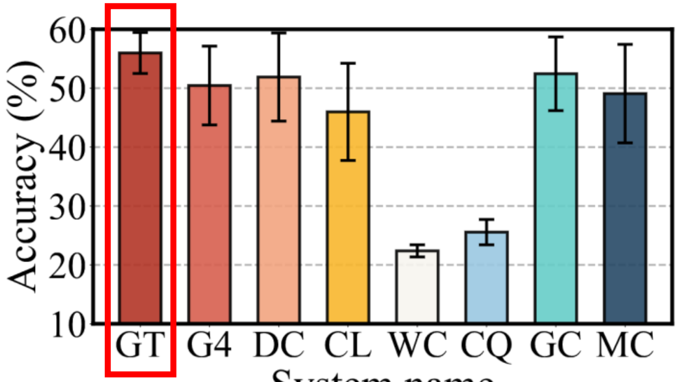
- Hardware
  - Fine-Tuning: A100 (80 GB)
  - Inference: RTX 4090 (24 GB)
- Software
  - Base Model: Llama2-13b
  - Agent: LangChain

- Heartbeat Detection (**HD**)
- Dataset: MIT-BIH

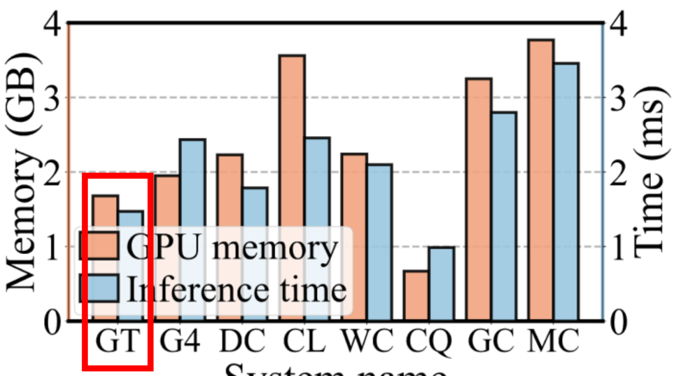
- Human Activity Recognition (**HAR**)
- Dataset: WiAR

- Multimodal HAR
- Dataset: Harmony

Multimodal HAR – Evaluation



(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

- Classification accuracy 13.44% ↑
- Moderate GPU memory
- Robustness ↑ Fluctuations ↓

GPIoT adopts model optimization methods (e.g., quantization and pruning) and data augmentation methods tailored for IoT data

GPIoT can incorporate more IoT-specific data processing and model optimization algorithms due to the embedded IoT domain knowledge during tuning.

More Experiments …

- Breakdown Evaluation of TDSLM & CGSLM
- Ablation Study
  - IoT-Oriented Data Augmentation
  - PECT
  - RTSLM
- User Study

Conclusion & Takeaways

- We present GPIoT, a tailored local code generation system IoT applications.
  - IoT-oriented text data augmentation method
  - PECT paradigm

- Future Works
  - Dynamic IoT knowledge databases
  - Continuous fine-tuning of local SLMs.

Thanks for Listening!

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

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