2025-1 PPPD

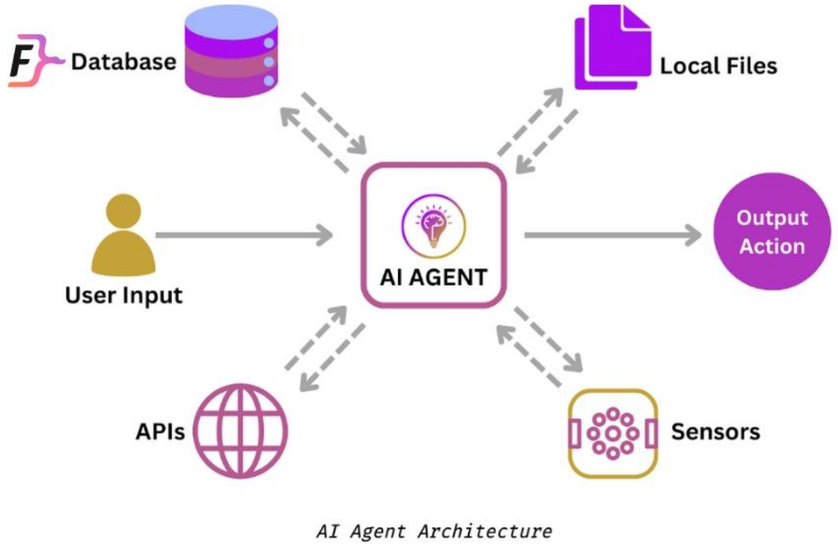
Programming in the AI Era

- Evaluating Code Generation of AI Agent with Benchmark and Application on VRPTW Problem

Team Members: 서재원, 송경현, 이동재, 임재민, 장준혁, 최지안

Contents

What is an AI Agent?



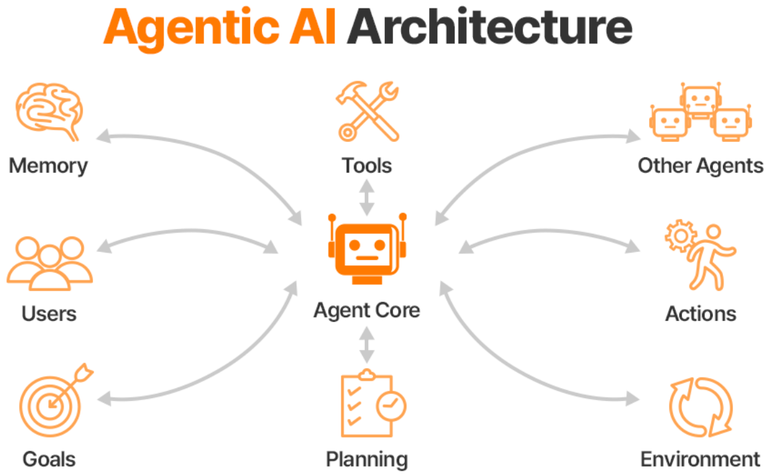
AI Agent Architecture

- AI agents are autonomous systems that make decisions and select the optimal input among multiple inputs to solve a given goal
- It is made up of repetition of the process of ' Perception → Processing → Action → Decision Making'

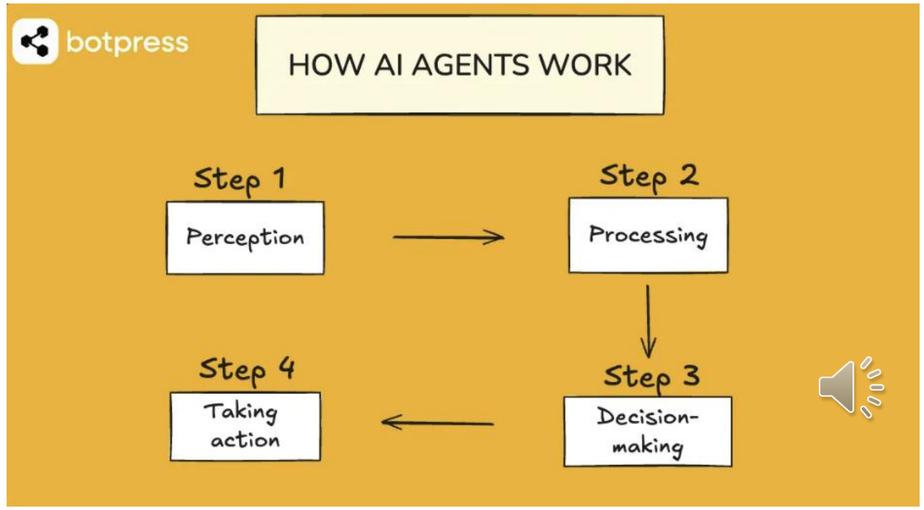
What is AI Agent?

concept

- Autonomously selects inputs from among multiple inputs and makes decisions on its own to solve given goals
- Perception -> Processing -> Action -> Decision making
- Uses LLM routing: assigns roles to multiple models and makes them interact with each other to find the right answer.



출처 : Blog.ori.co



How AI agents work

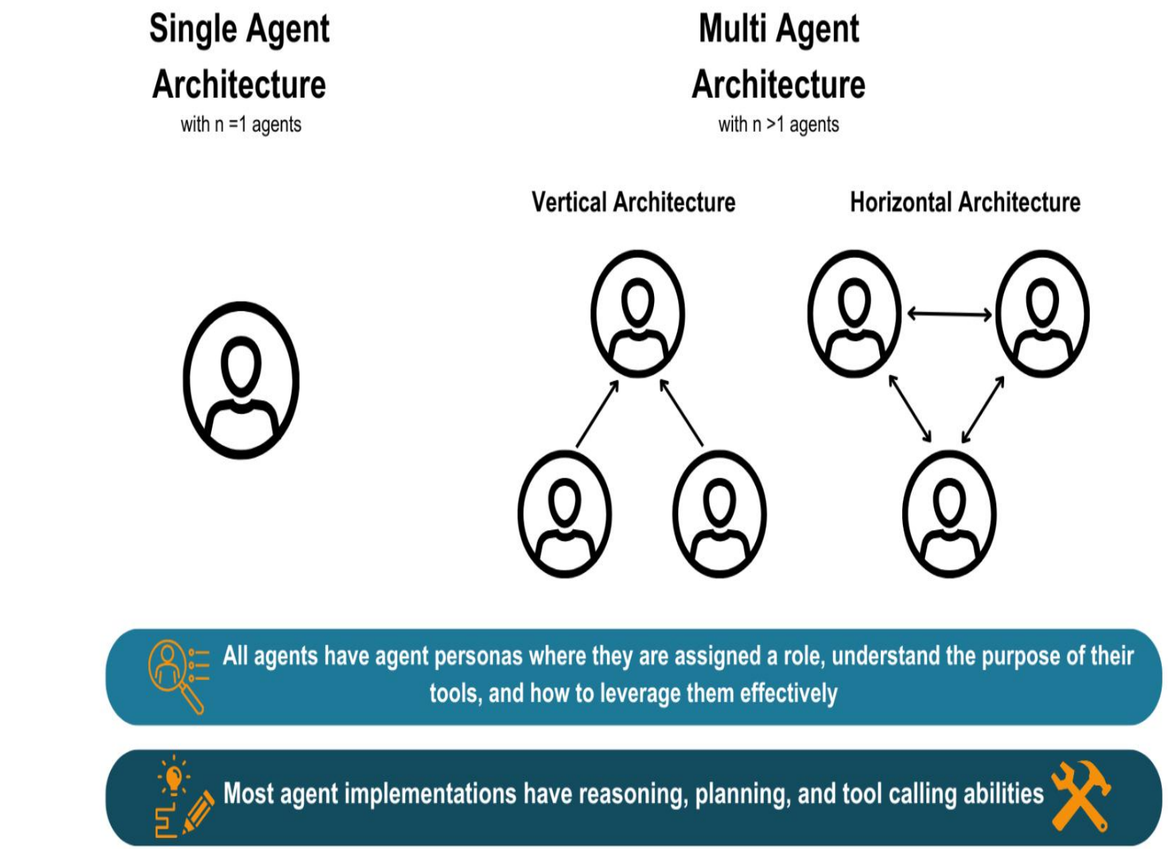


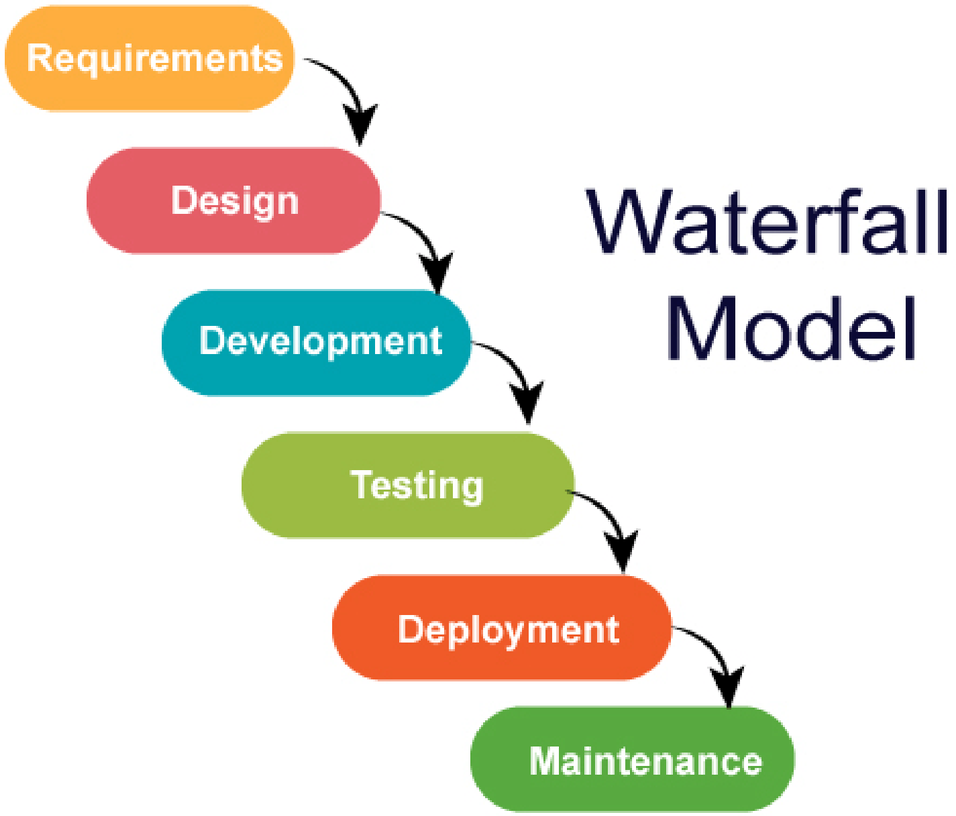
Figure 1: A visualization of single and multi-agent architectures with their underlying features and abilities

- In a multi-agent system, agents with different roles interact with each other to solve problems.
- Ability to solve problems autonomously without human intervention

Project Topic

- Developing AI Agent following software design process (Based on LangGraph)
- Compare AI agent with LLM with Benchmark and solving using case study (VRPTW)
- Evaluate limitations, implications, and directions of development of AI programming

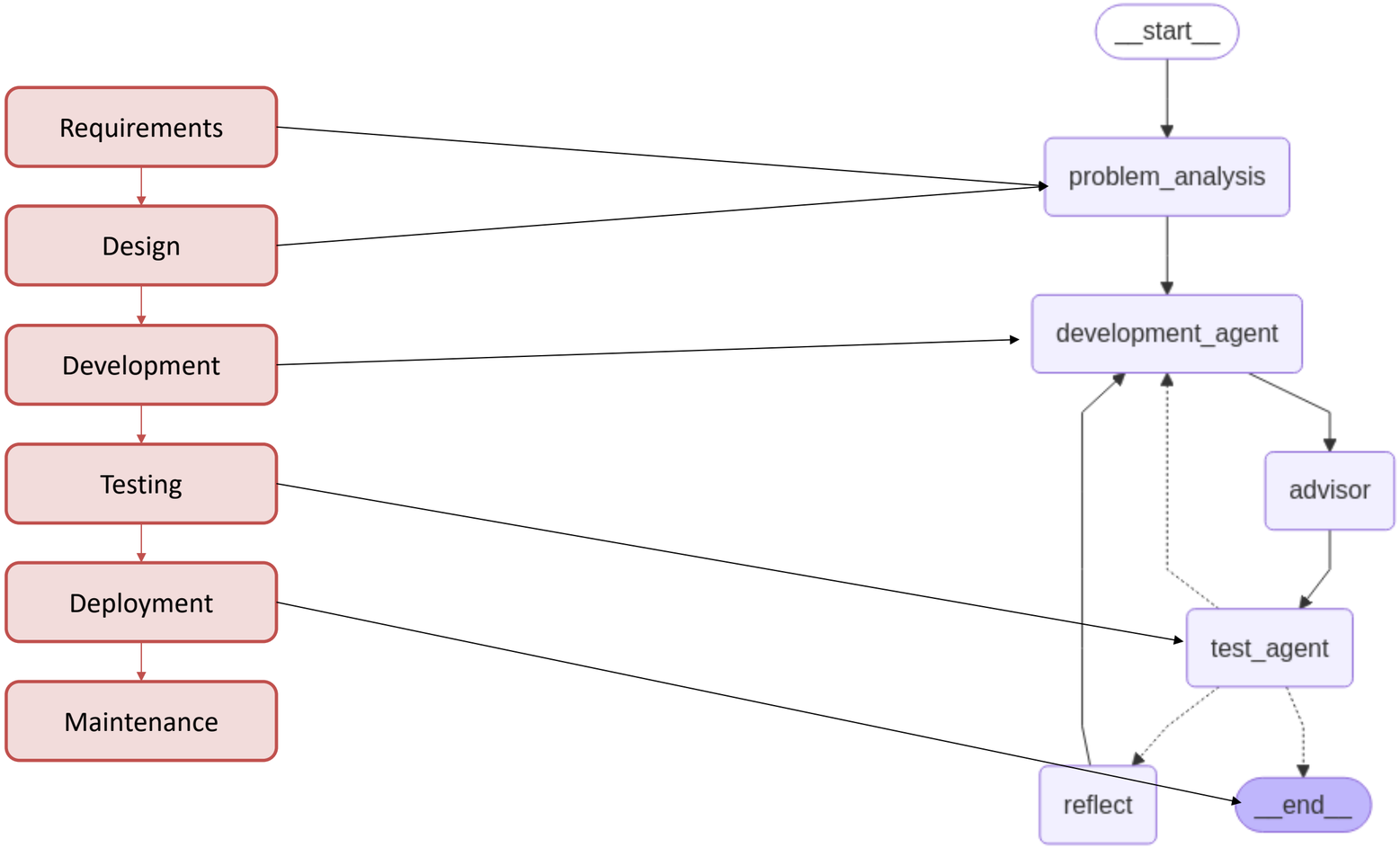
Process Influenced Our AI Agents



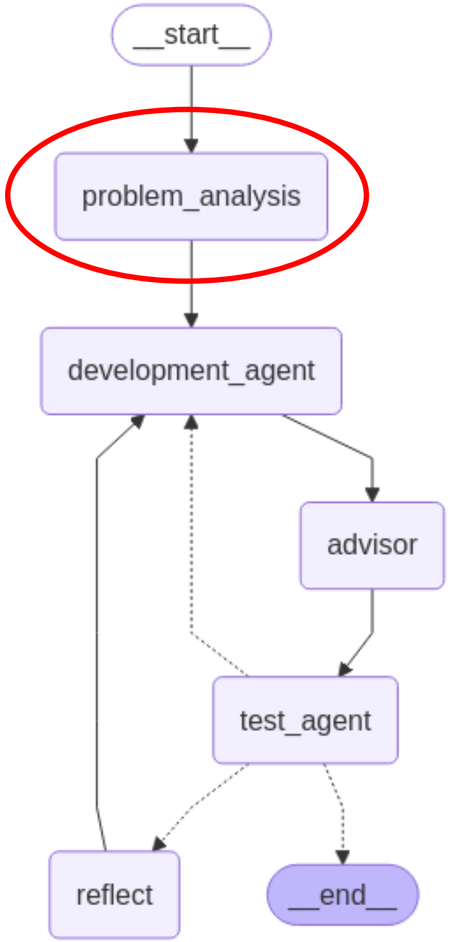
Influenced by Waterfall Model

- **Requirements:** Collect and document all software
- **System Design:** Translate requirements into system architecture
- **Implementation (Coding):** Develop the software by writing code
- **Testing:** Test the software to identify and fix defects, ensuring it meets the specified requirements.
- **Deployment:** Release the fully developed and tested software to the end-users.
- **Maintenance:** Perform ongoing support and maintenance to fix issues and update the

Structure of AI Agents



Structure of AI Agent

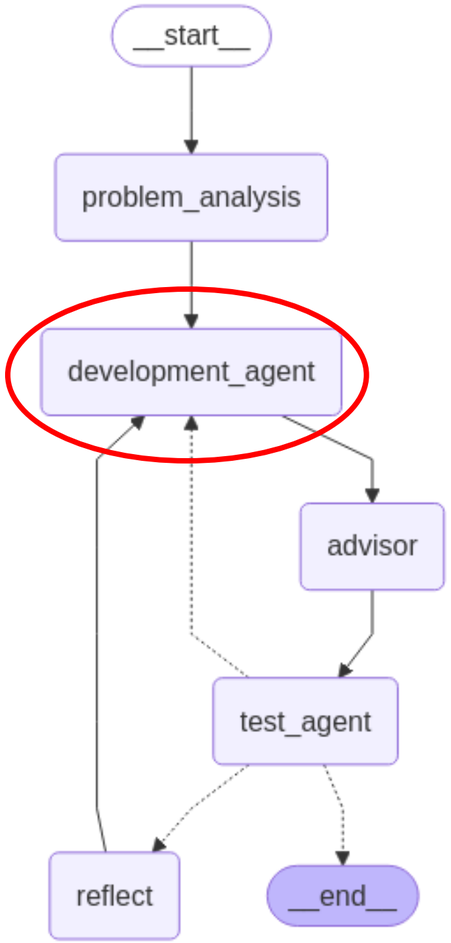



```
problem_defining_prompt = ChatPromptTemplate.from_messages(
    [
        (
            "system",
            """Role: You are a problem defining and analysis agent for general codign problems.\n
            Tasks:\n
            1. Define function and its objective\n
            2. Define input and output format\n
            3. Define functional requirements and constraints\n
            4. Wrute  pseudo code\n
    Structure your answer with a description of the problem defining. \n
    Keep length of token under 1600. Here is the user question:""",
        ),
        ("placeholder", "{messages}"),
    ])


class structure(BaseModel):
    analysis: str = Field(description="Description of the problem, functional requirements, constraints, objective.")
    design: str = Field(description="Description of the function, data structures,input format, output format.")
    pseudo_code: str = Field(description="pseudo_code for given problem")
```

- Define problem following given structure of Waterfall Model
- Format: problem description, design of function, pseudo code

Structure of AI Agent

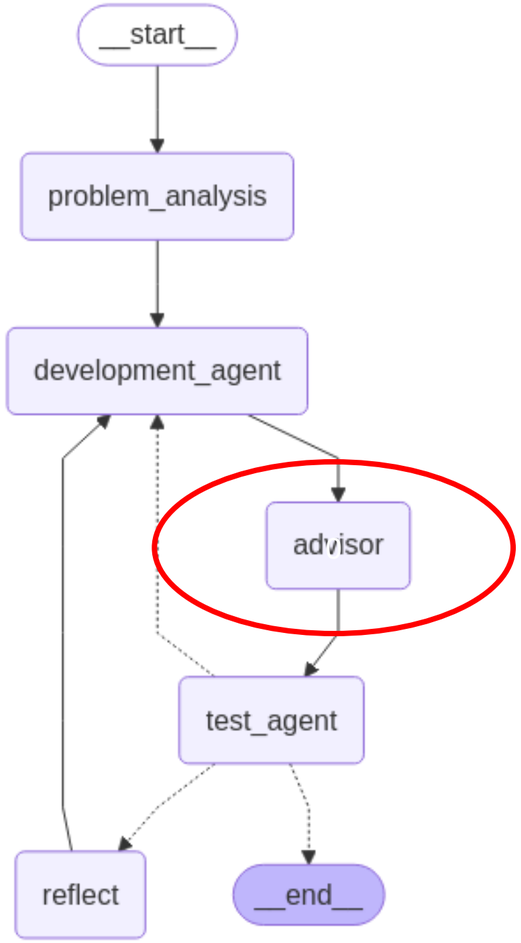



```python
code_gen_prompt = ChatPromptTemplate.from_messages(
    [
        (
            "system",
            """You are a coding assistant for general codign problems. Ensure any code you provide can be executed \n
    with all required imports and variables defined. Structure your answer with a description of the code solution. \n
    Then list the imports. And finally list the functioning code block. Here is the user question:""",
        ),
        ("placeholder", "{messages}"),
    ]
)


# Data model
class code(BaseModel):
    prefix: str = Field(description="Description of the problem and approach")
    imports: str = Field(description="Code block import statements")
    code: str = Field(description="Code block not including import statements")
```

- Generate code using given problem analysis from problem_analysis agent
- Format: prefix, imports, code
- Divide import part and code part so that ensure requirements and reduce errors

Structure of AI Agent

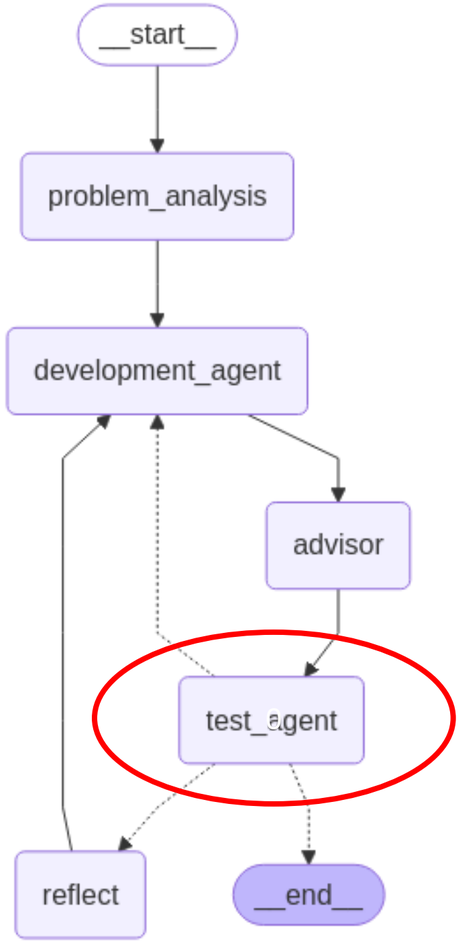



```
advise_prompt = ChatPromptTemplate.from_messages(
    [
        (
            "system",
            """Role: You are a code advisor agent for general codign problems.\n
            You should revise code before running it\n
            Tasks:\n
            1. Review the given code and check whether the code structure is correct or not\n
            2. Revise code where you find lack of logic\n
            3. Revise code where has probability of error\n

    After the all process above, give the revised version of code following the given format. Here is the given code:""",
        ),
        ("placeholder", "{messages}"),
    ])
```

- Revise code from development_agent before run, precheck the code
- Divide import part and code part so that ensure requirements and reduce errors

Structure of AI Agent

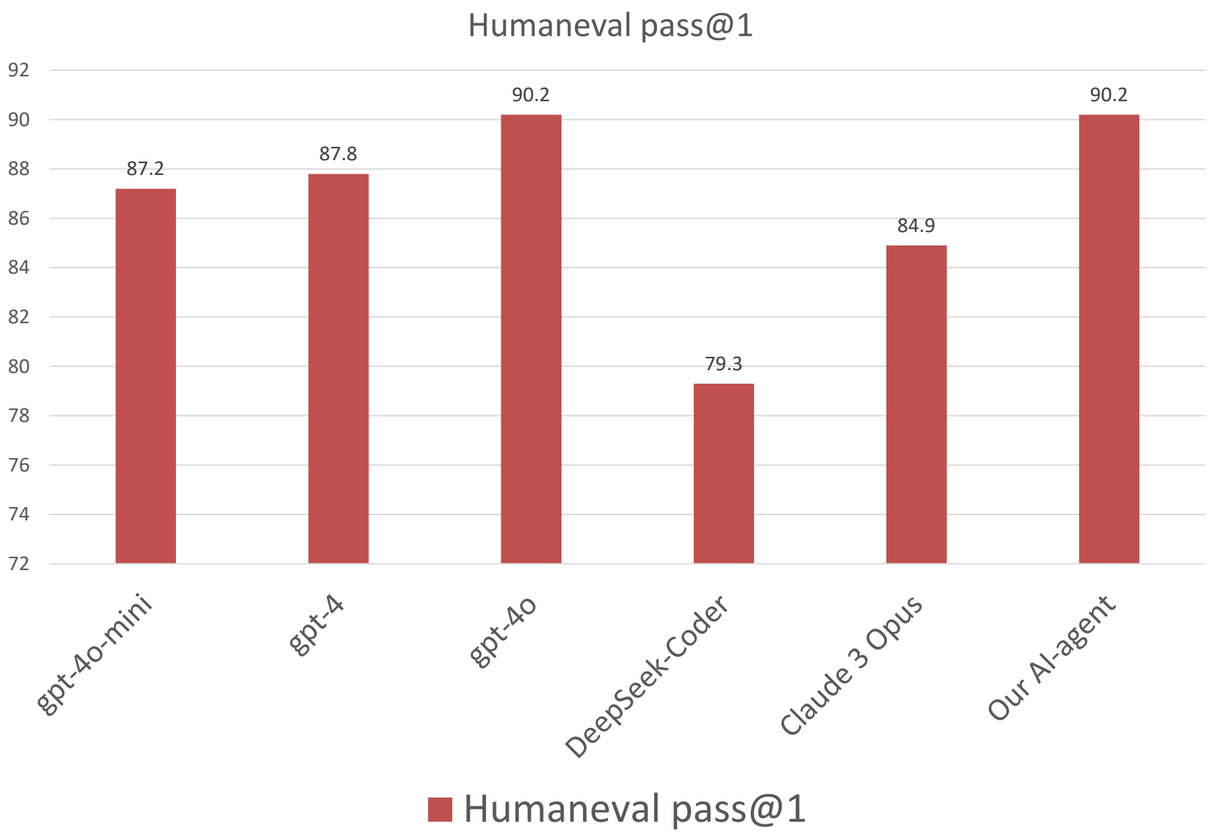


Test_agent

- Run code and check whether error occurs of not

```
While iteration != max_iteration:
        if error occurs while running:
                go to reflect agent -> reflect code with error analysis
                pass error analysis to development_agent and rewrite code
        if not error:
                export code
                break
```

Performance

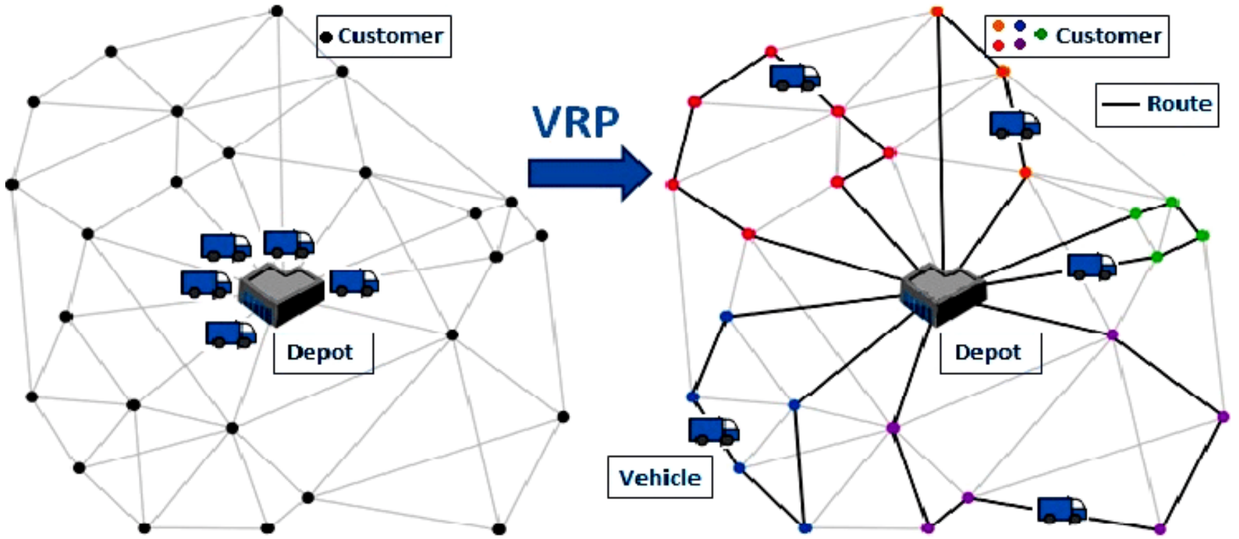


- Since our model is based on gpt-4o-mini, it's reasonable to compare with it.
- Performance at Humaneval Benchmark exceeded score of gpt-4o-mini and other LLMs'.

** Humaneval: LLM benchmark evaluating code generation for general hand-written problems. Compare percentage for 164 problems.*

VRPTW: Vehicle Routing Problem with Time Windows

- VRPTW is an extension of the classical Vehicle Routing Problem (VRP), where each customer must be served within a specific time window.
- Vehicles with limited capacity
- Customers with: Demands, Time windows (e.g., 9:00–11:00 AM)
- Minimize cost while respecting time and capacity constraints

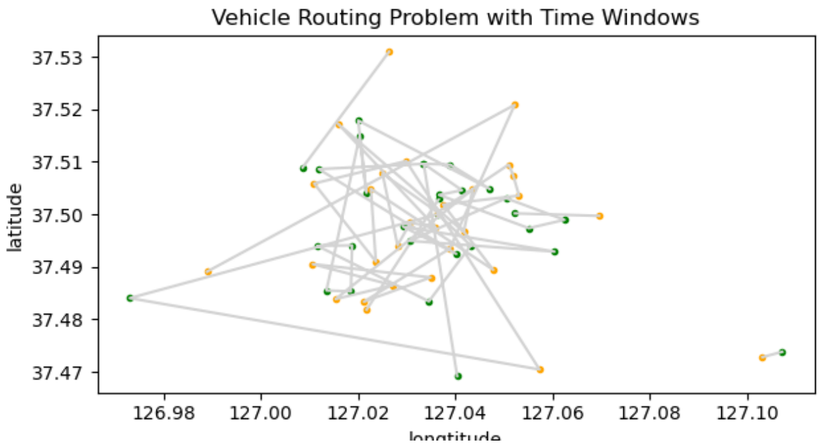


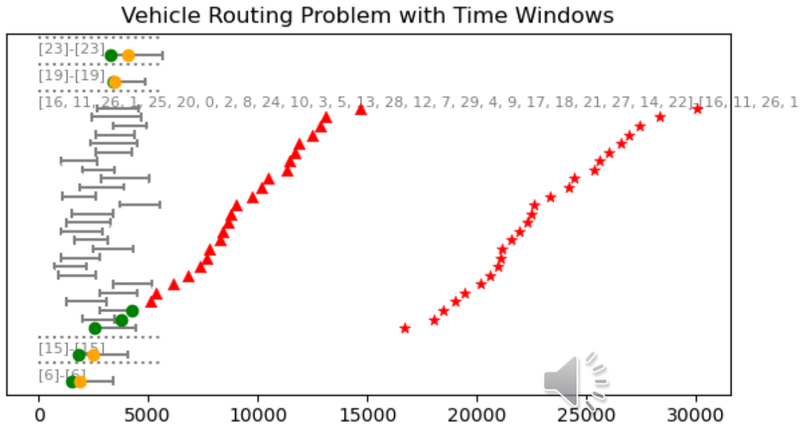
Compare gpt-4o vs Our AI Agent

Problem solved by gpt-4o

- User query: Read this file and make the code to solve VRPTW problem following conditions in file + Problem file
- Result Analysis
  - Generated code did not consider most of volume and time window constraints
  - Singular LLM still has difficulties to solve problem which has multiple constraints and definitions.


```
{
  "algorithm": "simple saving algorithm",
  "computation_time": 0.005,
  "average_cost": 6767.43,
```

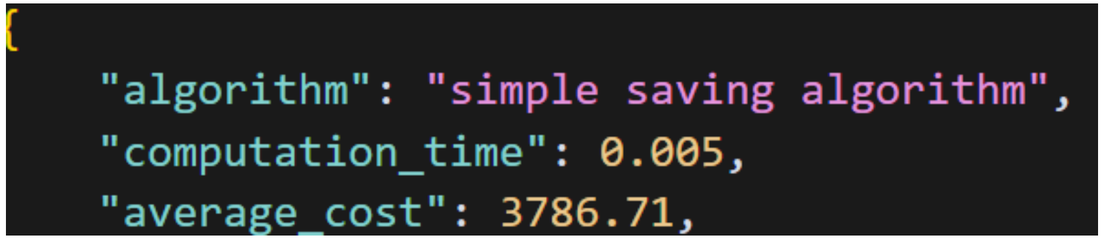





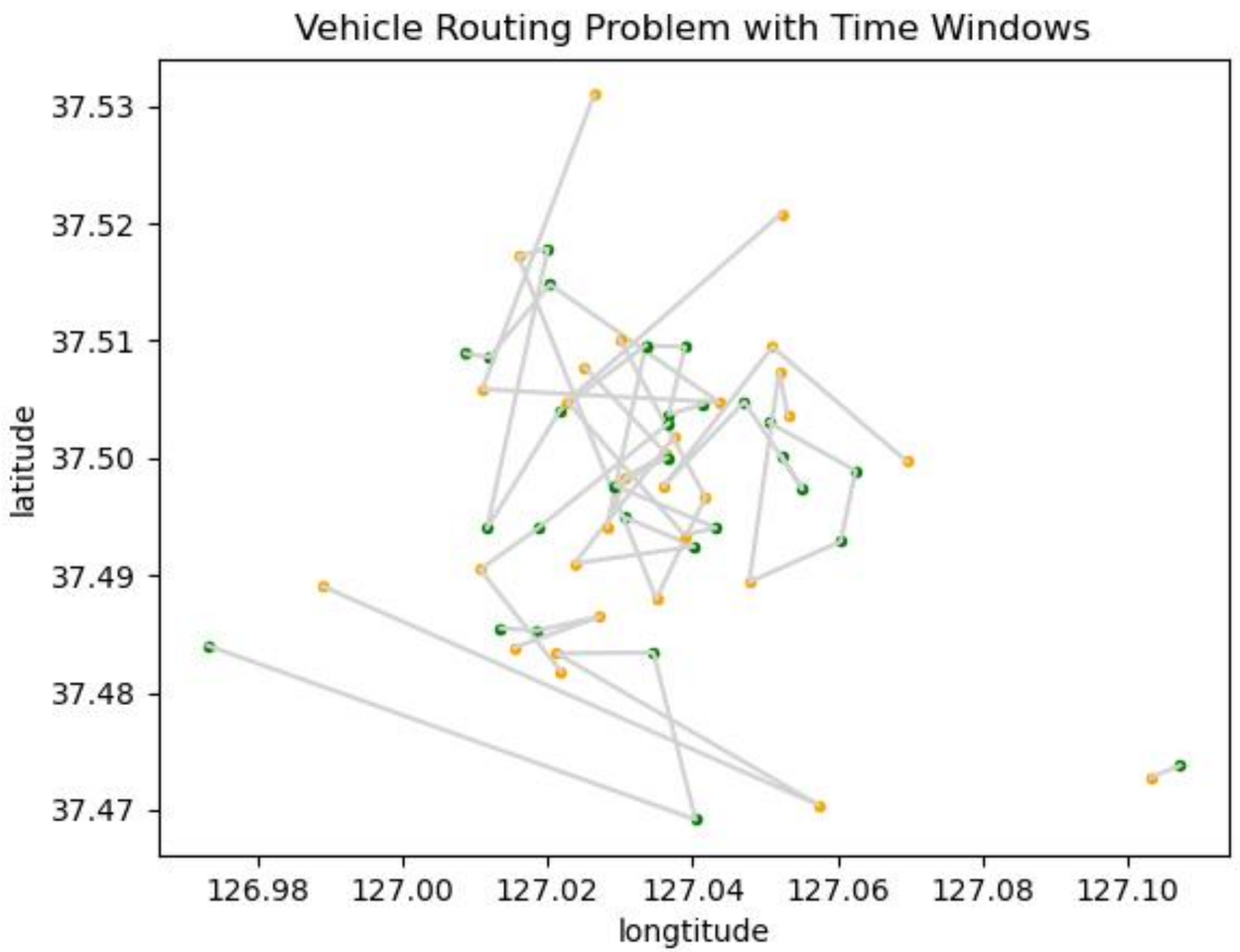
Compare gpt-4o-mini vs Our AI Agent

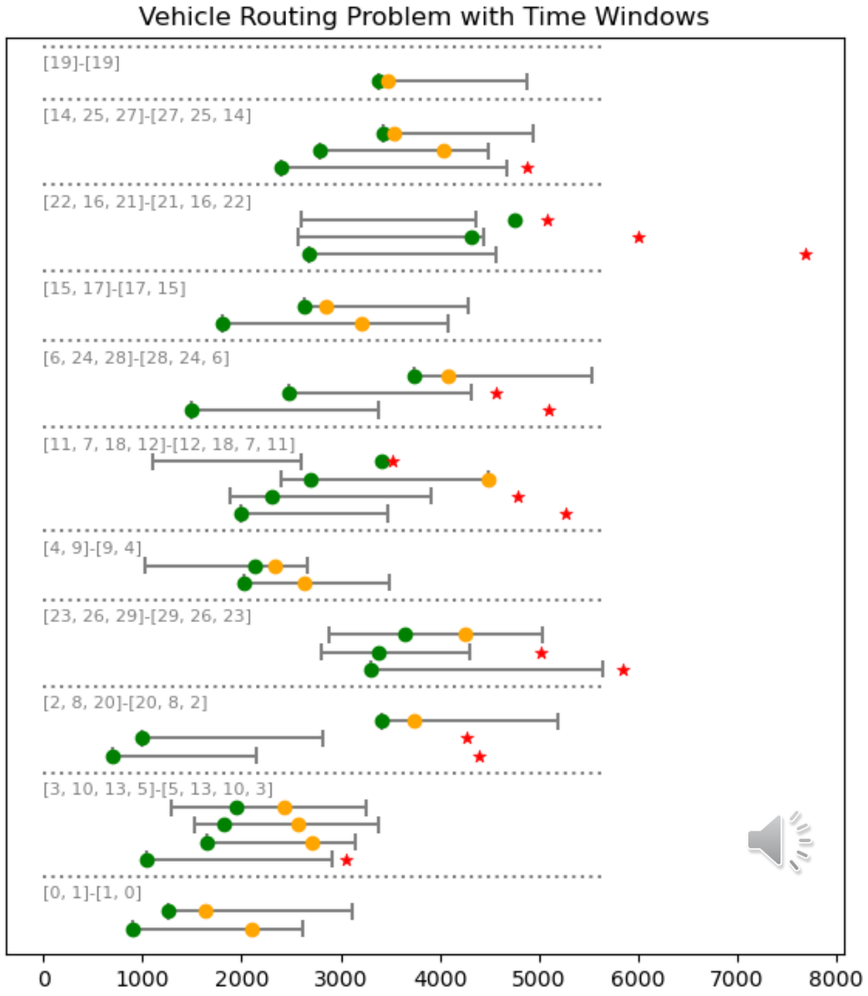
Problem solved by Our AI Agent

- User query: Read this file and make the code to solve VRPTW problem following conditions in file + Problem file
- Result Analysis
  - Generated code did not consider volume constraint but some of constraints were not followed
  - Merging algorithm has been well defined with considering volume capacity of vehicle.
  - It failed to find the optimal solution but algorithm has been fully constructed.
  - AI Agent showed better performance than singular LLM even though it was based on gpt-4o-mini, which is lighter model.



Compare gpt-4o-mini vs Our AI Agent





Result

>

1. Improved Generated Code Quality
2. Access to External Information
3. Minimize human intervention

Implications & Future outlook