AiDD

2024 AI+研发数字峰会

AI+ Development Digital summit

AI驱动研发变革 促进企业降本增效

北京站 08/16-17

大模型驱动的多智能体协同初探

钱忱 清华大学

演讲嘉宾

钱忱

清华大学 博士后

清华大学软件学院博士，现于清华大学自然语言处理实验室（THUNLP）担任博士后，清华大学水木学者，主要研究方向为预训练模型、自主智能体、群体智能；合作导师为孙茂松和刘知远教授，曾在ACL、SIGIR、ICLR、AAAI、CIKM等人工智能、信息管理、软件工程等相关的国际学术会议或期刊上以第一作者身份发表论文数篇。在群体智能方面，主持发布了大语言模型驱动的群体协作框架ChatDev、群体共学习范式Co-Learning、群体协同网络MacNet，面向任务完成和社会模拟的多智能体平台AgentVerse等。

目录
CONTENTS

PART 01
从大模型走向自主智能体

大模型驱动的自主智能体

- 基础大模型作为智能应用技术的内核，必须以**自主智能体作为载体**与**动态环境进行交互**，才能充分胜任动态复杂的智能应用，因此亟需发展大模型驱动的自主智能体技术

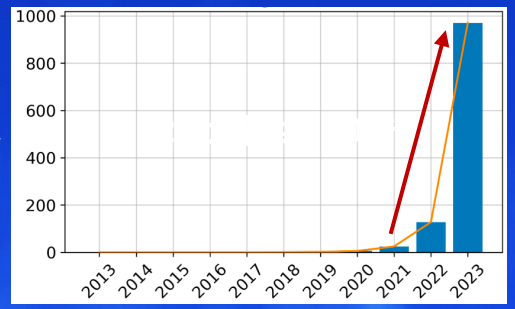
人工智能将进入智能体时代

自主智能体是OpenAI核心战略之一

2023 年 11 月 6 日，OpenAI在第一届开发者大会上正式提出构建自主智能体及相关生态的**战略目标**

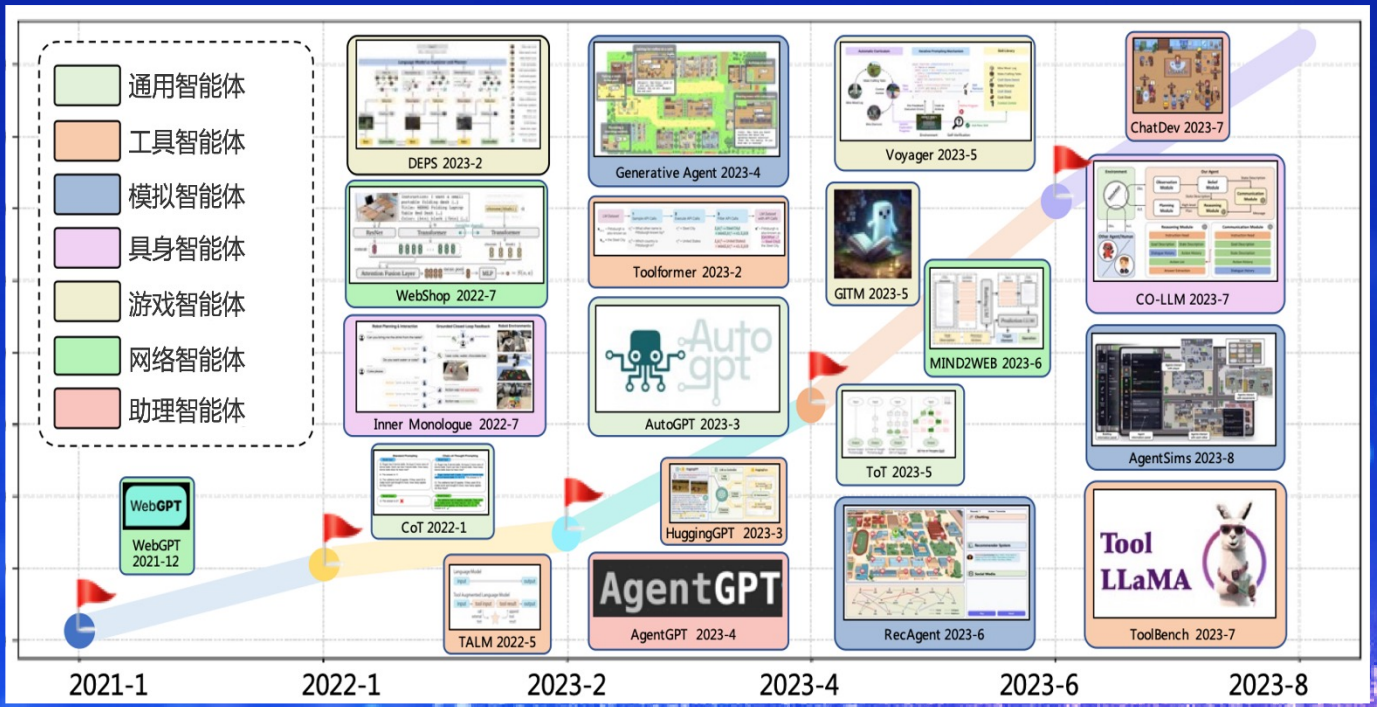
自主智能体相关研究**爆发式增长**

自主智能体相关研究在大模型带动下迅速成为人工智能研究必争之地，**2023年自主智能体研究论文数量呈井喷趋势**



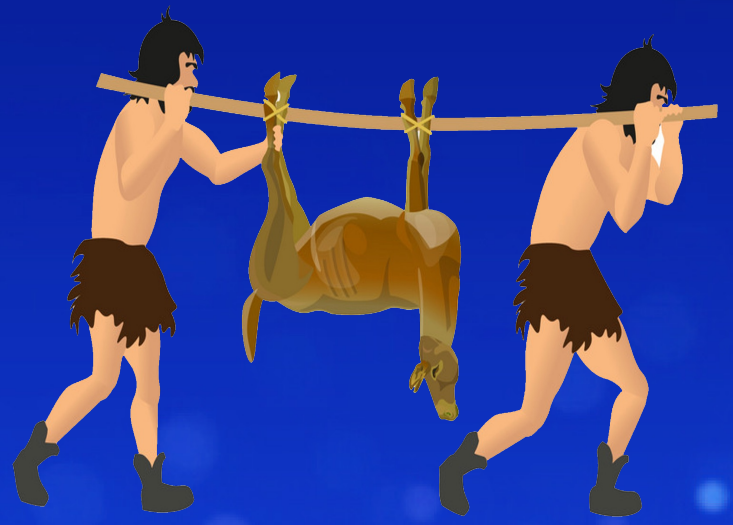
现处于争夺自主智能体技术高地的**重要战略窗口期**

目前自主智能体技术呈现百花齐放、多面竞争态势，**我国与人工智能领域国际领先研发机构**（OpenAI、微软、谷歌、Meta、斯坦福大学等）**处于同一起跑线**



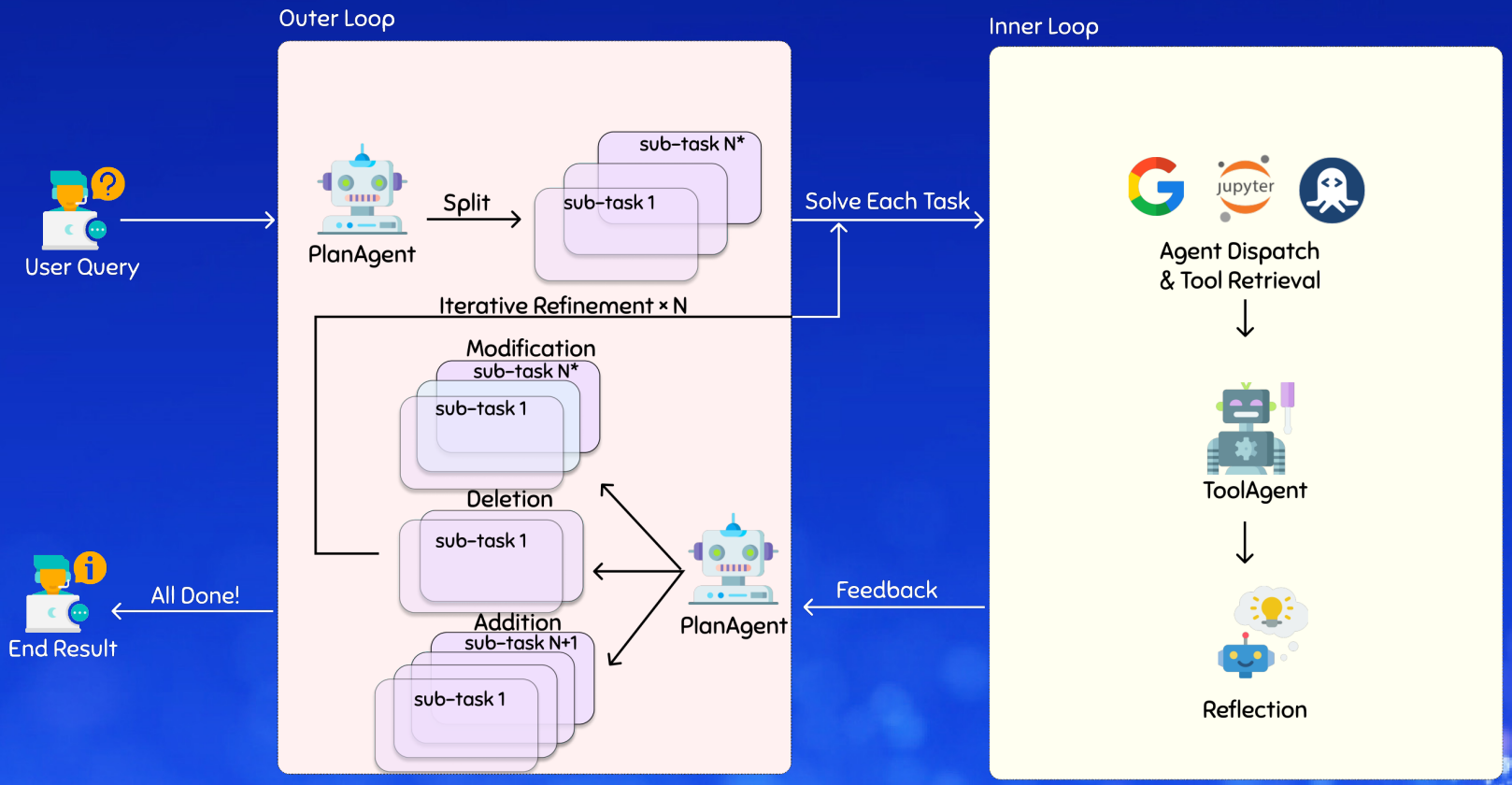
人类智能与人工智能

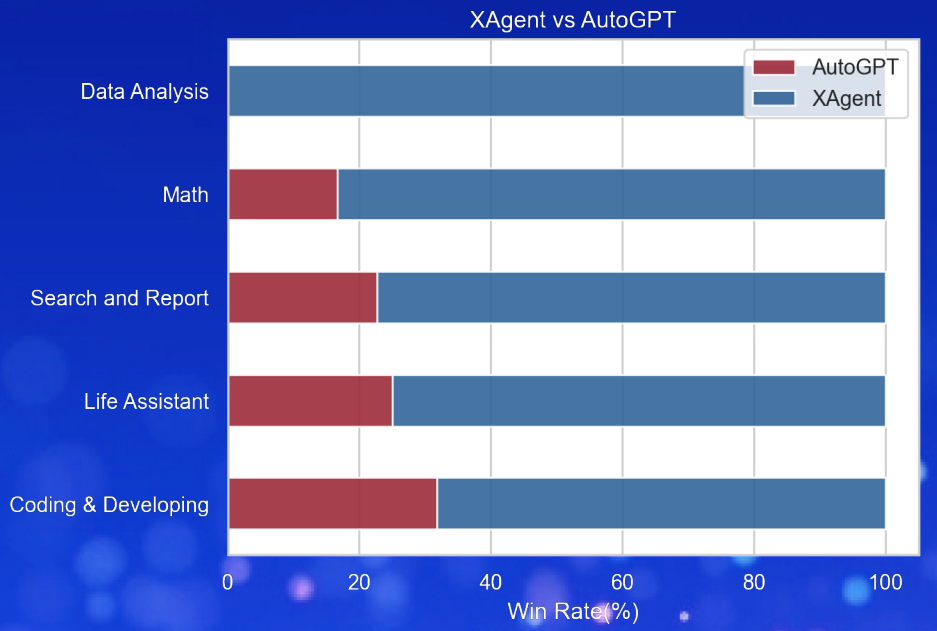
- 人工智能很可能即将走跟人类智能一样的发展路径

人类智能				
	脑容量小	脑容量大	工具使用	群体协作
人工智能	小模型	大模型	智能体	群体智能

XAgent：大模型驱动的自主智能体框架

- XAgent通过**双循环机制**协调决策制定和任务执行过程：外循环规划、内循环执行
- 外循环处理任务的高级管理和分配，内循环专注每个子任务的低级执行和优化





PART 02
大模型多智能体系统

多智能体系统的两种基本类型

任务完成型

清华ChatDev数字团队：基于语言交互的智能体数字公司，实现群体协作式软件开发

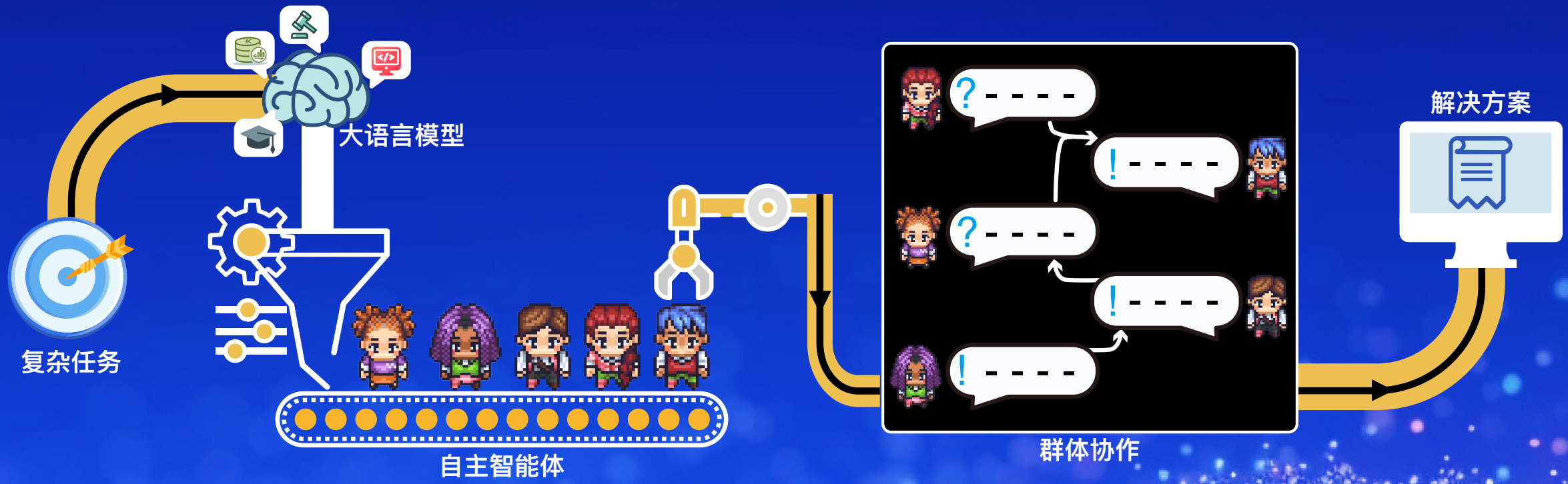
社会模拟型

斯坦福SmallVille小镇：基于层次规划的智能体社会小镇，实现人类社群行为的可信模拟

Chen Qian, et al. ChatDev: Communicative Agents for Software Development. In ACL-2024.
Park J S, et al. Generative agents: Interactive simulacra of human behavior. In UIST-2023.

多智能体自主交互：任务解决新范式

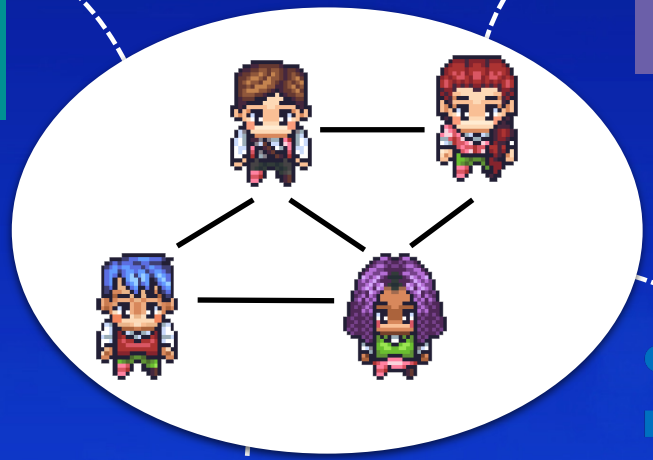
- 群体协作实现**多方优势互补**、配合无间，提高解决问题的**准确率**
- **语言交互**搭建沟通桥梁、缓解信息茧房，通过**任务分工**和**协作执行**处理复杂任务



多智能体系统核心研究框架

- **无目标导向型**：群体自由规划及行为执行（e.g., 社会模拟）
- **目标导向型**：制定群体特定目标以协同进行任务完成（e.g., 软件开发）

任务目标



社会组织

- 联盟结构：独立结构、层次化结构、中心化结构、全连接结构
- 组织规模：小规模、中规模、大规模

组织管理

- **行为关系**：合作关系、竞争关系
- **组织行为**：激励行为、团队凝聚力、劳动力多元化、资源竞争、同龄人压力、群体懈怠、评价焦虑、情绪感染等

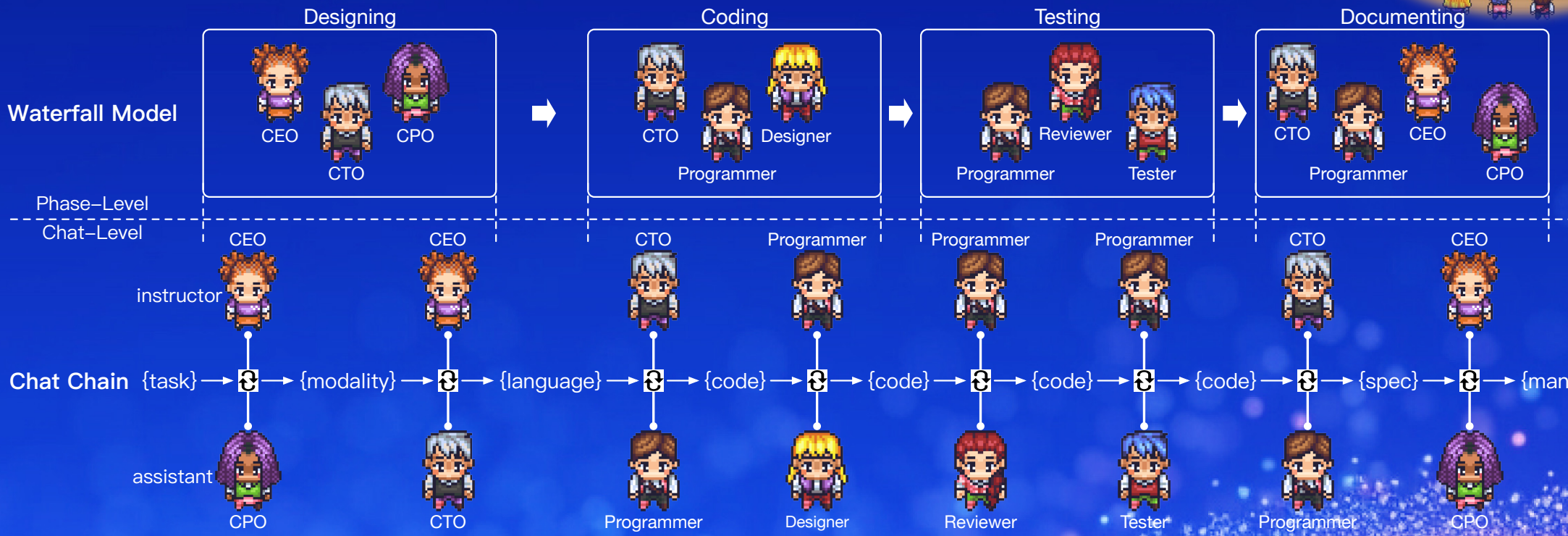
行为路由

- 顺序型：各子组织间依次行动
- 互斥型：各子组织间选择性行动
- 并发型：各子组织间并行行动

PART 03
多智能体协同及演化

ChatDev – 交互链进行编排

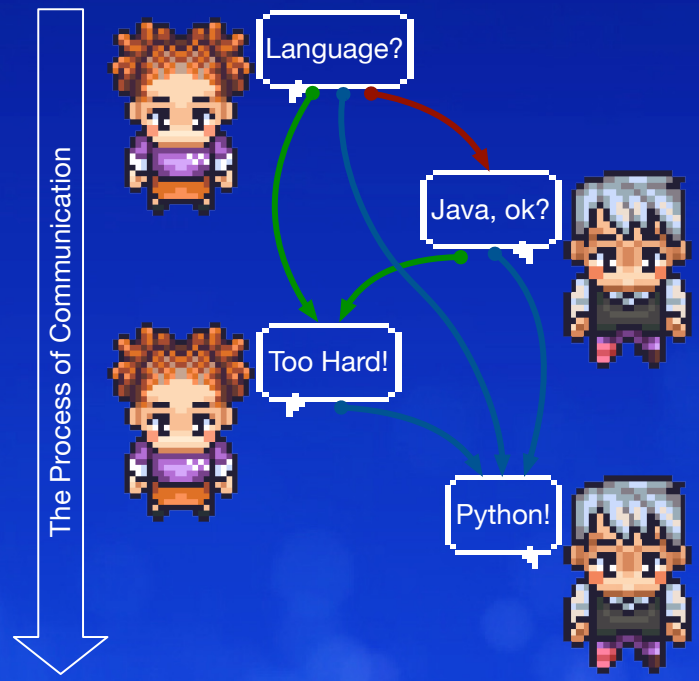
- 通过角色扮演交流实现智能体间的**方案提议**和**决策研讨**过程
- 交流链将复杂任务分解为由原子任务组成的 “**方案生产线**”

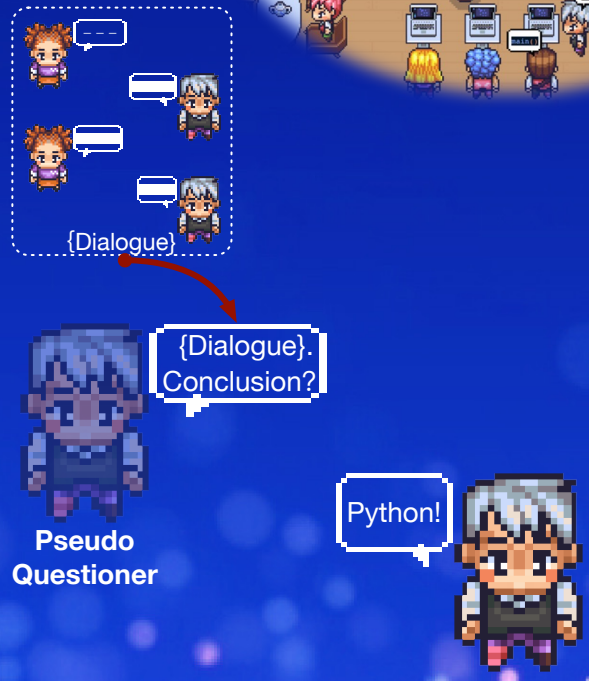


Chen Qian, et al. ChatDev: Communicative Agents for Software Development. In ACL-2024.

ChatDev – 交互机制

- 通过角色化、记忆流、自反思机制，实现自上下文感知的主交互





Chen Qian, et al. ChatDev: Communicative Agents for Software Development. In ACL-2024.

ChatDev – 交互模式

- 通过交互除幻机制（Communicative Dehallucination）缓解编码幻觉（Coding Hallucination）

Chen Qian, et al. ChatDev: Communicative Agents for Software Development. In ACL-2024.

多智能体协同实现“降本增效”

- 软件制作平均**时间小于7.0分钟**且制作**成本约$0.3美元**

Code Complete

: CTO (Chief Technology Officer) : Programmer

: **Unimplemented File**

```
sales_report.py
```

In our software, each file must strictly follow a markdown code block format, where the following tokens must be replaced such that "FILENAME" is the lowercase file name including the file extension, "LANGUAGE" in the programming language, "DOCSTRING" is a string literal specified in source code that is used to document a specific segment of code, and "CODE" is the original code:
FILENAME
“‘ LANGUAGE
”’
DOCSTRING
”’
CODE
“‘
As the Programmer, to satisfy the complete function of our developed software, you have to implement all methods in the `sales_report.py` file which contains an unimplemented class. Now, implement all methods of the `sales_report.py` and all other codes needed, then output the fully implemented codes, strictly following the required format.

: **sales_report.py**

```
...
def __init__(self, sales_data):
    self.sales_data = sales_data
def generate(self):
    self.generate_summary_report()
    self.generate_sales_by_product_category_report()
def generate_summary_report(self):
...
```

Code Diff on sales_report.py

```
 def generate(self):
-    pass
+    self.generate_summary_report()
+    self.generate_sales_by_product_category_report()
```

Code Review (Round 1)

: Reviewer : Programmer

: {Codes of current version}
<Comment>
The highest priority comment is that the `get_customer_lifetime_value` method in the `SalesData` class is currently returning the same value as the `get_customer_acquisition_cost` method. This is incorrect as the customer lifetime value should be calculated differently.
To fix this, you need to update the logic in the `get_customer_lifetime_value` method to calculate the average sales per customer over the lifetime of the business. You can do this by dividing the total revenue by the total number of customers.
Here's the revised code for the `get_customer_lifetime_value` method:

sales_data.py

```
def get_customer_lifetime_value(self):
    if len(self.data) == 0:
        return 0
    total_sales = sum(data["sales_figure"] for data in self.data)
    total_customers = len(set(data["customer_name"] for data in self.data))
    return total_sales / total_customers
```

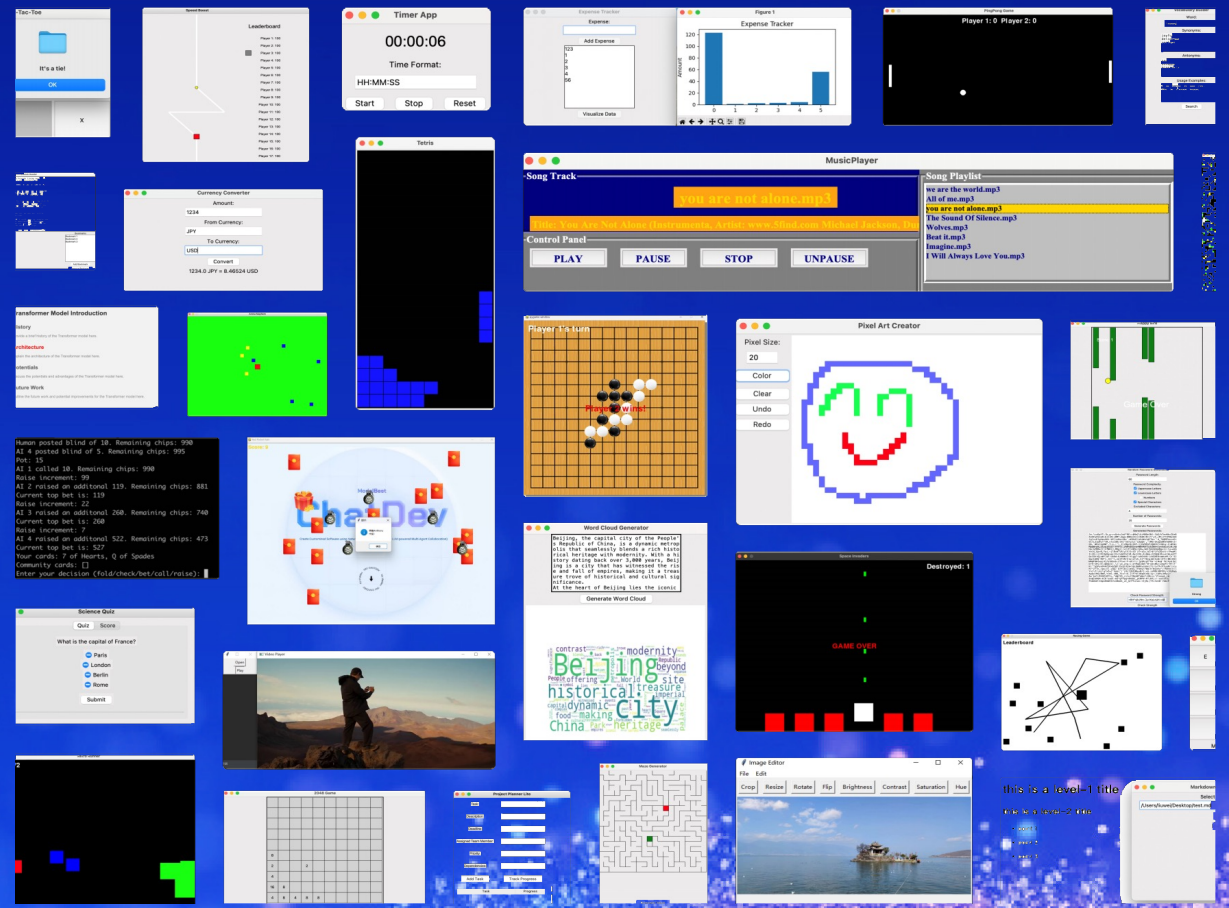
Please make this change and ensure that the `get_customer_lifetime_value` method returns the correct value.

: **sales_data.py**

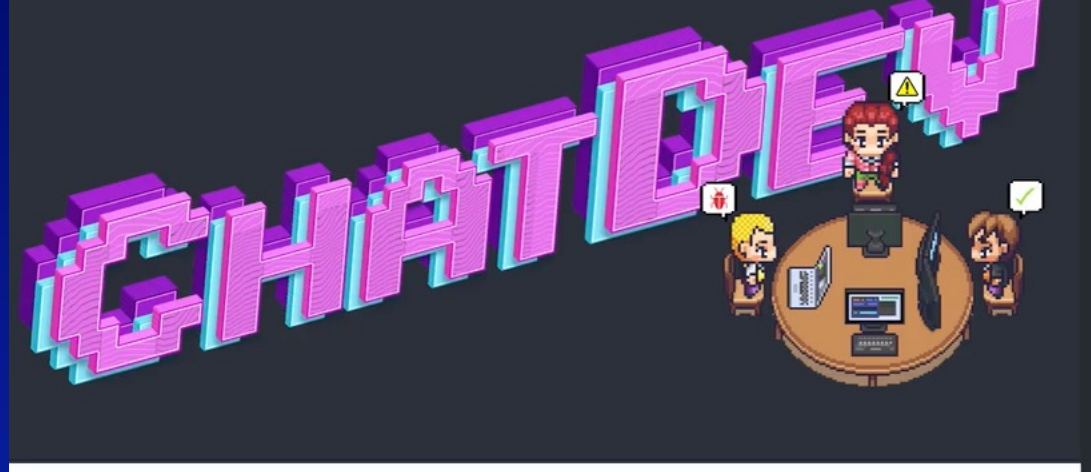
```
...
def get_customer_acquisition_cost(self):
    if len(self.data) == 0:
        return 0
    total_sales = sum(data["sales_figure"] for data in self.data)
    total_customers = len(self.data)
    return total_sales / total_customers
def get_customer_lifetime_value(self):
    if len(self.data) == 0:
        return 0
    total_sales = sum(data["sales_figure"] for data in self.data)
    total_customers = len(set(data["customer_name"] for data in self.data))
    return total_sales / total_customers
...
```

Code Diff on sales_data.py

```
 def get_customer_lifetime_value(self):
     if len(self.data) == 0:
         return 0
     total_sales = sum(data["sales_figure"] for data in self.data)
-    total_customers = len(self.data)
+    total_customers = len(set(data["customer_name"] for data in self.data))
```



Chen Qian, et al. ChatDev: Communicative Agents for Software Development. In ACL-2024.

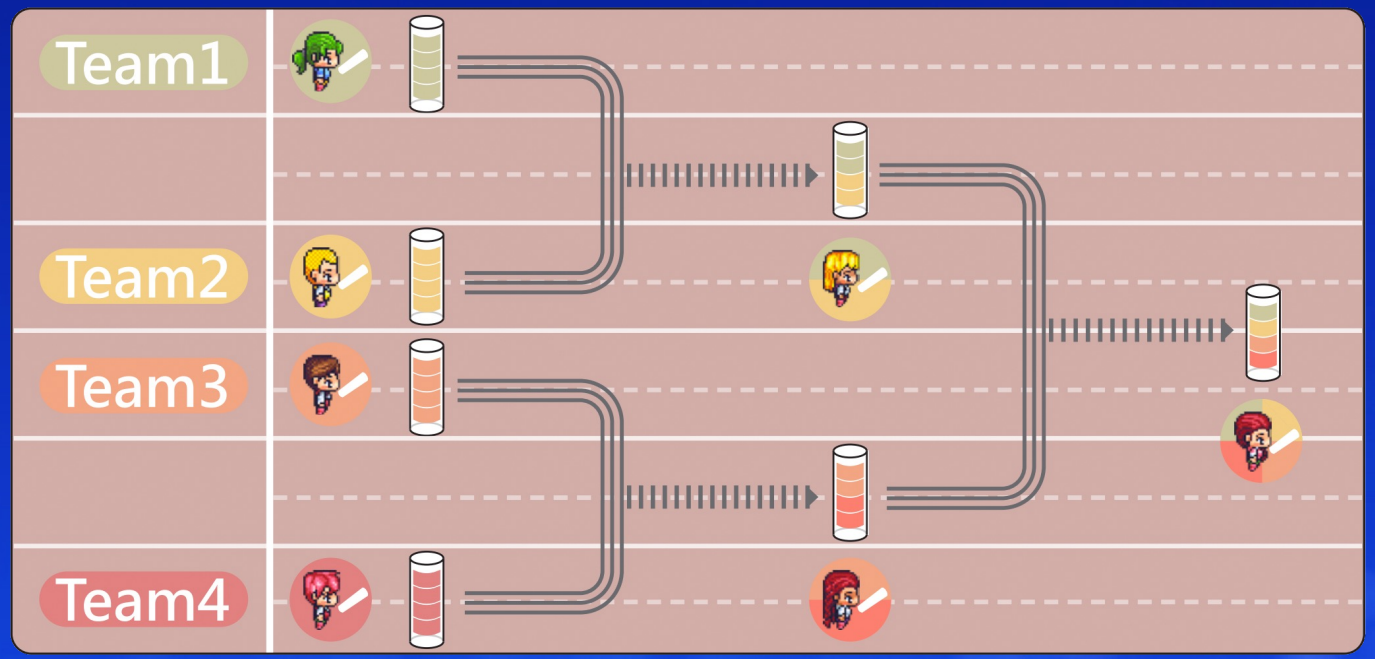


Replaying Speed

version_updates num_code_files num_png_files num_doc_files

CTC：从单队伍到多队伍的协作编排

- 为了超越单一团队执行的偏差，多队伍能够有效地从多方获取见解、促进跨团队间的交互，进而产出更优质的内容



Zhuoyun Du, Chen Qian, et al. Multi-Agent Software Development through Cross-Team Collaboration. In arXiv:2406.08979

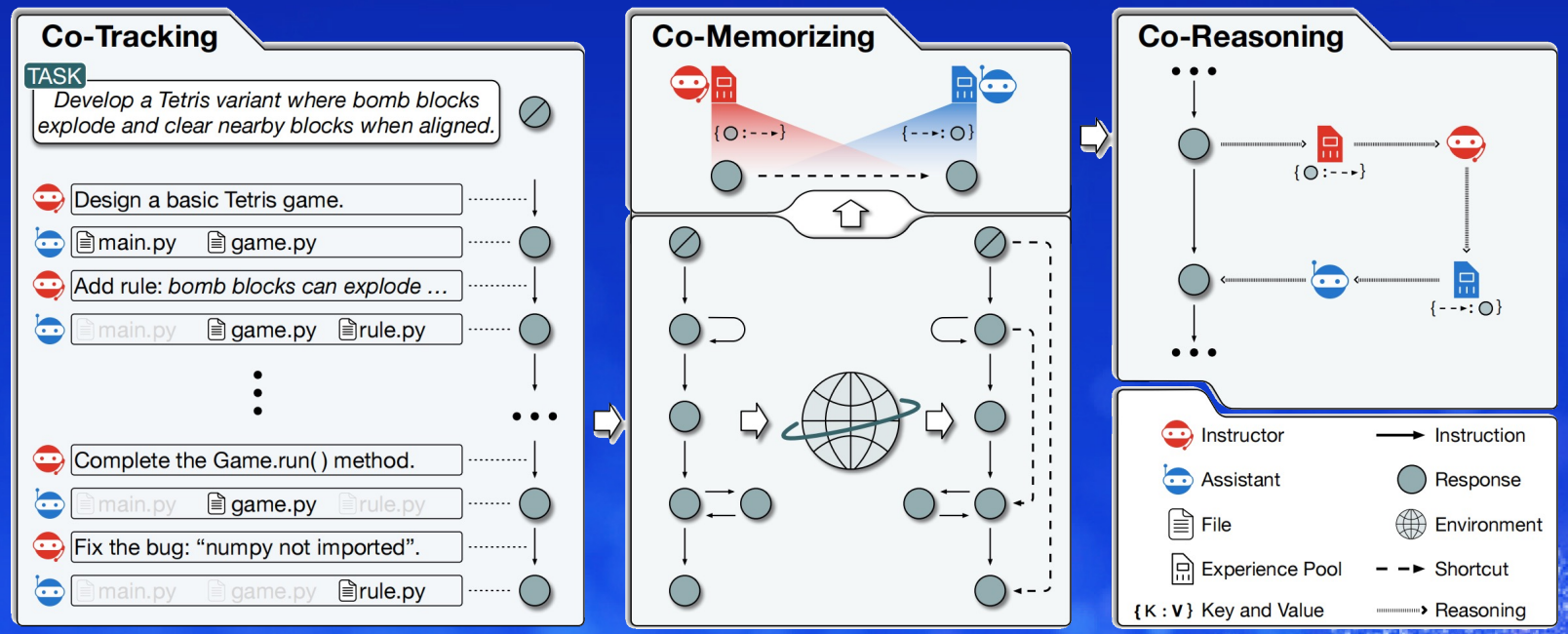
跨任务间的静态流程限制了推理效率

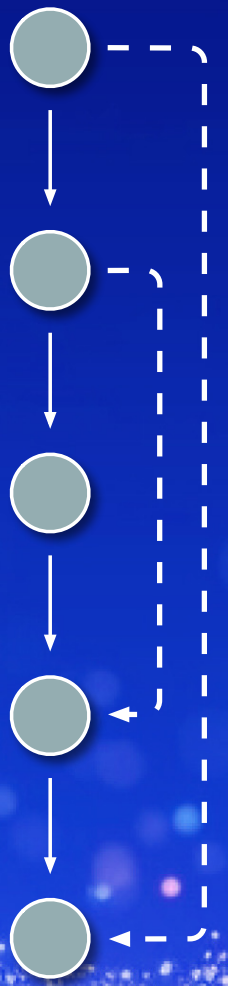
- 本质原因：智能体缺乏**跨任务**的过往经验

Qian C, Dang Y, et al. Experiential co-learning of software-developing agents. In ACL-2024.

Co-Learning：智能体共同经验习得

- 核心思路：跨任务**经验**迁移
  - 共同实践：训练任务上进行“彩排”，形成过往**执行轨迹**
  - 共同记忆：对执行轨迹进行“**捷径**”抽取和记忆
  - 共同推理：测试任务上利用捷径记忆实现**经验化推理**



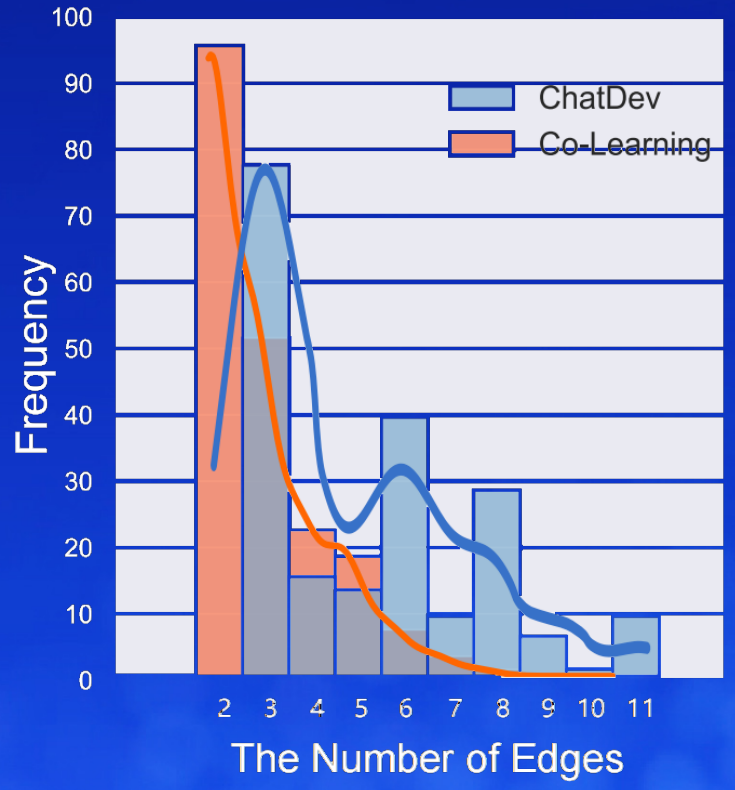


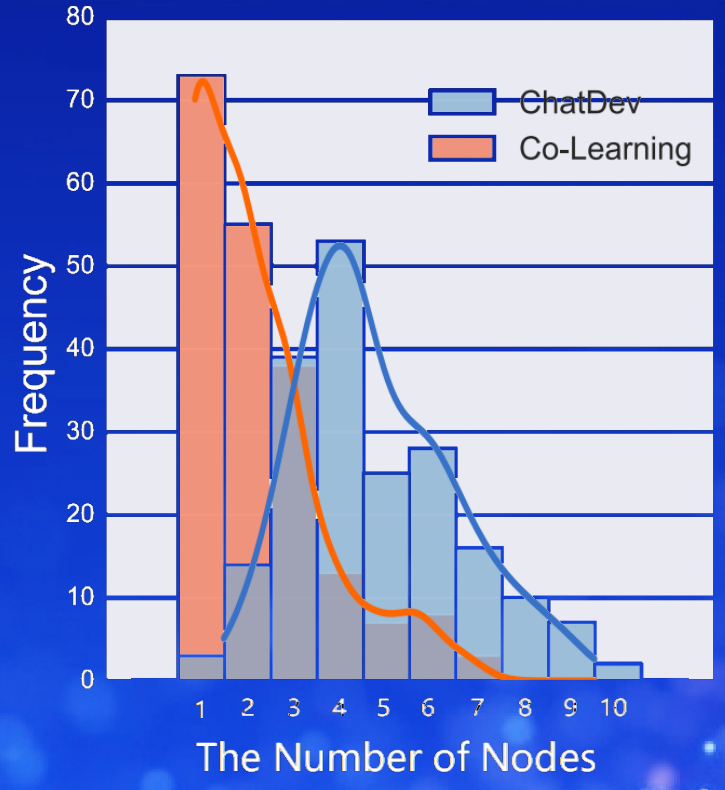
捷径作为经验

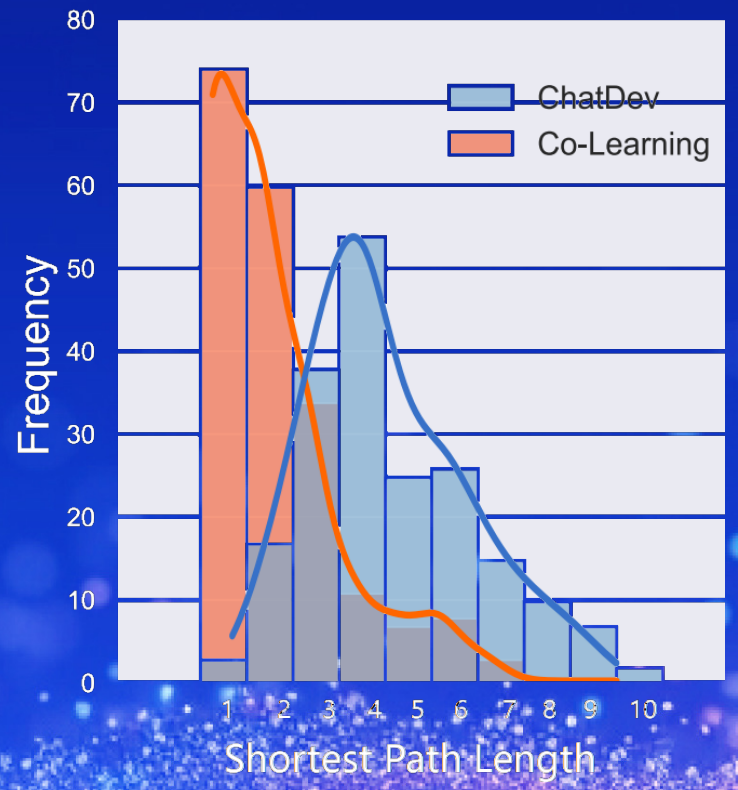
Qian C, Dang Y, et al. Experiential co-learning of software-developing agents. In ACL-2024.

Co-Learning：多智能体经验化共同学习框架

- 在经验化的任务执行过程中，智能体随着经验的积累实现**推理步骤的显著减少**，即以更少的步骤实现了更高质量的成品，实现群体推理过程的“**降本增效**”



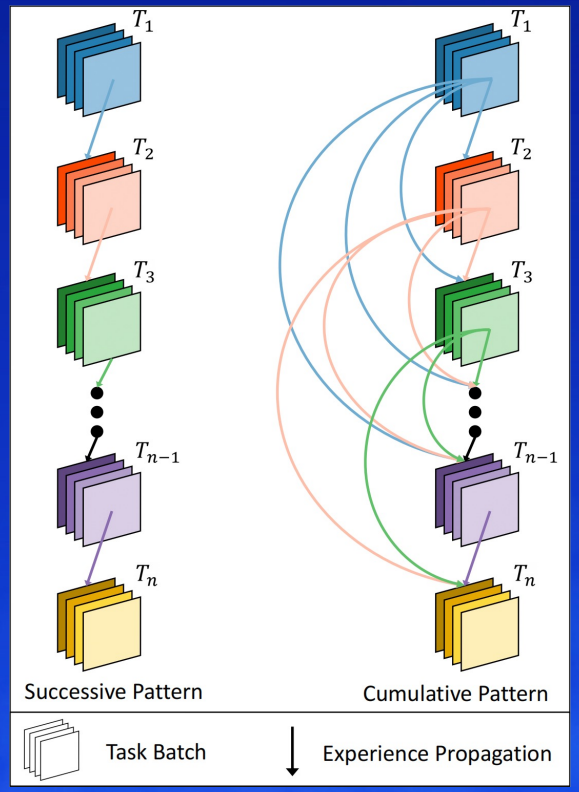




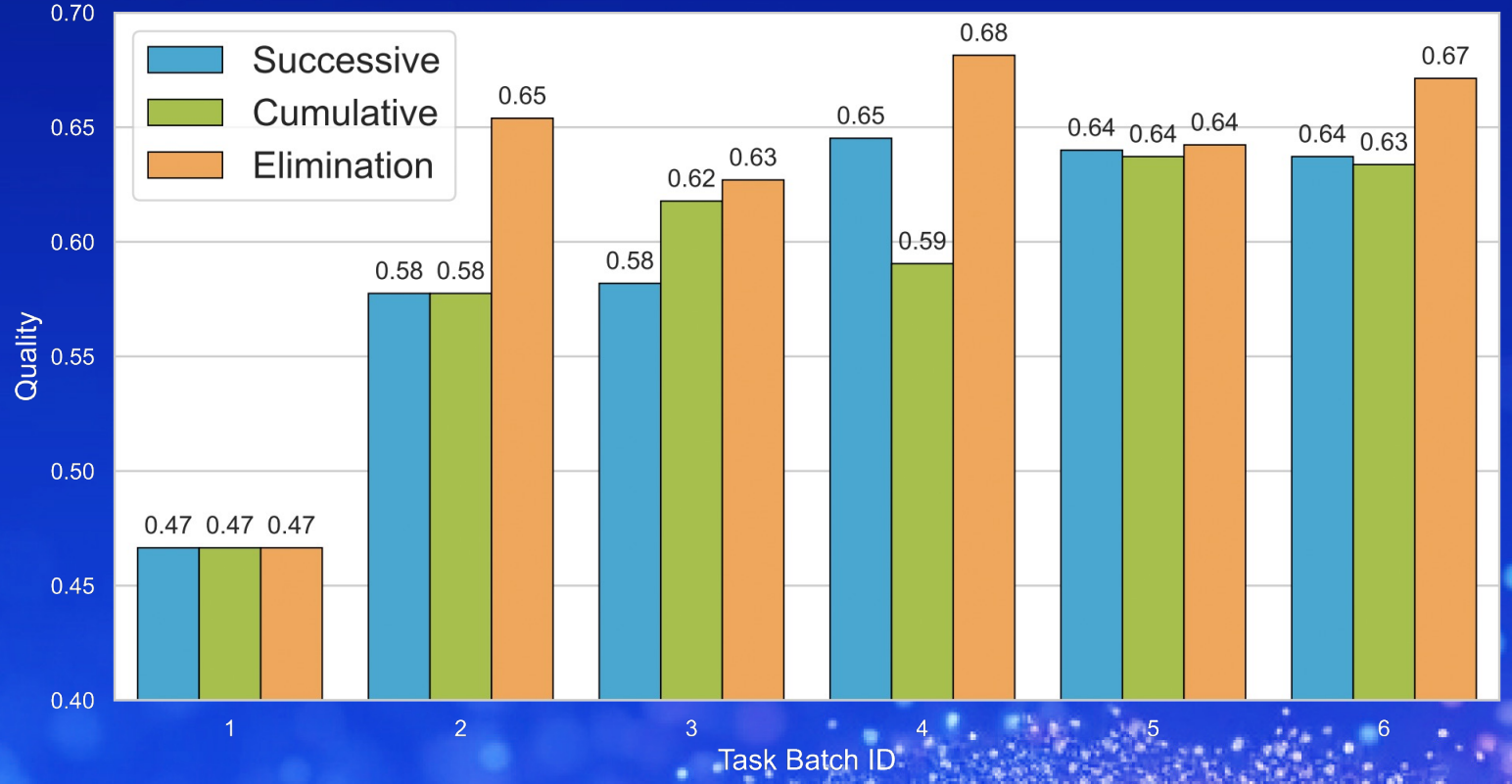
Qian C, Dang Y, et al. Experiential co-learning of software-developing agents. In ACL-2024.

Co-Evolving：经验的迭代优化

- 随**任务批次**的陆续执行，在Co-Learning积累**静态经验**的基础上，进行相继式和积累式两类**经验动态编辑**



捷径经验、相继式和积累式经验积累

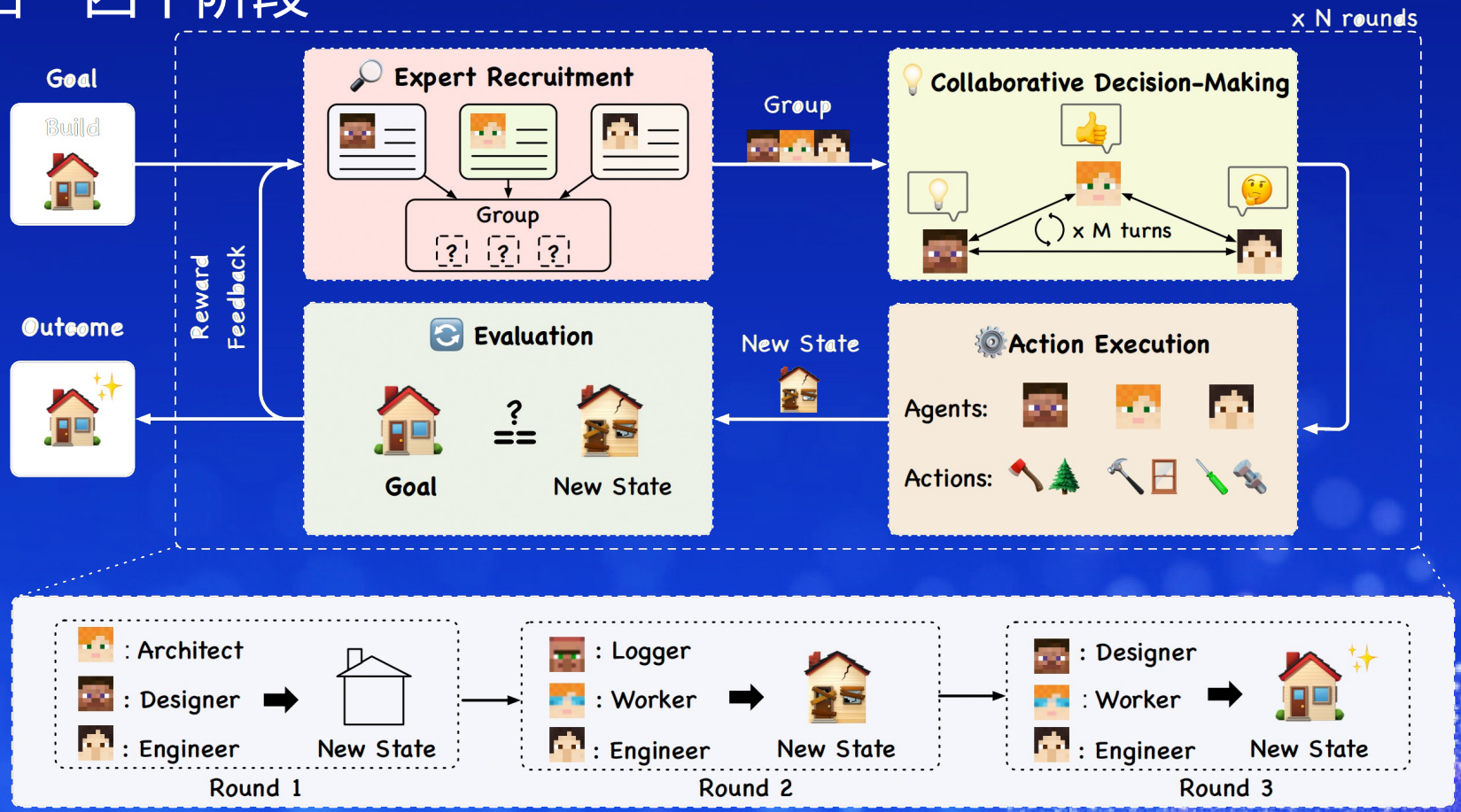


任务执行过程的协同效率随经验的动态积累而逐步提升

Qian C, Li J, et al. Iterative Experience Refinement of Software-Developing Agents. In arXiv:2405.04219

AgentVerse：通用多智能体平台

- 大模型群体协作的通用流程，包含“智能体招募”、“协同决策”、“动作执行”与“检验评估”四个阶段

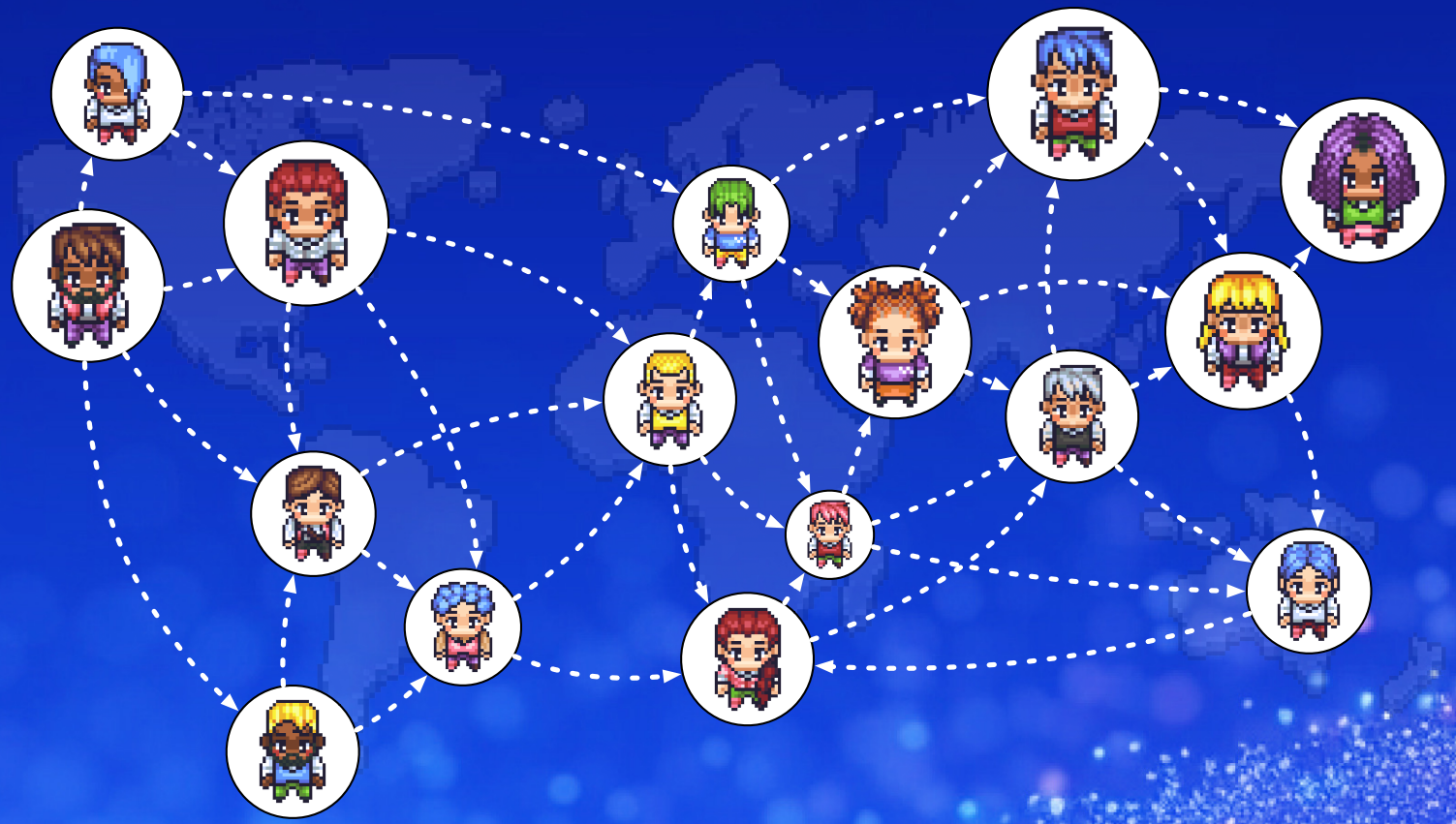


Weize Chen, Yusheng Su, et al. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors. In ICLR-2024.

PART 04
智能体协同的缩放法则

如何组织构建高可拓展的群体协同?

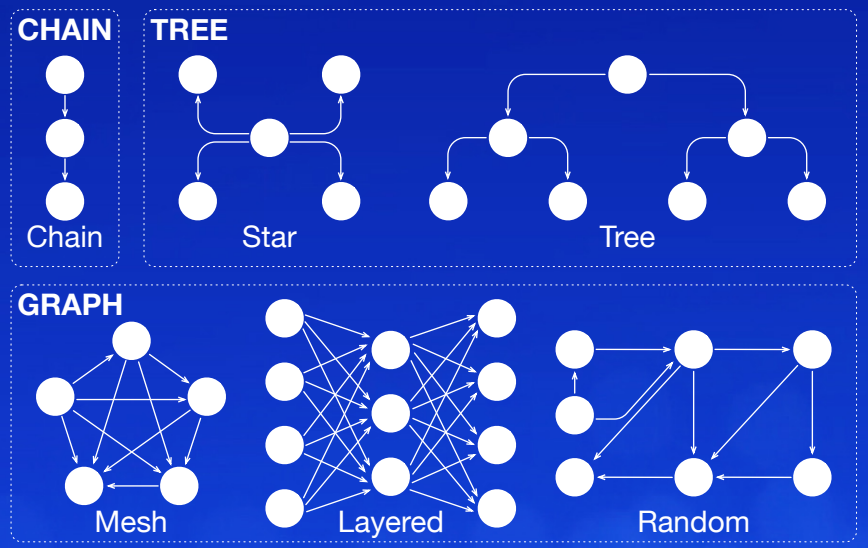
- 核心挑战在于设计一个**通用的组织结构**，实施**合理的路由策**略，并建立**有效的记忆管理**，以**实现高效和可扩展的群体协同**



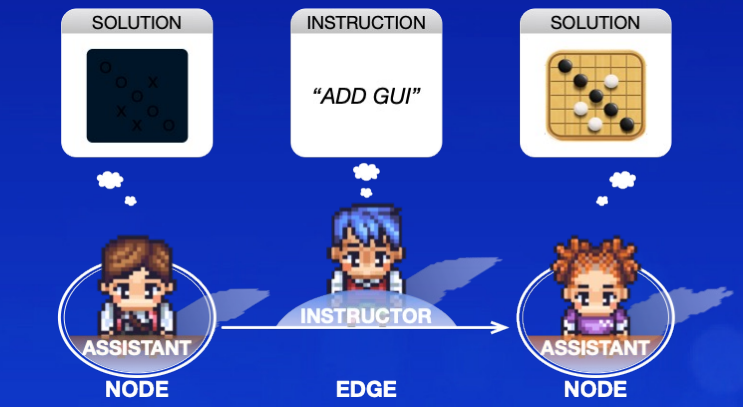
Qian C, Xie Z, Wang Y, et al. Scaling Large-Language-Model-based Multi-Agent Collaboration. In arXiv:2406.07155

MacNet：多智能体协作网

- 在**有向无环图**的拓扑之上**部署智能体**（**节点**上部署**执行者**、**边**上部署**发令者**），形成**多智能体协作网**（**M**ulti-**A**gent **C**ollaboration **Net**works，MacNet）



有向无环的拓扑结构



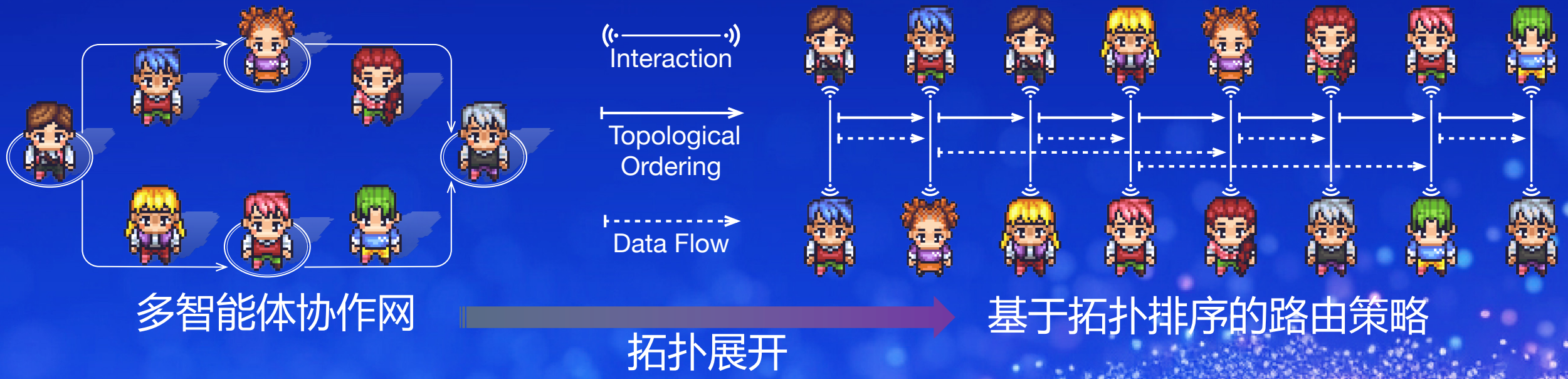
智能体部署

多智能体协作网

Qian C, Xie Z, Wang Y, et al. Scaling Large-Language-Model-based Multi-Agent Collaboration. In arXiv:2406.07155

MacNet：多智能体协作网

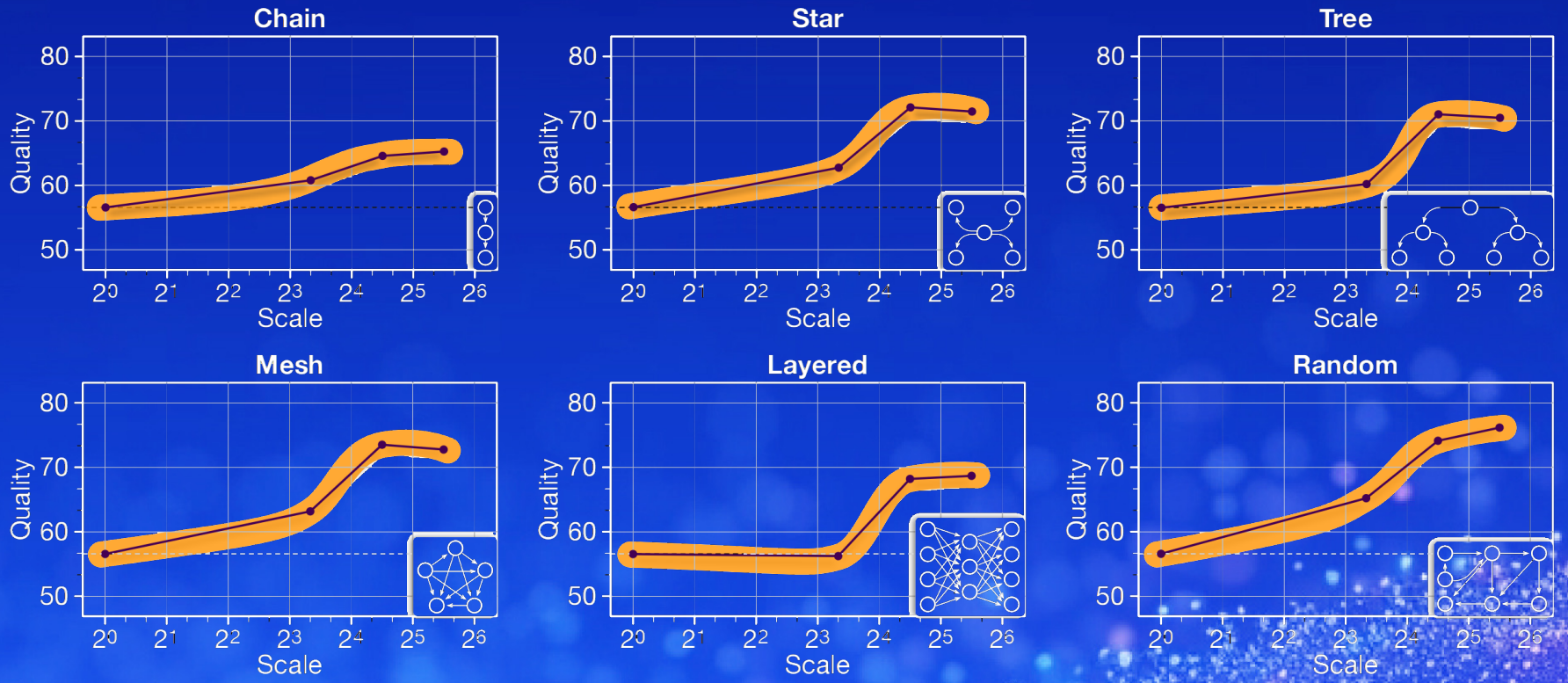
- 通过**拓扑排序**进行多智能体协作网的**遍历**，“展开”成智能体交互的**路由次序**
- 网络上仅传播交互后的**解决方案**（而非全程对话），构筑可拓展的**记忆管理**机制
  - 解决方案可从逻辑推理，跨到软件代码、连贯故事、法律文书等异构情景



Qian C, Xie Z, Wang Y, et al. Scaling Large-Language-Model-based Multi-Agent Collaboration. In arXiv:2406.07155

MacNet：多智能体协作网

- 支持大规模协同：可支持**多种异构拓扑**，甚至**承载上千个智能体协同工作**
- **小世界协同现象**：越接近小世界网络属性的拓扑，其综合性能更优越
- **协同缩放法则**：性能大致遵循**Sigmoid形**趋势，相较神经缩放法则**更“早”被观测**

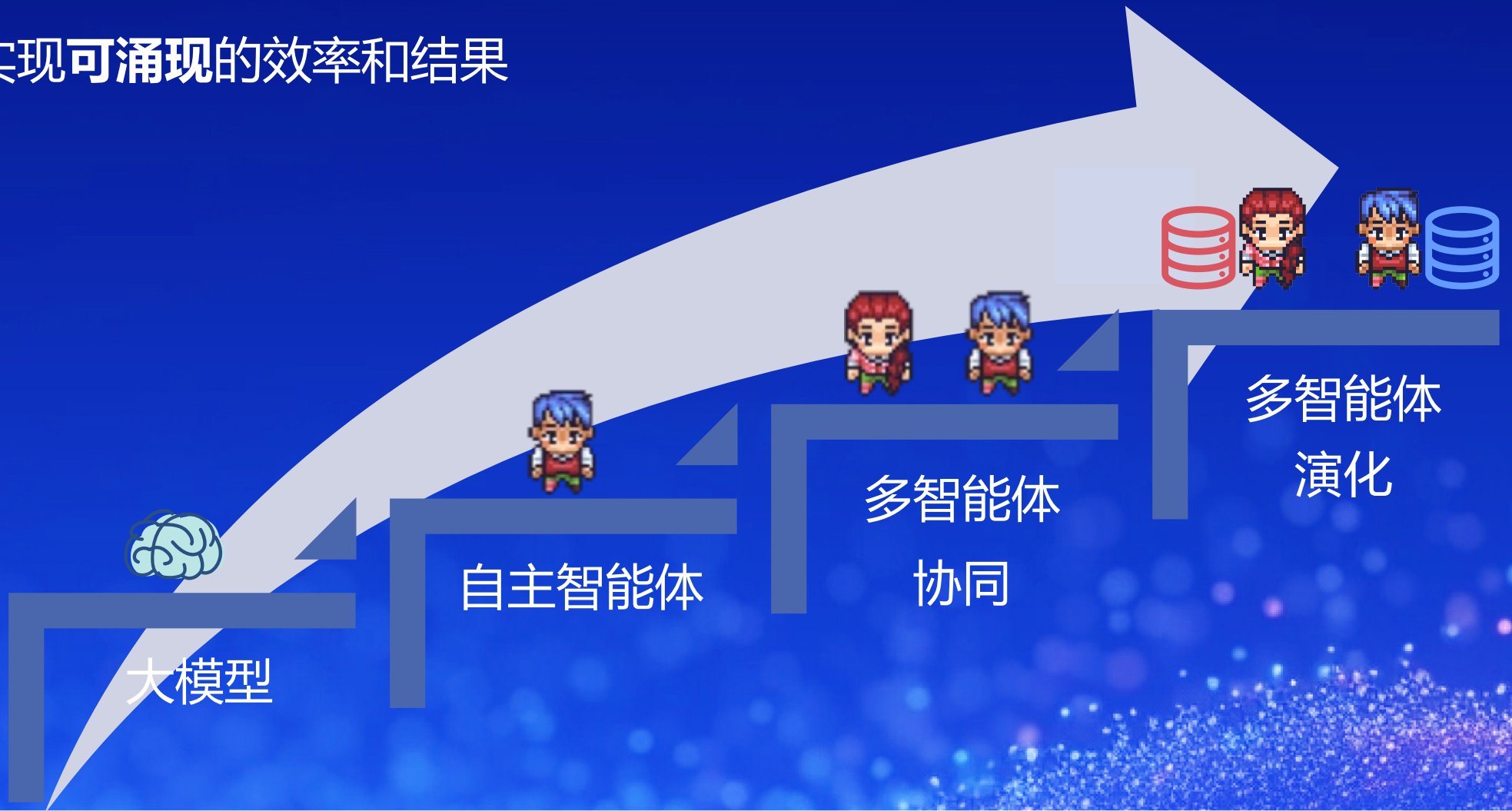


Qian C, Xie Z, Wang Y, et al. Scaling Large-Language-Model-based Multi-Agent Collaboration. In arXiv:2406.07155

PART 05
总结与展望

大模型智能体发展趋势

- 进阶智能体的愿景是从独立的实体进化为**可协作**和**可演化**的系统，通过**集体智慧**实现**可涌现**的效率和结果



可行方向


分工与合作
安全与检测
协商与决策
反馈与进化
角色与职责
文化与秩序
制度与规范
认知与共鸣
CHATDEV
IR:
AK:
ML:
KM:

拥抱“数字孪生，万物智联”的未来世界

Internet

信息高速，全球连接

全球信息高速公路，
将每一位**网民紧密相连**

Internet of Things

设备互联，生活便捷

物体间的沟通桥梁，
使每一件**设备灵活互联**

Internet of Agents

数字孪生，万物智联

交织万物的网络纽带，
绘制**数字孪生**的未来蓝图

相关资料

系统

https://github.com/OpenBMB/ChatDev

纲要

https://thinkwee.top/multiagent_ebook

Task Solving | Social Simulation

Communication | Organization | Evolvement

Profile | Memory | Planning | Tool

Foundation Models

§1: Communication

facilitating agent communication

Read

§2: Organization

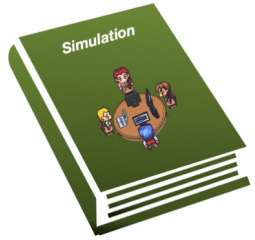
organizing agents effectively

Read

§3: Evolution

growing capabilities over time

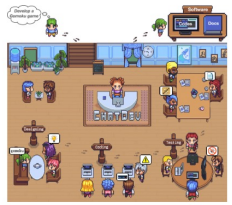
Read



§4: Simulation

simulating societal dynamics

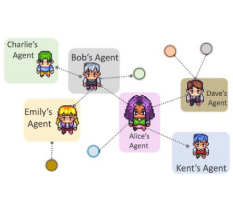
Read



ChatDev

Multi-Agent Collaboration for Software Development

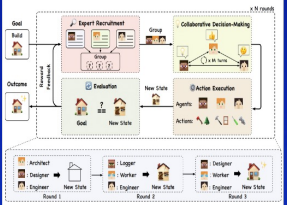
Paper | Code



iAgents

Bijective Social Networks of Humans and Agents

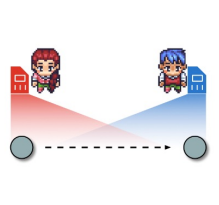
Paper | Code



AgentVerse

General-Purpose Multi-Agent Framework

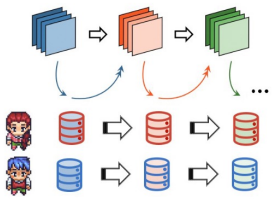
Paper | Code



Co-Learning

Cross-Task Experience Co-Leaning for Mutual Growth

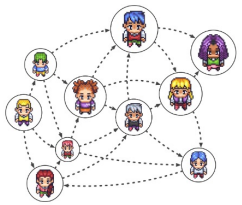
Paper | Code



Co-Evolving

Continuous Experience Refinement over Time

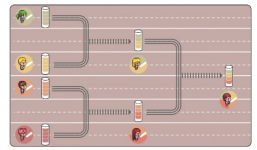
Paper | Code



MacNet

Exploring Collaborative Scaling Law

Paper | Code



CTC

Cross-Team Multi-Agent Orchestration

Paper | Code

ChatEval

Communication for Automated Evaluation

Paper | Code

THANKS