THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Artifacts Available V1.1 · Artifacts Evaluated Functional V1.1 · Artifacts Evaluated Reusable V1.1 · Results Replicated V1.1

AutoIOT: LLM-Driven Automated Natural Language Programming for AIoT Applications

Started Since 2023!

Leming Shen[1], Qiang Yang[2], Yuanqing Zheng[1], Mo Li[3]

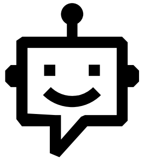
[1]The Hong Kong Polytechnic University, [2]University of Cambridge
[3]Hong Kong University of Science and Technology

ACM MobiCom 2025
Nov. 4-8, Hong Kong, China

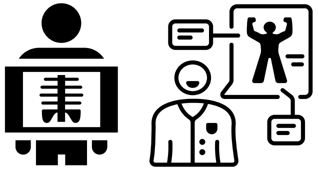
Large Language Models (LLMs)

- LLMs revolutionize our interactions with AI
- LLMs exhibit remarkable natural language understanding capabilities
- Promising applications: chatbot, medical diagnosis, etc.

ChatGPT Gemini DeepSeek



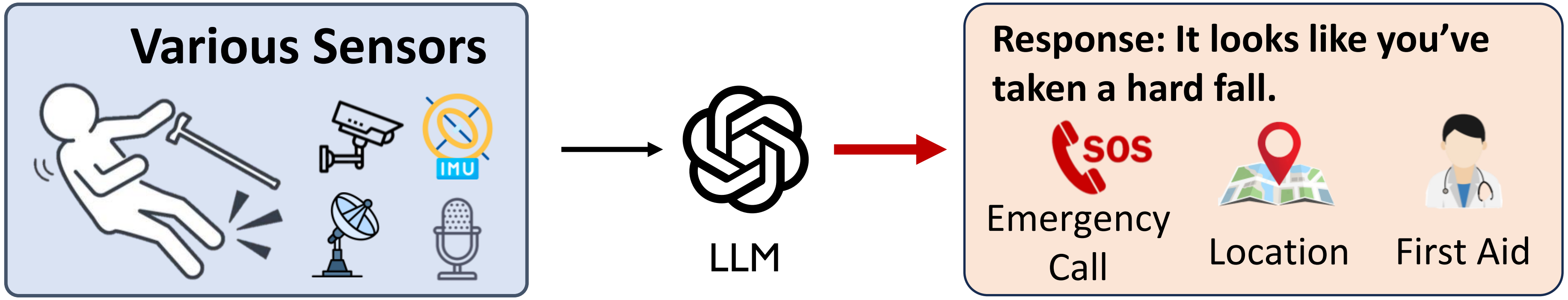
Chatbot



Medical Diagnosis

Pioneer Concept: Penetrative AI [1]

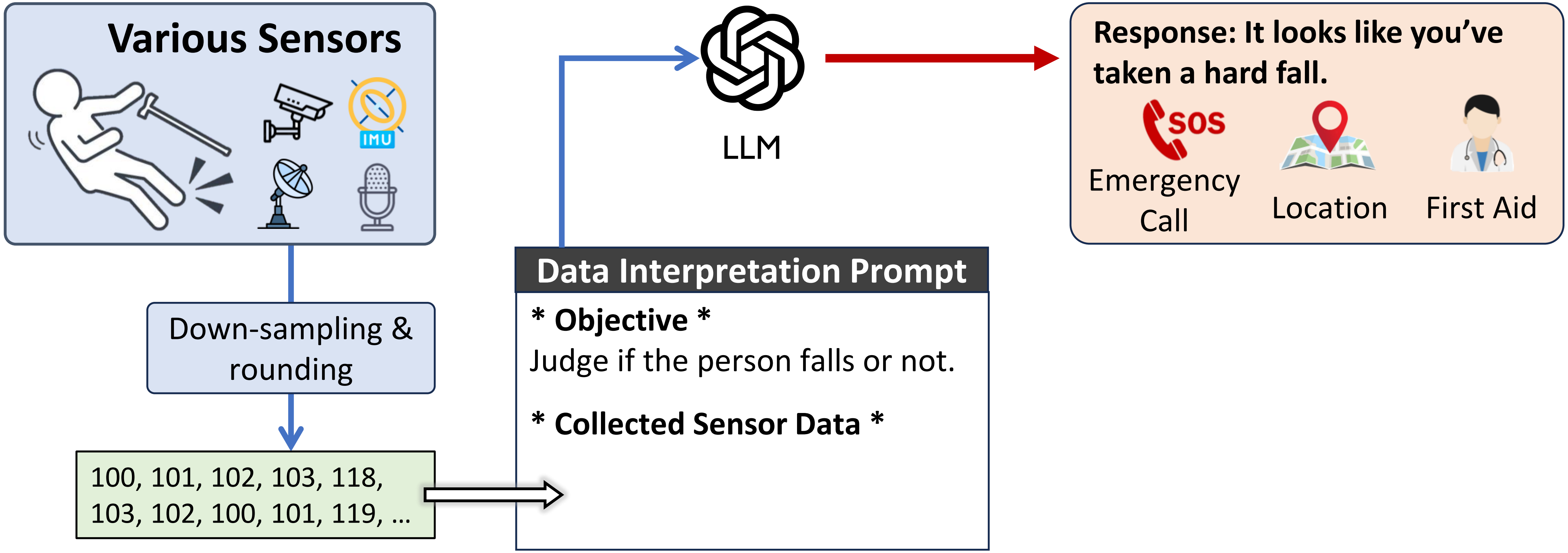
- LLMs can comprehend and even interact with the physical world



The Hong Kong Polytechnic University
香港理工大學

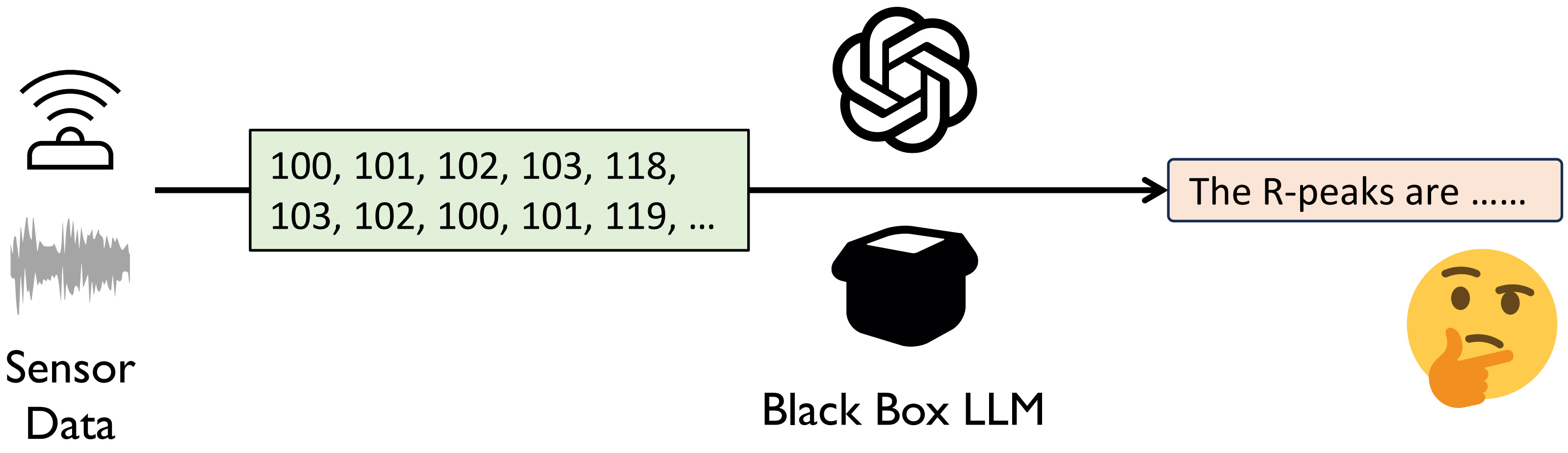
[1] Xu, Huatao, et al. "Penetrative ai: Making llms comprehend the physical world." Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications. 2024.

Penetrative AI – Basic Workflow



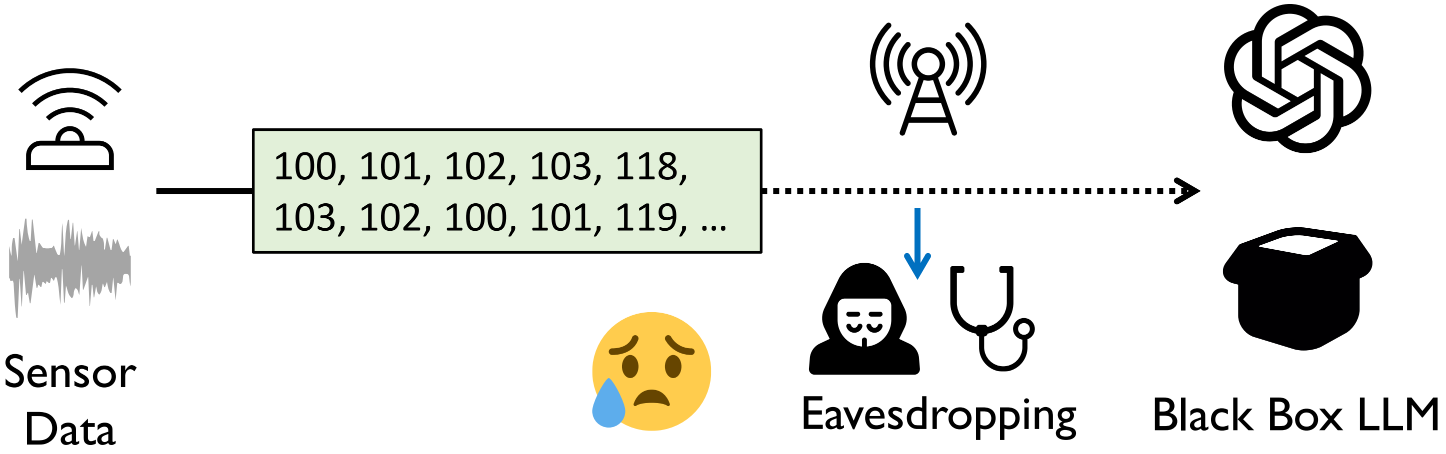
Penetrative AI (Limitation 1)

- Compromised trustworthiness of the inference results
- Hard to verify the correctness



Penetrative AI (Limitation 2)

- Transmitting sensor data over the network raises privacy concerns



Penetrative AI (Limitation 3)

- Sensor data often exhibits extensive length and high dimensionality
  - Prohibitive token costs
  - Increased response latency
  - Infeasible due to token limits

- Ideally, the integration of LLMs with AIoT should be trustworthy, privacy-preserving, and communication-efficient

- LLMs have shown their remarkable capabilities in code generation …

GitHub Copilot   Code Llama   CURSOR

Research Question

- Ideally, the integration of LLMs with AIoT applications should be trustworthy, privacy-preserving, and communication-efficient

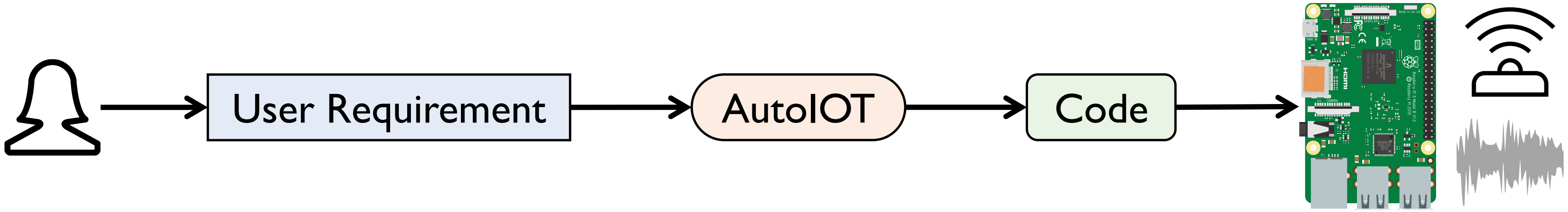
- LLMs have shown their remarkable capabilities in code generation …

Can we leverage LLMs to synthesize programs to fulfill AIoT application requirements?

GitHub Copilot   Code Llama   CURSOR

Our Solution: AutoIOT

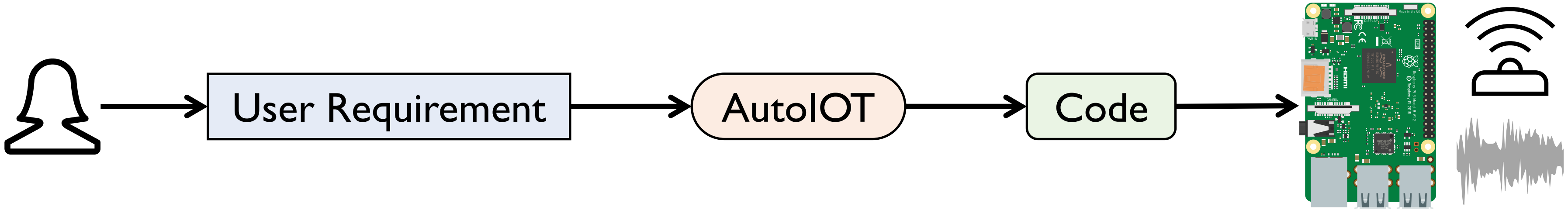
- Given a user requirement, AutoIOT automatically synthesizes programs, which are locally executed to perform various AIoT tasks.



Our Solution: AutoIOT

- Given a user requirement, AutoIOT automatically synthesizes programs, which are locally executed to perform various AIoT tasks.

- Enhance the explainability and trustworthiness
- Mitigate privacy concerns and reduce communication costs
- Efficiently process high-dimensional sensor data

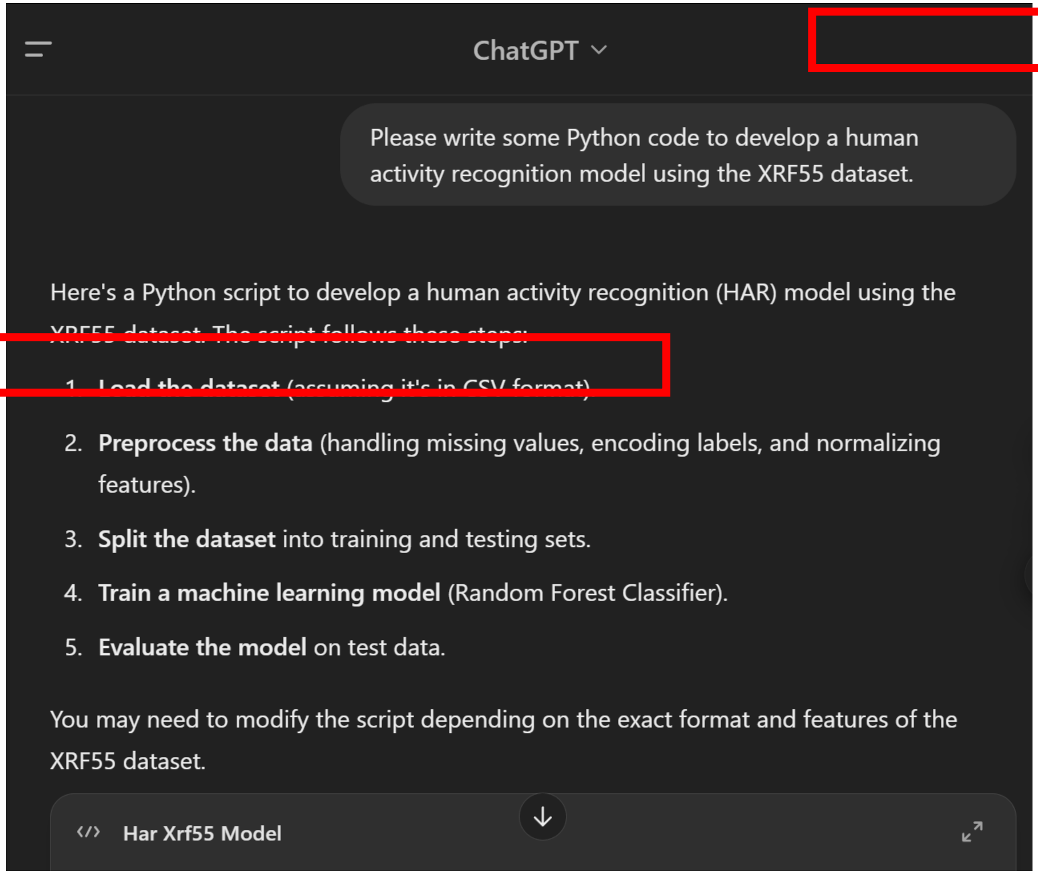


Technical Challenges of AutoIOT

- **C1**: Lack of Domain Knowledge in AIoT
- **C2**: High Complexity of AIoT Tasks
- **C3**: Heavy Intervention and Constant Feedback

C1: Lack of Domain Knowledge in AIoT

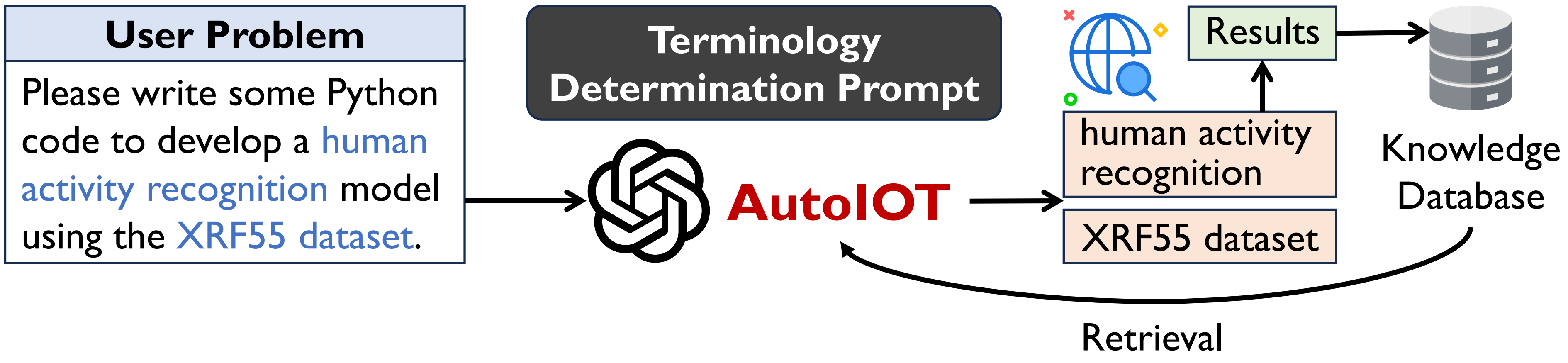
- LLMs are pre-trained on general corpus datasets.
- They may not include the latest AIoT domain knowledge.
- Hallucination issues



THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

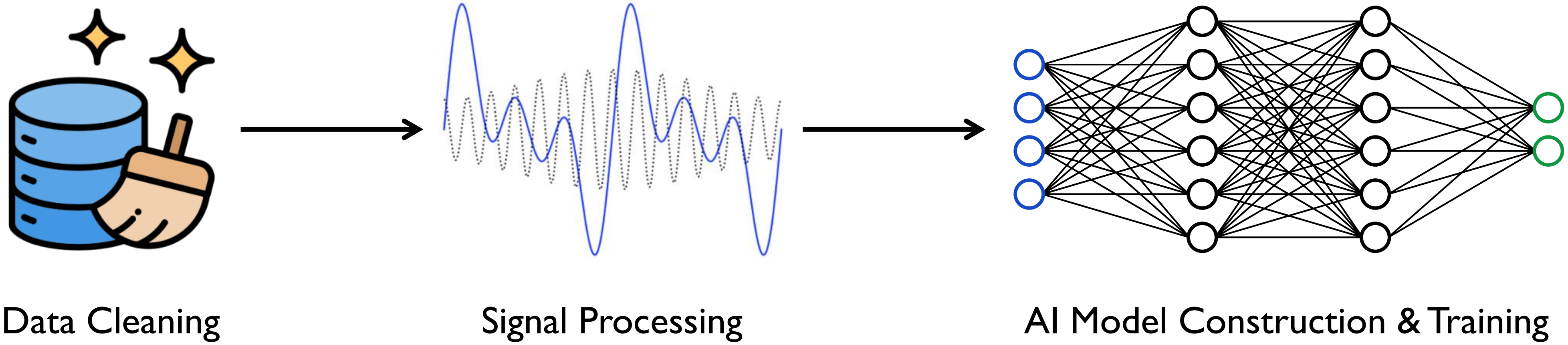
Solution: Background Knowledge Retrieval

- Terminology Determination & Searching
- Context Database Construction



C2: High Complexity of AIoT Tasks

- Existing works generate code for individual modules or functions
- AIoT applications typically require systematic designs and integration involving multiple functional components



C3: High Complexity of AIoT Tasks

- LLMs tend to provide simple and general code
  - Generates some null functions without concrete implementations
  - Imports some nonexistent packages

Help me implement a human activity recognition system using the XRF55 dataset.

Certainly! First, we need to perform data cleaning with some signal processing methods, then we can …… Here is a simple solution:

Python | Copy code

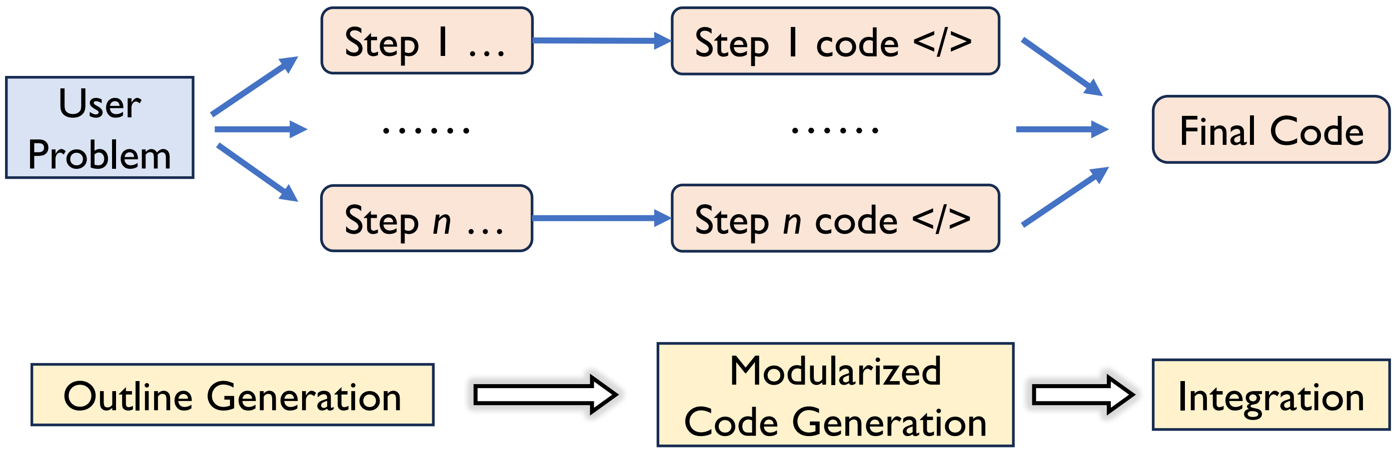
```python
import numpy as np
from XRF55 import dataset
def data_cleaning(signal):
    # Perform data cleaning
def signal_processing(signal):
    # Perform signal processing
def model_training(cleaned_signal):
    # Train the model
```

C3: Heavy Intervention and Constant Feedback

- Provide timely feedback and constantly intervene in the entire development process
  - Specific reference materials
  - Specific algorithms
  - Clearly described methods
  - Manual debugging

Solution 2: Automated Program Synthesis

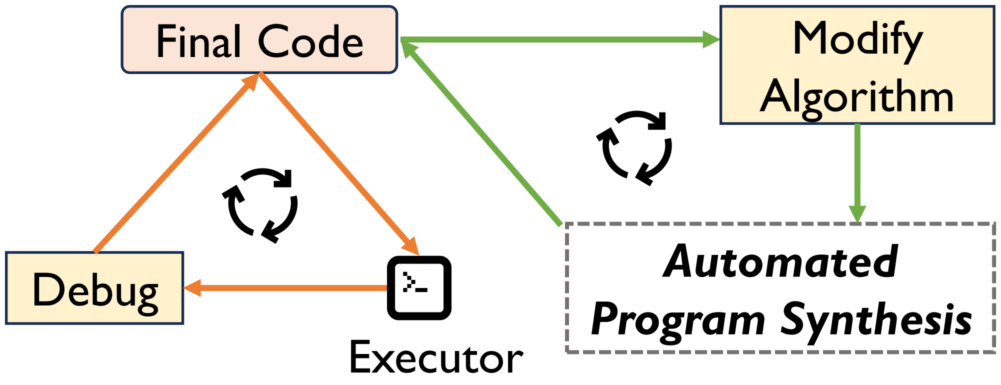
- Chain-of-Thought (CoT) prompts → step-by-step reasoning.
- Mimic human-like divide-and-conquer reasoning.



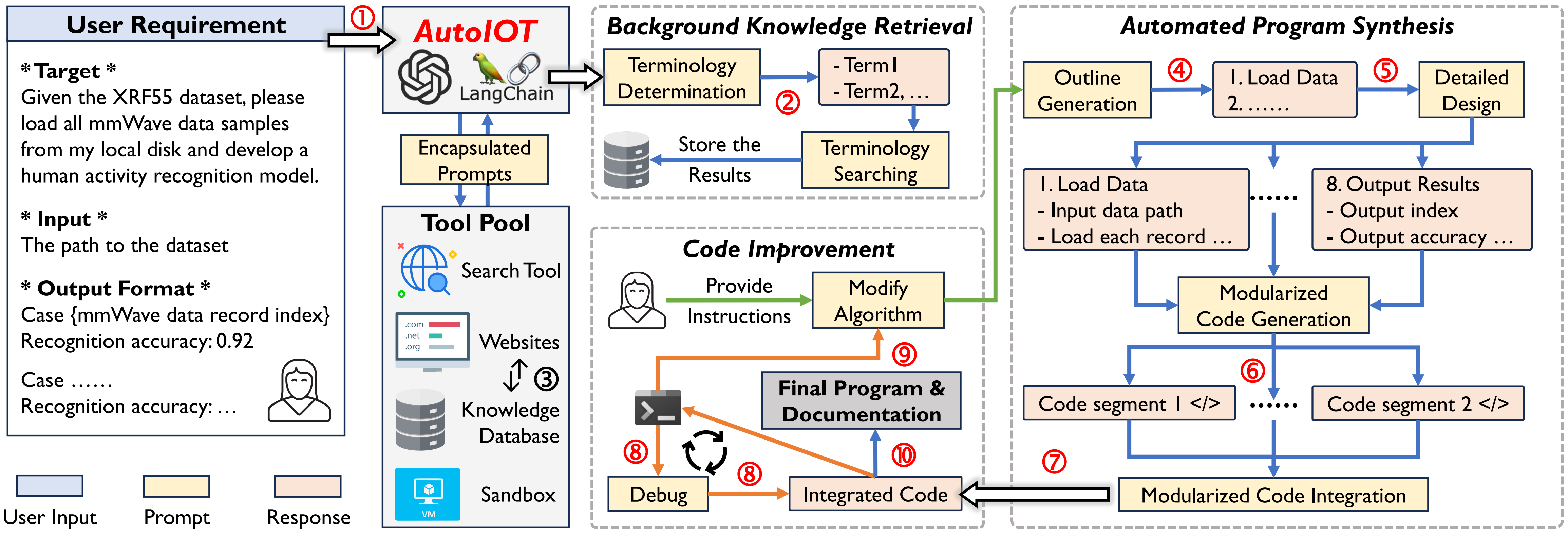
Solution 3: Automated Code Improvement

- Execute code in an executor
- Analyze executor's outputs for debugging
- Prompts the LLM to adopt more advanced algorithms

- This initiates a new recursive cycle of program synthesis



Put All Things Together – AutoIOT



Experiment Setup – Implementation

- Software Configurations
  - AutoIOT agent: LangChain
  - Default LLM: GPT-4
  - Web search tool: Tavily AI
  - Knowledge database: Chroma

- Hardware Configurations
  - A workstation installed with Ubuntu 20.04 LTS
  - An NVIDIA RTX 4090 GPU

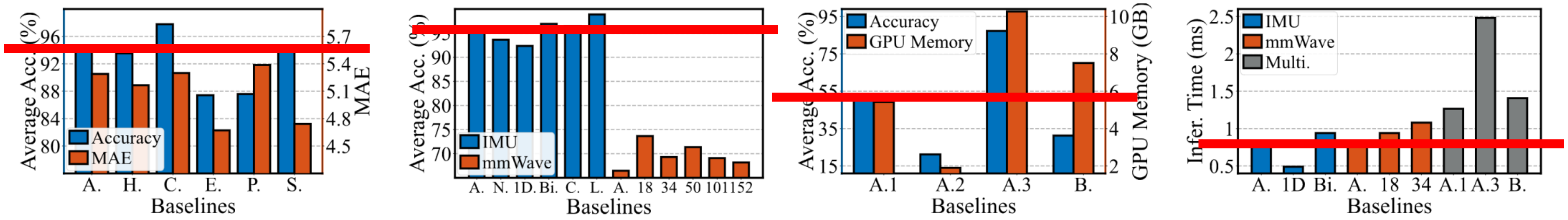
Experiment Setup – Applications

- **Heartbeat Detection** (MIT-BIH Arrhythmia Database [1]
- Baseline: Hamiltion, Christov, Engzee, Pan-Tompkins, SWT
- **IMU-based HAR** (WISDM dataset [2])
- Baseline: LSTM-RNN, 1D-CNN, Conv-LSTM, BiLSTM, NN
- **mmWave-based HAR** (XRF55 dataset (newly published) [3])
- Baseline: ResNet18, ResNet34, ResNet50, ResNet101
- **Multimodal HAR** (Harmony (newly published) [4])
- Baseline: Encoder-Classifier

THE HONG KONG POLYTECHNIC UNIVERSITY 香港理工大學

[1] Moody, George B., and Roger G. Mark. "The impact of the MIT-BIH arrhythmia database." IEEE engineering in medicine and biology magazine (2001).
[2] Kwapisz, Jennifer R., Gary M. Weiss, and Samuel A. Moore. "Activity recognition using cell phone accelerometers." ACM KDD (2011).
[3] Wang, Fei, et al. "Xrf55: A radio frequency dataset for human indoor action analysis.“ ACM IMWUT (2024).
[4] Ouyang, Xiaomin, et al. "Harmony: Heterogeneous multi-modal federated learning through disentangled model training.“ ACM MobiSys (2023).

Evaluation – Overall Performance

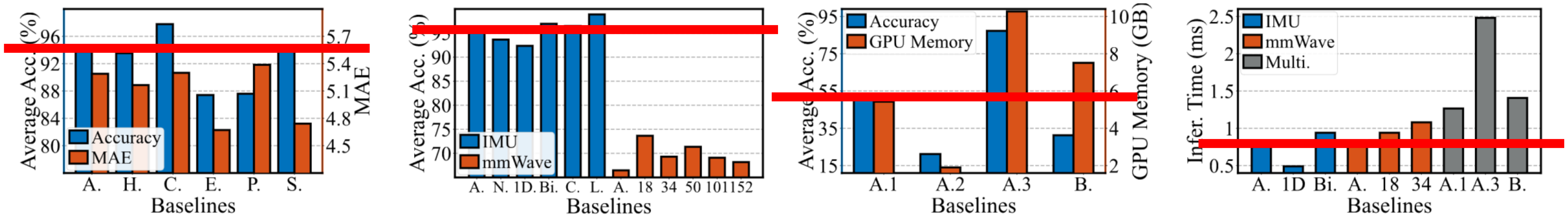


(a) Heartbeat detection (b) IMU & mmWave-based HAR (c) Multimodal HAR (d) Inference time per sample

Figure 10: The overall performance of the four IoT applications. In (a), A. for *AutoIOT*, H. for Hamiltion, C. for Christov, E. for Engzee, P. for Pan-Tompkins, and S. for SWT. In (b), N. for NN, 1D for 1D-CNN, Bi. for BiLSTM, C. for Conv-LSTM, L. for LSTM-RNN, and *n* for ResNet-*n*. In (c) & (d), A.1, A.2, and A.3 for three different *AutoIOT*-generated programs; B. for the baseline in the multimodal HAR application.

AutoIOT-synthesized programs can achieve comparable performance to the baselines and sometimes outperform them.

Evaluation – Overall Performance



(a) Heartbeat detection (b) IMU & mmWave-based HAR (c) Multimodal HAR (d) Inference time per sample

Figure 10: The overall performance of the four IoT applications. In (a), A. for *AutoIOT*, H. for Hamiltion, C. for Christov, E. for Engzee, P. for Pan-Tompkins, and S. for SWT. In (b), N. for NN, 1D for 1D-CNN, Bi. for BiLSTM, C. for Conv-LSTM, L. for LSTM-RNN, and *n* for ResNet-*n*. In (c) & (d), A.1, A.2, and A.3 for three different *AutoIOT*-generated programs; B. for the baseline in the multimodal HAR application.

- Advanced signal processing algorithms (knowledge retrieval) → High Accuracy
- Comprehensive module design (CoT prompt) → Robustness & Fewer Errors
- CUDA optimization strategy (iterative refinement) → Moderate GPU Usage

Further Experiments

- Ablation Study
  - Background knowledge retrieval
  - Chain-of-thought
  - Code improvement

- User Study
  - Objective Evaluation
  - Subjective Evaluation

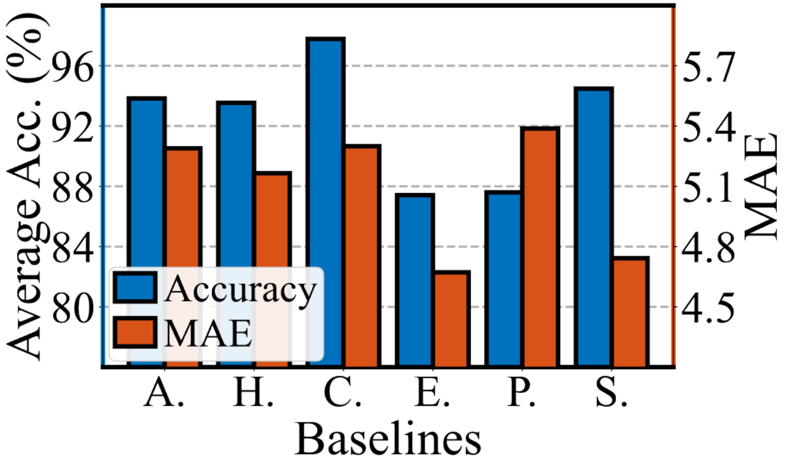
More details in our paper.

Evaluation – Lessons Learned

- Given a single performance objective, the LLM carries out extensive optimization, sometimes at the cost of other important metrics.

Heartbeat Detection
A large sliding window
high accuracy but low precision

A method that can elicit comprehensive and clear user requirements.



(a) Heartbeat detection

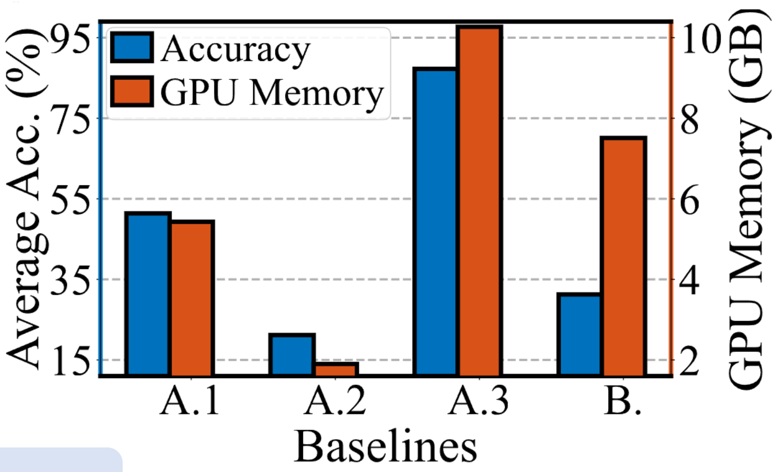
Evaluation – Lessons Learned

- AutoIOT can adjust the generated code to fulfill different user requirements considering constrained resources of IoT devices.

A.1: basic information of the task
A.2: constrained GPU memory + A.1
A.3: high accuracy demand + A.1

Developers need to specify detailed performance requirements.

Device Profiling



(c) Multimodal HAR

Conclusion & Takeaways

- AutoIOT is an LLM-driven automated natural language programming system for AIoT applications.
- Limitations of AutoIOT
  - Cloud LLM (GPT-4) → Privacy concerns & Unstable networks
  - Knowledge retrieval → Large model & Strong language processing capabilities
- Further Work – GPIoT (SenSys '25)
  - Collect IoT-relevant text generation datasets
  - Fine-tune multiple locally deployed small language models

THE HONG KONG POLYTECHNIC UNIVERSITY 香港理工大學

[1] Leming Shen, Qiang Yang, Xinyu Huang, Zijing Ma, and Yuanqing Zheng. "GPIoT: Tailoring Small Language Models for IoT Program Synthesis and Development." SenSys 2025.

Thanks for Listening!

- AutoIOT: LLM-Driven Automated Natural Language Programming for AIoT Applications
- Leming Shen, Qiang Yang, Yuanqing Zheng, Mo Li

ACM MobiCom 2025
Nov. 4-8, Hong Kong, China