CARD：迈向多智能体拓扑结构的条件设计

Tongtong Wu[1, *, †]、**Yanming Li**[2, *]、**Ziye Tang**[2],**Chen Jiang**[2],**Linhao Luo**[1],
Guilin Qi[2],**Shirui Pan**[3],**Gholamreza Haffari**[1]1莫纳什大学 [2]东南大学 [3]格里菲斯大学

arXiv:2603.01089v1 [cs.CL] 1 Mar 2026

摘要

基于大语言模型（LLM）的多智能体系统在代码生成和协同推理等任务中展现出强大的能力。然而，这些系统的有效性和鲁棒性在很大程度上取决于其通信拓扑，而该拓扑往往是固定的或通过静态方式学习得到，因而忽略了模型升级、工具变更或知识来源变化等现实动态因素。为解决这一局限性，我们提出 CARD（条件智能体图设计器），这是一种用于实现 AMACP（自适应多智能体通信协议）的条件图生成框架。CARD 将动态环境信号明确纳入图构建过程，从而支持训练阶段和运行时的拓扑适应。借助条件变分图编码器和环境感知优化，CARD 能够生成有效且具有韧性的通信结构，以应对模型能力或资源可用性的变化。在 HumanEval、MATH 和 MMLU 上的实证结果表明，CARD 始终优于静态基线和基于提示的基线，在多种条件下实现了更高的准确率和鲁棒性。源代码见：https://github.com/ Warma10032/CARD。

1 引言

由大语言模型（LLM）（OpenAI，2024a；Chen，2024）驱动的多智能体系统（Liu 等人，2025；Li 等人，2026）近期在各种复杂任务中展现出卓越能力，涵盖从代码合成（Chen 等人，2023）到协同推理（Liu 等人，2023）。通过将各模型的内部知识、自然语言生成和推理能力与外部工具（Zhang 等人，2024a；b）或同伴大语言模型相结合，这些系统能够有效地分解问题（Yao 等人，2023）、协调子目标（Liang 等人，2024），并整合多样的信息来源（Lee 等人，2023；Min 等人，2025）。然而，通信拓扑规定了智能体之间的互连方式，对系统性能有显著影响，关系到解决方案的质量以及系统对模型升级（Wu 等人，2024a）、应用程序接口修改（Wu 等人，2026；2024b）和数据源波动（Min 等人，2025）等变化条件的鲁棒性。

当前的拓扑设计方法通常分为两类（Bei 等人，2025；Liu 等人，2025）。许多系统依赖手工设计的流程（Hong 等人，2024）或预定义智能体序列（Wu 等人，2023），这些方法在稳定且充分了解的场景中表现良好，但缺乏适应性。相较之下，近期方法通过反向传播“文本梯度”（Zhuge 等人，2024）或借助可微模块对智能体间连接进行参数化（Zhang 等人，2025a），自动学习通信结构。

[1*] 这些作者对本研究的贡献相同。[2†] 通讯作者：tongtong.wu@monash.edu。

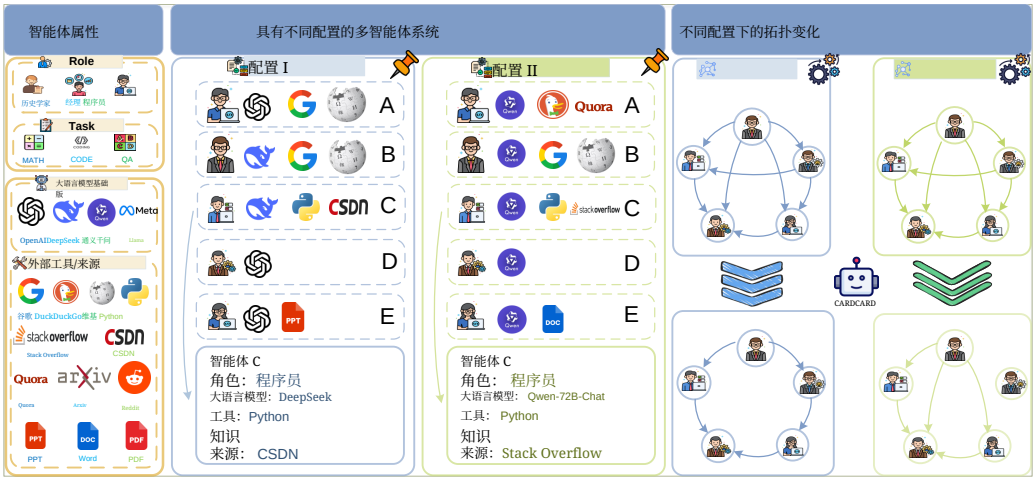


图 1：两种环境配置下的智能体属性及相应通信拓扑，展示了拓扑由任务需求、模型库的能力和可用资源共同决定。

然而，这些学习型拓扑通常假设环境是静态的，未能考虑瞬时外部因素。因此，当条件发生变化时，例如升级模型基础（如 GPT-4o $\rightarrow$ GPT-5）、工具可用性发生变化，或数据源质量下降，静态拓扑或朴素学习得到的拓扑就会变得脆弱，导致冗余交互或信息流中断（见图 1）。

自适应大语言模型多智能体通信协议（AMACP）：

给定任务/查询 q， q 的最优通信拓扑应满足以下协议逻辑： **有效性：** 通信结构必须能够有效地产生满足给定任务 q要求的合格解。**成本效益：** 通信结构应在给定条件下，尽量降低解决 q 所需的总体资源消耗（例如模型使用量、*API* 调用次数和令牌成本）。**自适应性：** 通信结构应根据条件变化动态调整，确保在各种可用性情况下保持稳健。

我们通过形式化定义自适应多智能体通信协议（AMACP），并借助条件智能体图设计器（CARD）将其具体实现，以弥补这一空白。CARD 是一个条件图生成框架，它（i）通过智能体的配置文件通道和条件通道来表示每个智能体，（ii）对动态环境信号进行编码，以及（iii）解码生成交互图，其边会在训练时和运行时随条件变化而调整，无需重新训练。该框架的目标是在任务效用与考虑条件的通信成本之间取得平衡，从而满足 AMACP 所要求的有效性、成本效益和自适应性。在 HUMANEVAL、MATH 和 MMLU 上进行实验，并模拟环境变化（模型升级、工具可用性变化、数据源扰动）后，CARD 相较于静态拓扑、仅依赖提示词的拓扑或朴素学习得到的拓扑均取得显著提升，同时在静态场景下仍具有竞争力。我们的主要贡献如下：

- 对 **AMACP** 的形式化定义：一种能够在动态外部条件下实现自适应多智能体通信的协议。
- 介绍**CARD**，这是一种条件图生成框架，可从环境状态中显式学习有效且自适应的智能体拓扑。

- 全面的实证验证表明，在动态条件下，CARD 的表现始终优于现有的固定拓扑和学习型拓扑基线。
- 对拓扑适应进行详细分析，阐明环境状态条件化如何提升多智能体协作的效率与稳健性。

2 相关工作

协作式大语言模型智能体。 早期关于基于大语言模型的多智能体通信的工作依赖于人工定义的协作流程，涵盖从非交互式查询和思维链提示到辩论框架以及固定的树状或基于图的结构（Wei 等人，2022年；Yao 等人，2023；Besta 等人，2024）。为减少手工设计这些流程的工作量，已开发出 GPT-Swarm（Zhuge 等人，2024）、G-Designer（Zhang 等人，2025a）和 Aflow（Zhang 等人，2025b）等自动化拓扑学习方法。这些方法通过可微模块或启发式搜索优化智能体连接，在静态环境中取得了优异表现。然而，它们仍然假设环境是平稳的（Hu 等人，2025；Wang 等人，2025），并且缺乏应对模型能力、工具访问权限或数据质量变化的机制（Wu 等人，2024a）。

作为图的多智能体。 尽管分布式系统文献中已经提出了一些动态通信协议（Wei 等人，2022年；Yao 等人，2023；Liang 等人，2024），但在不断变化的条件下，大多数学习型拓扑仍然是静态且脆弱的，例如模型升级、外部工具可靠性的波动或数据源质量的变化（Pareja 等人，2020；Chang 等人，2020）。在这些场景中，预定义或未经充分优化的图可能会产生冗余交互或受阻的信息流（Liu 等人，2025）。为弥合这一差距，我们引入了条件智能体图设计器，该设计器明确根据外部信号（例如模型版本、工具性能、数据源保真度）对图生成进行条件化，从而生成自适应且稳健的多智能体拓扑。

3 问题形式化

我们首先对基于大语言模型的多智能体系统（MAS）的拓扑和协议设计空间进行形式化，将 CARD 框架（图 2）建立在定义明确的构造之上。

基于大语言模型的多智能体系统的拓扑结构。多智能体系统表示为有向图 $\mathcal{G} = (\mathcal{V},\ \mathcal{E})$，其中每个节点 $v_i \in \mathcal{V}$ 表示一个基于大语言模型的智能体，每条有向边 $(v_i,\ v_j) \in \mathcal{E}$ 表示从 v_i 到 v_j的通信路径。每个智能体 v_i 由以下属性描述：档案属性 P_i，包括 [角色身份、模型基础、工具访问权限和历史状态]；以及条件属性 C_i，用于捕捉运行时环境条件。我们将条件 C 建模为多个特征的组合，其中每个特征对应一个不同的语义方面 $\mathcal{F}$，例如模型类型、工具可用性或任务复杂度。形式化地，

$$C = \{c_1, c_2, \ldots, c_k\}, \quad c_i \in \mathcal{F}_i, \tag{1}$$

$$\mathcal{C} = \mathcal{F}_1 \times \mathcal{F}_2 \times \cdots \times \mathcal{F}_k, \tag{2}$$

为确保异构特征之间的语义一致性，我们使用统一的预训练语言模型对结构化条件进行编码，将所有特征维度对齐到共享的嵌入空间，使模型能够在推理过程中处理未见过的特征组合。

3.1 通信流水线

给定用户查询 $\mathcal{Q}$，我们的系统会在多智能体拓扑$\mathcal{G} =$（$\mathcal{V}$，$\mathcal{E}$）上执行 K 轮通信，其中 $\mathcal{V} = \{v_1, \ldots, v_N\}$ 是智能体集合，$\mathcal{E}$表示有向通信边。

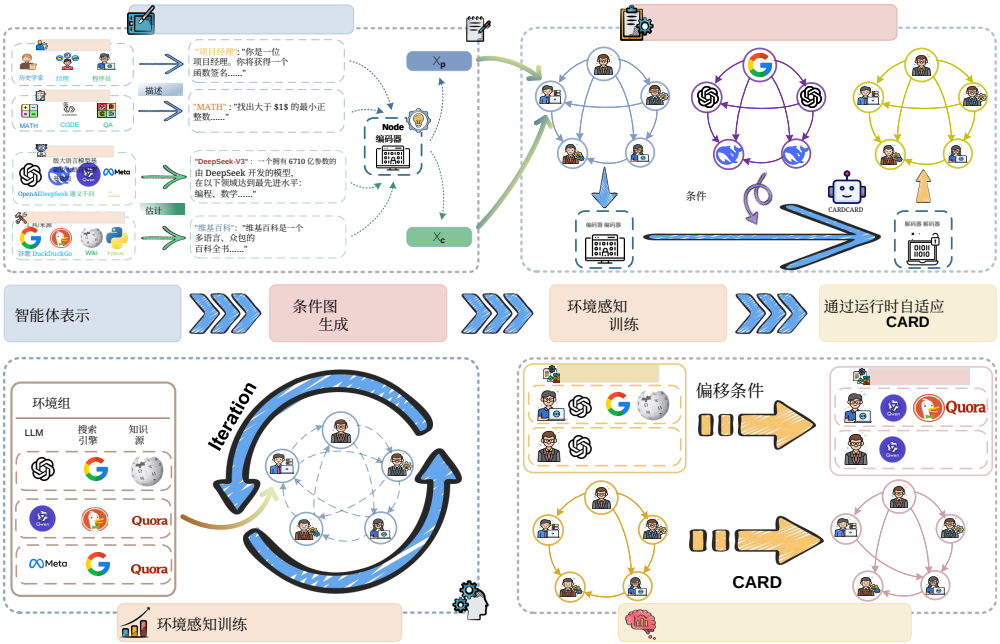


图 2：条件智能体图设计器（CARD）框架概览。智能体画像与动态环境条件被编码为嵌入，条件图生成模块（编码器 → 条件适配 → 解码器）利用这些嵌入生成自适应通信拓扑。随后，CARD 进行环境感知训练，在不断变化的资源配置下迭代优化图，并部署运行时自适应，以根据新的环境状态自动更新多智能体拓扑。

拓扑调度。 为确保信息流有效，拓扑调度函数 φ 会确定智能体的一个排列 $\sigma = \{v_{(1)}, \ldots, v_{(N)}\}$，并遵循无环依赖：

$$\varphi : \mathcal{G} \to \sigma, \quad \text{such that} \quad \forall j > i,\ v_{(j)} \notin \mathcal{N}_{\text{in}}(v_{(i)}), \tag{3}$$

其中 $\mathcal{N}_{\text{in}}(v_{(i)})$ 表示智能体 $v_{(i)}$ 的上游邻居集合。

消息传播。 在每个通信轮次 $t \in \{1, \ldots, K\}$，每个智能体 v_i 接收 (i) 系统级提示词 $\mathcal{P}_{\text{sys}}^{(t)}$、(ii) 用户级提示 $\mathcal{P}_{\text{usr}}^{(t)}$，以及 (iii) 同一轮次中其入边邻居的响应集合：

$$\mathcal{R}_i^{(t)} = v_i\Big(\mathcal{P}_{\text{sys}}^{(t)}, \mathcal{P}_{\text{usr}}^{(t)}, \{\mathcal{R}_j^{(t)} : v_j \in \mathcal{N}_{\text{in}}(v_i)\}\Big), \tag{4}$$

其中 $\mathcal{R}_i^{(t)}$ 是智能体 v_i 在轮次 t 生成的响应。

输出聚合。 经过 K 轮交互后，系统汇总所有智能体的最终轮输出，形成系统最终响应：

$$\alpha^{(K)} = \text{Aggregate}\big(\mathcal{R}_1^{(K)}, \ldots, \mathcal{R}_N^{(K)}\big), \tag{5}$$

其中 Aggregate(·) 表示对终端响应进行特定任务的聚合函数（例如投票、选择或摘要生成）。

3.2 AMACP：自适应多智能体通信协议

为确保拓扑构建具有意义，我们定义 AMACP：

AMACP 定义

给定查询 $\mathcal{Q}$，通信拓扑 $\mathcal{G}$ 必须满足：*1.* 有效性：最大化条件任务效用 $u(\mathcal{G}(\mathcal{Q}|\mathcal{C})$；*2.* 成本效益：最小化条件财务成本 $w(\mathcal{G};\mathcal{C})$；*3.*自适应性：根据环境条件 $\mathcal{C}$ 的变化调整拓扑。这些目标共同编码在以下优化问题中：

$$\min_{\mathcal{G}\in\mathbb{G}} \mathcal{L}_{\text{AMACP}}(\mathcal{G};\mathcal{Q},\mathcal{C}) = -u(\mathcal{G}(\mathcal{Q}\mid\mathcal{C})) + \beta\cdot w(\mathcal{G};\mathcal{C}), \tag{6}$$

where $u(\cdot)$ 表示任务特定效用函数，$w(\cdot)$ 表示条件通信成本，*and* $\beta\in\mathbb{R}^{+}$ 是一个可调节的权衡超参数。

4 条件智能体图设计器

我们提出了条件智能体图设计器（CARD），用于构建能够根据环境进行自适应调整的多智能体拓扑。CARD 包含四个关键阶段：(1) 智能体表示，(2) 条件图生成，(3) 环境感知训练，以及 (4) 运行时自适应。完整工作流程总结于算法 1，并在图 2 中进行了可视化。

智能体表示。 给定查询 Q 和环境配置 C，CARD 首先构建一个初始多智能体网络。每个智能体 $v_i\in V$ 由两个部分描述：首先，配置向量$P_i=[\mathcal{T}_p(\text{基础}_i)$，角色$_i$，$\mathcal{T}_p(\text{插件}_i)]$,用于描述智能体的静态属性，包括其基础模型、指定角色和支持的工具。这里，$\mathcal{T}_p(\cdot)$表示一种自然语言模板函数，用于将类别特征（例如模型名称、角色身份、插件类型）转化为文本嵌入。其次，条件向量 $C_i=\mathcal{T}_c(\mathcal{C}_i)$ 用于描述 v_i的运行时环境状态，例如模型可用性、令牌成本或 API 可靠性。函数 $\mathcal{T}_c(\cdot)$生成用于编码动态系统条件的文本描述。随后，这些表示将被编码为用于条件图生成的节点特征。模板实例化示例见附录 F，涉及 $\mathcal{T}_p$ 和 $\mathcal{T}_c$。

条件图生成 给定用户查询 $\mathcal{Q}$ 和初始环境配置后，CARD$\tilde{}$ $\tilde{}$会在 $N=|\mathcal{V}|$ 个智能体上构建初步的智能体图 $\mathcal{G}=(\mathcal{V},\ \mathcal{E})$。每个智能体 $v_i\in\mathcal{V}$ 都关联一段配置文本 $\mathcal{P}_i$ 和一段条件文本 $\mathcal{C}_i$，分别将其嵌入为 X_i^p 和 X_i^c。沿$\tilde{}$个智能体堆叠后，得到 $X_p=[X_1^p,\ldots,X_N^p]$和 $X_c=[X_1^c,\ldots,X_N^c]$。边集 $\mathcal{E}$根据锚定拓扑 $\mathcal{A}$（例如链式结构或星型结构）进行初始化，为初始连接性提供结构先验。

为获得经过优化且能感知查询和上下文的通信拓扑，CARD 使用编码器–解码器图生成模块。编码器由两个可学习的图编码器 ϕ_p 和 ϕ_c组成，用于生成配置通道和条件通道的潜在表示：

$$\mathbf{H}_p=\phi_p(H_p\mid X_p,\mathcal{A};\Theta_p), \tag{7}$$
$$\mathbf{H}_c=\phi_c(H_c\mid X_c,\mathcal{A};\Theta_c), \tag{8}$$

其中$\mathbf{H}_p=[\mathbf{h}_1^p,\ldots,\mathbf{h}_N^p]$ 和$\mathbf{H}_c=[\mathbf{h}_1^c,\ldots,\mathbf{h}_N^c]$ 表示潜在状态，Θ_p,Θ_c为编码器参数。解码器 ψ_θ 随后以这些潜在状态和一个查询为条件估计成对边概率

emb嵌入$\mathbf{h}_{\mathcal{Q}}$（查询被视为一个辅助节点，对两个通道中的所有智能体进行注意力计算）：

$$\psi(S \mid \mathbf{H}_p, \mathbf{H}_c) = \prod_{i,j} \psi\big(S_{ij} \,\big|\, \mathbf{h}_i^p, \mathbf{h}_i^c, \mathbf{h}_j^p, \mathbf{h}_j^c, \mathbf{h}_{\mathcal{Q}}; \Theta_d\big)\,, \tag{9}$$

其中，$S_{ij} \in [0,1]$ 是预测链接概率，Θ_d 是解码器参数。最后，通过对预测邻接矩阵进行阈值化处理，得到通信拓扑：

$$\mathcal{E}_{\text{com}} = \{(v_i, v_j) \mid S_{ij} > \tau\}, \qquad \mathcal{G}_{\text{com}} = (\mathcal{V}, \mathcal{E}_{\text{com}}), \tag{10}$$

阈值 $\tau \in$（0,1）由用户指定或通过验证集选定。所得 $\mathcal{G}_{\text{com}}$ 作为下游多智能体通信与推理的骨干，并根据静态配置和动态运行时状态进行自适应调节。

环境感知训练。给定查询 Q 和环境条件 $\mathcal{C}$（第 3 节），CARD 通过迭代处理采样得到的（$Q, \mathcal{C}$）对，并在 $\mathcal{G}_{\text{com}}$上运行 $K \in \mathbb{N}$ 轮多智能体交互来进行训练。在通信轮次 $t \in \{1, \ldots, K\}$，智能体 v_i 接收系统级提示 $\mathcal{I}_{\text{sys}}^{(t)}$、用户级提示 $\mathcal{I}_{\text{usr}}^{(t)}$以及上游消息 $\{\mathcal{R}_j^{(t)} \mid v_j \in \mathcal{N}_{\text{in}}(v_i)\}$，并通过公式 4 生成响应 $\mathcal{R}_i^{(t)}$。经过K 轮后，任务特定聚合算子 AGGREGATE(·)（例如投票、选择或摘要生成）将终端响应合并为系统输出 $\alpha^{(K)}$，如公式 5 所示。

令 Θ_p、Θ_c 和 Θ_d 分别表示配置编码器 ϕ_p、条件编码器 ϕ_c 和图解码器 ψ的参数。我们通过对实现 AMACP 目标（方程 6）的 CARD 损失进行梯度下降来优化这些参数：

$$\mathcal{L}_{\text{CARD}}(Q, \mathcal{C}; \Theta_p, \Theta_c, \Theta_d) = \underbrace{-\,u\Big(\alpha^{(K)}\Big)}_{\text{task utility}} + \beta \quad \underbrace{w(\mathcal{G}_{\text{com}}; \mathcal{C})}_{\text{condition-aware cost}}\,, \tag{11}$$

其中，u(·) 是任务特定效用（例如准确率/正确概率），w(·) 衡量条件化通信成本，而 $\beta > 0$ 用于平衡效用与成本。具体而言，为了提升通信效率~并保持性能，我们对解码器输出的软 通信图 $\mathcal{G}_{\text{com}}$ 施加正则化。令$S \in [0,1]^{N \times N}$ 为预测的（有向）链接概率矩阵，并定义 $p_{ij} := S_{ij}$ 为边（$v_i \to v_j$）在条件 $\mathcal{C}$下处于激活状态的概率。令 $\text{Cost}_{ij} \geq 0$ 表示边（i、j）上的预期令牌级推理成本（它是基础模型和交换令牌数量的函数）。条件感知正则项为：

$$\min_{\widetilde{\mathcal{G}}_{\text{com}} \in \mathbb{G}} w\Big(\widetilde{\mathcal{G}}_{\text{com}}, \mathcal{C}\Big) = \sum_{(i,j) \in \widetilde{\mathcal{G}}_{\text{com}}} \text{Cost}_{ij}\, p_{ij}, \tag{12}$$

其中，$\mathbb{G}$ 是 $\mathcal{V}$上可接受的（软）有向图空间。实际执行时使用的 $\mathcal{G}_{\text{com}}$ 通过对 S进行阈值处理获得；训练时则通过 S 反向传播，并依据公式 11 更新（Θ_p，Θ_c，Θ_d）。

通过 CARD 进行运行时适应。部署时，当外部条件发生变化（例如基础模型能力、工具可靠性或成本），CARD 无需重新训练即可更新通信拓扑，方法是根据刷新后的条件信号解码出新边：

$$\mathcal{G}_{\text{com}}^{\text{new}} = \psi\big(\,\phi_p(X_p),\, \phi_c(X_c^{\text{new}}),\, \mathcal{A}\big), \tag{13}$$

其中，X_p 编码智能体配置文件静态（角色、基础模型和工具），X_c^{new}编码更新后的运行时条件，ϕ_p，ϕ_c 将这些信息映射为潜在节点状态，$\mathcal{A}$ 是锚点先验（例如链式结构、星型结构或全连接结构），而ψ 则解码边概率（以 τ为阈值），从而生成修订后的邻接矩阵。这种单次重新计算方式能在实时变化下维持稳健且高效的协作。

5 实验

数据集与指标。我们在三个标准基准测试上评估 CARD：编程代码生成（HumanEval）（Chen 等人，2021）、数学推理（MATH）（Hendrycks 等人，2021b），以及通用推理和语言理解（MMLU）（Hendrycks 等人，2021a）。

基线与设置。我们将我们的方法与三类方法进行比较：**原生大语言模型**，利用模型的原生能力直接生成答案；**手工设计的智能体**，包括在单智能体设置下的思维链（思维链）（Wei 等人，2022年），以及在多智能体配置下的大语言模型辩论（Du 等人，2024）和随机图；以及**自动优化拓扑**，包括*GPT-Swarm*（Zhuge 等人，2024）和*G-Designer*（Zhang 等人，2025a）（二者与我们采用相同的图形式化方法）的图学习技术，以及启发式规则优化器*Aflow*（Zhang 等人，2025b）。我们评估了来自不同提供商的多种语言模型，这些模型分别代表不同的技术范式、训练方法和架构设计。更多详情请参见附录 C 和 G。

5.1 主要结果

表 1：HumanEval、MATH 和 MMLU 上多智能体与拓扑设计方法的评估。“Mul.”、“Auto.”和“Cond.”分别表示是否支持多智能体协作、自动化拓扑设计和条件配置，其中 ✓ 表示“是”，而 ✗ 表示“否”。对于自动化方法，青色单元格表示域内适配（在同一大语言模型上训练和测试），绿色单元格表示域外适配（泛化至未见过的大语言模型），而黄色 单元格表示平均性能。

方法	Mul.	自动化	条件	gpt4o-mini	deepseek-v3	Llama3-70B	GPT-4o	qwen-72B	Avg.
HumanEval									
Vanilla	✗	✗	✗	85.83	92.50	76.66	85.83	86.66	85.50
CoT	✗	✗	✗	88.33	93.33	78.33	90.00	88.33	87.66
随机图	✓	✗	✗	87.50	94.16	73.33	90.00	85.00	86.00
大语言模型辩论	✓	✗	✗	91.66	95.00	75.83	87.50	80.00	86.00
GPTswarm	✓	✓	✗	89.16	92.50	78.33	91.66	81.66	86.66
Aflow	✓	✓	✗	90.83	92.50	**85.83**	93.33	86.66	89.83
G-Designer	✓	✓	✗	89.16	94.16	75.00	88.33	85.83	86.50
CARD	✓	✓	✓	**93.33**	**95.83**	81.66	**93.33**	**88.33**	**90.50**
MATH									
Vanilla	✗	✗	✗	59.16	74.16	41.66	70.00	71.66	63.33
CoT	✗	✗	✗	61.66	80.00	41.66	65.00	73.33	64.33
随机图	✓	✗	✗	60.00	88.33	38.33	70.00	66.67	64.67
大语言模型辩论	✓	✗	✗	59.16	85.00	46.66	71.66	71.66	66.83
GPTswarm	✓	✓	✗	67.50	90.83	46.66	67.50	78.33	70.16
Aflow	✓	✓	✗	**80.83**	91.66	40.00	75.83	80.83	73.83
G-Designer	✓	✓	✗	70.00	91.66	47.50	75.00	79.16	72.66
CARD	✓	✓	✓	73.33	**91.66**	**48.33**	**76.67**	**82.50**	**74.50**
MMLU									
Vanilla	✗	✗	✗	77.12	86.27	75.16	86.93	79.74	81.04
CoT	✗	✗	✗	81.05	92.16	75.16	88.89	83.66	84.18
随机图	✓	✗	✗	79.74	90.85	75.82	86.93	83.00	83.27
大语言模型辩论	✓	✗	✗	80.39	90.20	76.47	88.24	84.31	83.92
GPTswarm	✓	✓	✗	82.35	89.54	77.78	86.27	84.31	84.05
Aflow	✓	✓	✗	79.74	84.97	77.12	89.54	83.00	82.87
G-designer	✓	✓	✗	83.00	90.85	77.12	86.93	84.31	84.44
CARD	✓	✓	✓	**84.97**	**93.46**	**80.39**	**89.54**	**84.97**	**86.67**

条件设计（CARD）始终带来最佳的整体性能。在 HumanEval 上达到 90.50%、MATH 上达到 74.50%、MMLU 上达到 86.67%，值得注意的是，CARD 在 15 个模型–基准组合中的 13 个组合中取得或并列取得最高分，展现出跨不同大语言模型基座的强大稳健性。

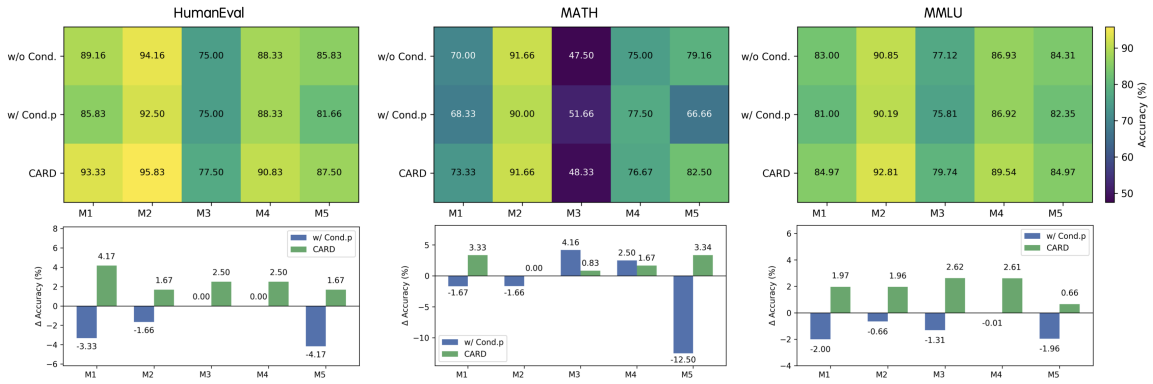


图3：无条件、提示级条件与 CARD 在 HumanEval、MATH 和 MMLU 上跨大语言模型基座（M1–M5，与表1相同）的性能与增益。**上：**绝对准确率（%）。**下：** Δ 相对于无条件基线的准确率（%）。

随着设计抽象程度的提升，性能收益逐步累积。单智能体方法（Vanilla、思维链）确立了具有竞争力的基线，但缺乏协作。固定的多智能体拓扑（随机图、大语言模型辩论）通过实现并行推理带来了适度提升（+0.5–2.0 个百分点）。自动拓扑学习器（GPT-Swarm、Aflow、G-Designer）进一步通过优化静态通信结构提升性能（+1.0–4.0 个百分点）。最后，CARD 的条件自适应通过根据环境信号调整拓扑，相较于这些静态设计额外带来 +0.5–3.0 个百分点的优势。

条件自适应在域外设置下尤其有效。通过显式地以模型与工具状态为条件，CARD 缩小了域内与域外评估之间的差距。例如，在 MATH 上，当从 deepseek-v3 切换到 qwen-72B 时，G-Designer 的准确率从 91.66% 降至 79.16%，而 CARD 的降幅更小（从 91.66% 降至 82.50%），凸显了其对未见设置的适应能力。

5.2 如何在大语言模型拓扑生成中嵌入条件?

图3报告了一项消融研究，比较将环境条件注入多智能体拓扑生成的不同方式。我们比较了无条件基线（无条件）、一种将条件描述符附加到系统提示的简单提示级注入（提示级条件），以及我们的 CARD 方法，该方法将条件直接嵌入图生成模块。每种变体均在五大大语言模型基座上于 HumanEval、MATH 和 MMLU 中进行评估，报告绝对精度以及相对于无条件基线的 Δ 精度。

CARD 在所有基准测试上均带来稳健的非负增益。 而简单的提示条件化可能适得其反，在基座 M5 上使 MATH 最多下降 12.50%，在基座 M1 上使 MMLU 下降 2.00%。CARD 在每一个模型–基准测试对上都持续带来正向提升（例如，MATH 上为 +0.83% 至 +3.34%，MMLU 上为 +0.66% 至 +2.62%，HumanEval 上为 +2.50% 至 +23.33%），证明结构化拓扑自适应远比仅提示方法更为可靠。

CARD 弥补了较弱基线模型的不足。提升幅度最大的是最具挑战性的设置，例如使用 M3 时在 MMLU 上提升 +2.62 %，以及使用 M5 时在 MATH 上提升 +3.34 %，这表明 CARD 能够缩小域外性能差距，并减轻能力较弱的

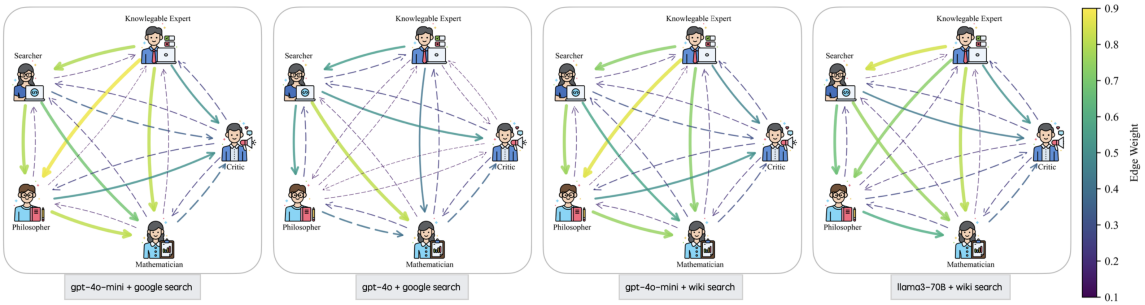


图 4：不同条件下 CARD 拓扑矩阵的可视化（矩阵及相关性分析详情见附录 D）。边的粗细表示智能体之间的通信概率。从左至右展示配置 1 至 4（表 10）。

大语言模型基座的局限性。关于隔离来源/工具效应和局部条件扰动的进一步分析见附录 B.2。

5.3 环境条件如何塑造最终通信架构？

图 4 展示了四种具有代表性的实验配置，将两种大语言模型能力（GPT-4o-mini 与 GPT-4o 或 Llama3-70B）和两种搜索引擎（Google 搜索与 Wiki Search）组合起来，以说明 CARD 如何实现条件自适应。通过可视化由此得到的拓扑矩阵，我们旨在量化模型能力和检索质量的变化如何影响边密度、方向流动模式及整体图结构。本案例研究验证了 CARD 能够根据动态环境信号调整多智能体通信图，并揭示了该协议在现实运行条件变化下的实际表现。

模型能力越弱，越需要密集协作。在配置 1（GPT-4o-mini + Google 搜索）中，平均边权重显著高于配置 2（GPT-4o + Google 搜索），这表明较小的模型会通过加强多智能体通信来弥补其固有能力较低的不足。

更换搜索引擎保留了全局结构，但改变了局部流向。将 GPT-4o-mini 的 Google 替换为维基（配置 1 与配置 3）后，皮尔逊相关为 $r = 0.9797$ 和 $p = 0.0006$（附录 D.2），证实整体拓扑几乎完全相同。然而在局部，知识专家→搜索者之间的边权重明显下降，而搜索者→数学家之间的连接降幅大于搜索者→哲学家，这反映了特定领域检索效能的差异。

最低能力 + 搜索质量较低时，对外部知识的依赖最强。Llama3-70B + 维基配置（配置 4）形成了最密集的图，平均边权重也最高，表明当模型能力和搜索质量都降低时，对外部信息的依赖达到峰值（配置 1 与配置 4：$r = 0.7789$、$p = 0.0679$（附录 D.2））。

附录 B.1 提供了按模型能力和规模划分的更多定量分析，附录 B.2 则提供了按工具/知识来源划分的分析。附录 B.3 详细说明了智能体数量变化时的可扩展性，附录 B.4 总结了面对定向攻击时的鲁棒性以及准确率与成本权衡。

6 结论

我们提出了 AMACP，这是一种自适应多智能体通信协议；同时提出了 CARD，这是一种条件图生成框架，可根据动态环境调整基于大语言模型的智能体拓扑。在 HumanEval、MATH 和 MMLU 上进行的实验表明，在模拟变化（模型升级、工具变更和数据扰动）下，CARD 的准确率比静态设计和基于提示的设计最高提升了三个百分点，同时仍能保持较高的成本效益。拓扑可视化凸显了 CARD 根据智能体能力和资源质量调整通信模式的能力。未来的工作将扩展至更大规模的智能体集群，整合在线强化学习以实现持续适应，并在真实世界的多智能体应用中验证 CARD。有关局限性及未来研究方向的详细讨论，请参阅附录 A。

7 伦理声明

本研究仅使用公开可用的数据集和模型，不涉及人类受试者或私有数据。我们认识到自主多智能体大语言模型系统可能带来的广泛社会风险，并鼓励在采取适当安全保障措施的前提下负责任地部署。

8 可复现性声明

为支持可复现性，我们在 https://anonymous.4open.science/r/agentgraph-FF9A 提供了完整的源代码、训练与评估脚本以及提示模板。所有实验均基于公开可用的基准测试（HumanEval、MATH、MMLU），并使用开源或可通过 API 访问的大语言模型；附录 C–F 中记录了完整的实现细节、模型配置和超参数。结果取多次运行的平均值，所有指标和可视化均由脚本生成，便于核验。

参考文献

Yuanchen Bei、Weizhi Zhang、Siwen Wang、Weizhi Chen、Sheng Zhou、Hao Chen、Yong Li、Jiajun Bu、Shirui Pan、Yizhou Yu、Irwin King、Fakhri Karray 和 Philip S. Yu。《图遇见AI智能体：分类、进展与未来机遇》。*CoRR*，abs/2506.18019，2025。网址：https://doi.org/10. 48550/arXiv.2506.18019。

Maciej Besta、Nils Blach、Ales Kubicek、Robert Gerstenberger、Michal Podstawski、Lukas Gianinazzi、Joanna Gajda、Tomasz Lehmann、Hubert Niewiadomski、Piotr Nyczyk 和 Torsten Hoefler。《思维图：使用大语言模型解决复杂问题》。收录于*AAAI* 会议论文集，第17682–17690页，2024。网址：https://doi.org/10.1609/aaai.v38i16.29720。

Xiaofu Chang、Xuqin Liu、Jianfeng Wen、Shuang Li、Yanming Fang、Le Song 和 Yuan Qi。基于神经交互过程的连续时间动态图学习。载于 *CIKM* 论文集，第 145–154 页，2020 年。URL https://doi.org/10.1145/3340531.3411946。

Bei Chen、Fengji Zhang、Anh Nguyen、Daoguang Zan、Zeqi Lin、Jian-Guang Lou 和 Weizhu Chen。Codet：通过生成测试进行代码生成。载于 *ICLR* 论文集，2023 年。URL https://openreview. net/forum?id=ktrw68Cmu9c。

Huajun Chen. 大型知识模型：视角与挑战。*Data Intell.*，6(3):587–620，2024。网址：https://doi.org/10.3724/2096-7004.di.2024.0001。

Mark Chen、Jerry Tworek、Heewoo Jun、Qiming Yuan、Henrique Ponde de Oliveira Pinto、Jared Kaplan、Harri Edwards、Yuri Burda、Nicholas Joseph、Brockman 等。评估基于代码训练的大语言模型。*CoRR*，abs/2107.03374，2021。网址：https://arxiv.org/abs/2107.03374。

DeepSeek-AI、Aixin Liu、Bei Feng、Bing Xue、Bingxuan Wang、Bochao Wu、Chengda Lu、Chenggang Zhao、Chengqi Deng、Chenyu Zhang、Chong Ruan、Damai Dai、Daya Guo、Dejian Yang、Deli Chen 等。Deepseek-v3 技术报告。*CoRR*，abs/2412.19437，2024。网址：https://doi.org/10.48550/ arXiv.2412.19437。

DeepSeek-AI、Daya Guo、Dejian Yang、Haowei Zhang、Junxiao Song、Ruoyu Zhang、Runxin Xu、Qihao Zhu、Shirong Ma、Peiyi Wang、Xiao Bi 等。Deepseek-r1：通过强化学习激励大语言模型的推理能力。*CoRR*，abs/2501.12948，2025。网址：https://doi.org/10.48550/arXiv. 2501.12948。

Yilun Du、Shuang Li、Antonio Torralba、Joshua B. Tenenbaum 和 Igor Mordatch。通过多智能体辩论提升语言模型的事实性和推理能力。载于*ICML* 论文集，第《机器学习研究论文集》第 235 卷，第 11733–11763 页，2024。网址：https://proceedings. mlr.press/v235/du24e.html。

Abhimanyu Dubey、Abhinav Jauhri、Abhinav Pandey、Abhishek Kadian、Ahmad Al-Dahle、AieshaLetman、Akhil Mathur、Alan Schelten、Amy Yang、Angela Fan、Anirudh Goyal 等人。Llama 3 模型群。*CoRR*，abs/2407.21783，2024。网址：https://doi.org/10.48550/arXiv.2407. 21783。

Dan Hendrycks、Collin Burns、Steven Basart、Andy Zou、Mantas Mazeika、Dawn Song 和 Jacob Steinhardt。衡量大规模多任务语言理解能力。载于*ICLR* 论文集。OpenReview.net，2021a。网址：https://openreview.net/forum?id=d7KBjmI3GmQ。

Dan Hendrycks、Collin Burns、Saurav Kadavath、Akul Arora、Steven Basart、Eric Tang、Dawn Song 和 Jacob Steinhardt。《使用 MATH 数据集衡量数学问题求解能力》。载于*NeurIPS* 数据集与基准论文集，2021b。网址：https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html。

Sirui Hong、Mingchen Zhuge、Jonathan Chen、Xiawu Zheng、Yuheng Cheng、Jinlin Wang、Ceyao Zhang、Zili Wang、Steven Ka Shing Yau、Zijuan Lin、Liyang Zhou、Chenyu Ran、Lingfeng Xiao、Chenglin Wu 和 Jürgen Schmidhuber。《MetaGPT：用于多智能体协作框架的元编程》。载于*ICLR* 论文集，2024。网址：https://openreview.net/forum?id=VtmBAGCN7o。

Nan Hu、Yike Wu、Jiaye Li、Huikang Hu、Guilin Qi、Songlin Zhai、Yingrui Chen、Tianx- ing Wu、Tongtong Wu、Jiaoyan Chen 等。Taxreasoning：基准测试——在不断演变的税法背景下进行知识密集型数学推理。载于*AAAI* 论文集，2025。链接：https://research.manchester.ac.uk/en/publications/taxreasoning-benchmarking-knowledge-intensive-mathematical-reason/。Sungwoo Lee、Younghyun Oh、Hyunhoe An、Hyebhin Yoon、Karl J. Friston、Seok Jun Hong 和 Choong- Wan Woo。受生命启发的内感受人工智能：面向自主自适应智能体。*CoRR*，abs/2309.05999，2023。链接：https://doi.org/10.48550/arXiv.2309.05999。Yanming Li、Xuelin Zhang、WenJie Lu、Ziye Tang、Maodong Wu、Haotian Luo、Tongtong Wu、Zijie Peng、Hongze Mi、Yibo Feng、Naiqiang Tan、Chao Huang、Hong Chen 和 Li Shen。谁应获得奖励？SHARP：面向多智能体系统的基于夏普利信用分配的优化，2026。链接：https: //arxiv.org/abs/2602.08335。

Tian Liang、Zhiwei He、Wenxiang Jiao、Xing Wang、Yan Wang、Rui Wang、Yujiu Yang、Shuming Shi 和 Zhaopeng Tu。通过多智能体辩论鼓励大语言模型进行发散性思维。载于*EMNLP* 论文集，第17889–17904页，2024。链接：https://doi.org/10.18653/v1/2024. emnlp-main.992。

Ruibo Liu、Jason Wei、Shixiang Shane Gu、Te-Yen Wu、Soroush Vosoughi、Claire Cui、Denny Zhou 和 Andrew M. Dai。心智之眼：通过模拟实现基于接地的语言模型推理。载于 *ICLR* 会议论文集，2023。URL https://openreview.net/forum?id=4rXMRuoJlai。

Yixin Liu、Guibin Zhang、Kun Wang、Shiyuan Li 和 Shirui Pan。图增强大语言模型智能体：当前进展与未来展望。*CoRR*，abs/2507.21407，2025。URL https: //doi.org/10.48550/arXiv.2507.21407。

Dehai Min、Kailin Zhang、Tongtong Wu 和 Lu Cheng。Quco-rag：面向动态检索增强生成，从预训练语料库量化不确定性。*CoRR*，abs/2512.19134，2025。URL https://doi.org/10.48550/arXiv.2512.19134。

OpenAI. 你好 GPT-4o，2024a。网址 https://openai.com/index/hello-gpt-4o/。

OpenAI. GPT-4o mini：推进成本高效的智能，2024b。网址 https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/。

Aldo Pareja、Giacomo Domeniconi、Jie Chen、Tengfei Ma、Toyotaro Suzumura、Hiroki Kanezashi、Tim Kaler、Tao B. Schardl和Charles E. Leiserson。EvolveGCN：面向动态图演化图卷积网络。载于*AAAI*会议论文集，第5363–5370页，2020。网址 https://doi.org/10.1609/aaai. v34i04.5984。

Yu Wang、Chi Han、Tongtong Wu、Xiaoxin He、Wangchunshu Zhou、Nafis Sadeq、Xiusi Chen、Zexue He、Wei Wang、Gholamreza Haffari、Heng Ji和Julian J. McAuley。迈向全生命周期认知系统。机器学习研究汇刊，2025，2025。网址 https://openreview.net/forum?id=LZ9FmeFeLV。

Jason Wei、Xuezhi Wang、Dale Schuurmans、Maarten Bosma、Brian Ichter、Fei Xia、Ed H. Chi、Quoc V. Le 和 Denny Zhou。《思维链提示能够激发大语言模型的推理能力》。载于*NeurIPS* 论文集，2022年。网址：http://papers.nips.cc/paper_files/paper/2022/hash/ 9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html。

Qingyun Wu、Gagan Bansal、Jieyu Zhang、Yiran Wu、Shaokun Zhang、Erkang Zhu、Beibin Li、Li Jiang、Xiaoyun Zhang 和 Chi Wang。《AutoGen：通过多智能体对话框架实现新一代大语言模型应用》。*CoRR*，abs/2308.08155，2023。网址：https://doi.org/10.48550/arXiv.2308. 08155。

Tongtong Wu、Linhao Luo、Yuan-Fang Li、Shirui Pan、Thuy-Trang Vu 和 Gholamreza Haffari。《大语言模型的持续学习：综述》。*CoRR*，abs/2402.01364，2024a。网址：https://doi.org/10.48550/arXiv.2402.01364。Tongtong Wu、Weigang Wu、Xingyu Wang、Kang Xu、Suyu Ma、Bo Jiang、Ping Yang、Zhenchang Xing、Yuan-Fang Li 和 Gholamreza Haffari。《Versicode：迈向版本可控代码生成》。*CoRR*，abs/2406.07411，2024b。网址：https://doi.org/10.48550/arXiv.2406.07411。Tongtong Wu、Rongyi Chen、Wenjie Du、Suyu Ma、Guilin Qi、Zhenchang Xing、Shahram Khadivi、Ramesh Periyathambi 和 Gholamreza Haffari。《环境感知代码生成：我们距离目标还有多远？》*CORR*，abs/2601.12262，2026。网址：https://doi.org/10.48550/arXiv.2601.12262。

An Yang、Baosong Yang、Binyuan Hui、BoZheng、Bowen Yu、Chang Zhou、Chengpeng Li、Chengyuan Li、Dayiheng Liu、Fei Huang、等。《通义千问2技术报告》。*CoRR*，abs/2407.10671，2024。网址：https://doi.org/10.48550/arXiv.2407.10671。

Shunyu Yao、Dian Yu、Jeffrey Zhao、Izhak Shafran、Tom Griffiths、YuanCao 和Karthik Narasimhan。思维树：使用大语言模型进行审慎解题。收录于*NeurIPS* 论文集，2023。网址：http://papers.nips.cc/paper_files/paper/2023/hash/ 271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html。

Guibin Zhang、Yanwei Yue、Xiangguo Sun、Guancheng Wan、Miao Yu、Junfeng Fang、Kun Wang、Tianlong Chen 和 Dawei Cheng。G-designer：通过图神经网络构建多智能体通信拓扑。收录于*ICML* 论文集，2025a。网址：https://openreview.net/forum?id=LpE54NUnmO。

Jiayi Zhang、Jinyu Xiang、Zhaoyang Yu、Fengwei Teng、Xiong-Hui Chen、Jiaqi Chen、Mingchen Zhuge、Xin Cheng、Sirui Hong、Jinlin Wang、Bingnan Zheng、Bang Liu、Yuyu Luo 和 Chenglin Wu。Aflow：自动化智能体工作流生成。收录于*ICLR* 论文集，2025b。网址：https://openreview. net/forum?id=z5uVAKwmjf。

Jintian Zhang、Xin Xu、Ningyu Zhang、Ruibo Liu、Bryan Hooi 和 Shumin Deng。《探索大语言模型智能体的协作机制：社会心理学视角》。载于*ACL* 论文集，第 14544–14607 页，2024a。网址：https://doi.org/10.18653/v1/2024.acl-long.782。

Xilin Zhang、Zhixin Mao、Ziwen Chen 和 Shen Gao。面向数据分析的高效工具增强型多智能体框架。*Data Intell.*，6(4):923–945，2024b。网址：https://doi.org/10.3724/2096-7004. di.2024.0013。

Mingchen Zhuge、Wenyi Wang、Louis Kirsch、Francesco Faccio、Dmitrii Khizbullin 和 Jürgen Schmidhuber。《GPTswarm：作为可优化图的语言智能体》。载于 *ICML* 会议论文集，第 235 卷，第 62743–62767 页，2024 年。网址：https://proceedings.mlr.press/v235/zhuge24a.html。

附录目录

A 局限性 **15**

B 分析与讨论 15

B.1 条件的定量分析：模型规模与推理能力.15 B.2 条件的定量分析：工具与知识资源.16 B.3 多智能体可扩展性分析.16 B.4 多智能体鲁棒性与成本效益分析.17

C 实现细节 18 D 源数据 19

D.1 可视化邻接矩阵.19 D.2 邻接矩阵相关性分析.20

E 算法工作流 20 F 提示词 21 G 实验配置集 26

A 局限性

本研究聚焦于多智能体通信拓扑的条件优化与适应。然而，本研究并未明确根据环境变化更新智能体级配置，例如各智能体的提示词或内部档案。实践中，同时优化通信拓扑与智能体行为（包括提示词增强和工具选择策略），或许能进一步提升系统性能。探索这一方向仍是未来研究的一个开放课题。此外，尽管我们的形式化方法将多智能体系统表示为图，提供了一种易于认知理解和分析的抽象方式，但基于图的表示可能不足以捕捉复杂现实场景中的领域特定细微差异。例如，在软件工程工作流中，流程约束、工具依赖关系和执行语义往往至关重要。未来的研究可以融入人在回路中的专业知识，例如软件开发最佳实践和调试启发式方法，并探索结合符号先验与已学习的智能体适应机制的混合模型。

B 分析与讨论

B.1 条件的定量分析：模型规模与推理能力

我们进一步开展实验分析，以探究模型能力、模型规模、外部工具和知识来源的变化如何影响多智能体拓扑设计及整体性能。

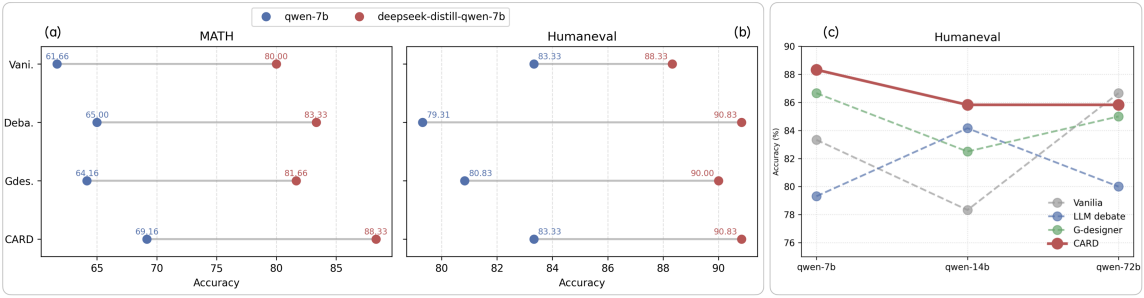


图 5：**L　左**：不同推理能力的大语言模型在 MATH 和 HumanEval 上的准确率力。**右在同一大语言模型系列中**，不同模型规模的准确率。

更强大、规模更大的基础模型能够提升多智能体性能，而 CARD 会进一步放大这些优势；但在简单任务上，单个强大的大语言模型可能因通信开销较低而胜过多智能体协作。在 MATH 和 HumanEval 基准测试上，将 qwen-7b 升级为能力更强的 deepseek-distill-qwen-7b 后，所有方法的准确率均有所提升，其中 CARD 的绝对增益最大，表现也最佳（例如，MATH 准确率从 69.16 % 提升至 88.33 %（+19.17 个百分点））。然而，在使用能力较强的模型处理简单基准测试时（图 5 右），由于存在冗余通信，Vanilla 单智能体的表现略胜多智能体。这表明，只有当任务复杂度足以抵消协同成本时，多智能体才能发挥优势。

CARD 对外部工具和知识资源变化表现出卓越的稳健性 证据：在 Google 搜索、DuckDuckGo 和维基百科之间切换外部工具时，CARD 在 HumanEval 上的准确率仅从 85.62 % 降至 83.00 %（图 6（左）），降幅小于其他方法。同样，在不同知识来源（维基百科、Tutorialspoint、Quora）中，CARD 在数据最丰富的来源上取得了最高性能（85.62 %），即使在信息量较少的来源上，也比其他方法高出 1–2 个百分点

这种韧性凸显了 CARD 在不同资源条件下维持强大多智能体拓扑的能力。

B.2 条件的定量分析：工具与知识资源

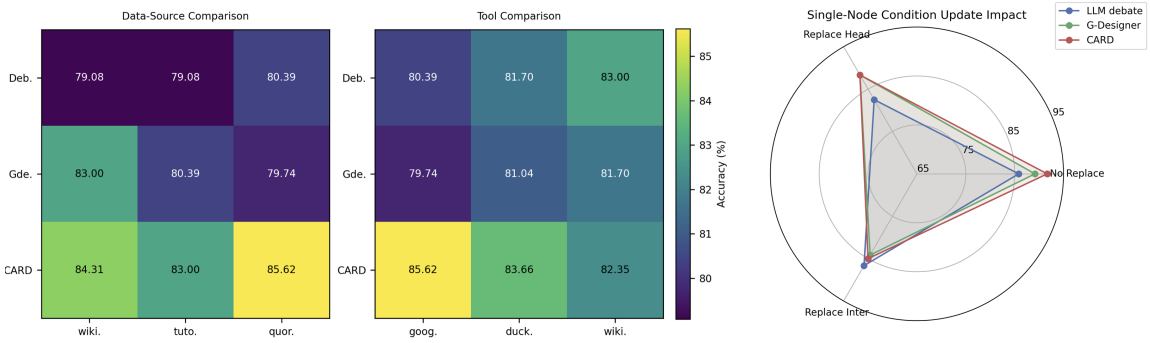


图 6：**左**：使用不同外部知识来源时的性能比较。**中**：使用各种可用工具（搜索引擎）时的性能。**右**：仅更新单个节点的条件而非全局智能体条件时，对多智能体在 HumanEval 上性能的影响。

我们评估外部条件，即知识来源和检索工具，如何影响 CARD 框架的性能。实验使用 HumanEval，并采用与正文相同的基础大语言模型和智能体角色设计。唯一变化的因素是条件特征：在三个知识语料库（维基百科、Tutorialspoint、Quora）之间切换、更换搜索工具（谷歌、DuckDuckGo、WikiSearch），以及通过修改单个节点的条件嵌入进行局部扰动。所有配置均采用相同的检索预算、提示词格式和随机种子。

在知识来源和工具变化的情况下，CARD 始终优于基线方法，并且在条件较弱时性能下降幅度更小。准确率从使用维基百科时的 85.6% 降至使用 Quora 时的 83.0%，但 CARD 仍明显优于大语言模型辩论和 G-Designer。这种韧性源于条件感知图生成，它会根据上游内容质量调整边密度和智能体协调。结果表明，面对领域偏移或嘈杂的外部知识时，拓扑级适应比仅提示条件化具有更强的鲁棒性。

在局部扰动测试中，仅更改一个节点（根节点或中间节点）的条件，性能仍能分别保持在 88.3% 和 85.0%，显著优于基线方法。这表明，CARD 无需对整个图进行重新训练，即可实现低成本的局部重新配置。模块化设计支持在生产环境中进行实用的在线适应，使系统能够针对工具变动或热修复进行轻量级更新，同时在动态的现实约束下保持准确率稳定。

B.3 多智能体可扩展性分析

我们通过将基础大语言模型分为域内（gpt4o-mini、deepseek-v3、llama3）和域外（GPT-4o、qwen-72B）两种设置，评估 CARD 相较于 G-Designer 的可扩展性和鲁棒性。

随着智能体数量增加，CARD 的扩展效果优于 G-Designer，在域外设置下尤其明显。如图 7 所示，随着智能体数量增加，CARD 的 MMLU 得分始终更高，在域外设置下的提升尤为显著（10 个智能体时比 G-Designer 高出最多 +1.99 个百分点）。在域内情况下，两种方法的表现都优于 Vanilla，但 CARD 的上升趋势更为陡峭

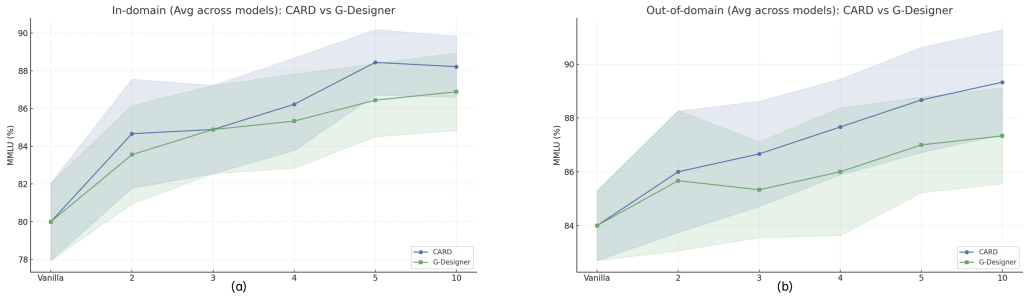


图 7：不同智能体数量下 CARD 与 G-Designer 在 MMLU 上的表现。CARD 在 (a) 域内设置和 (b) 域外设置中始终优于 G-Designer，且在领域偏移下优势更大；阴影区域表示 95% 置信区间。

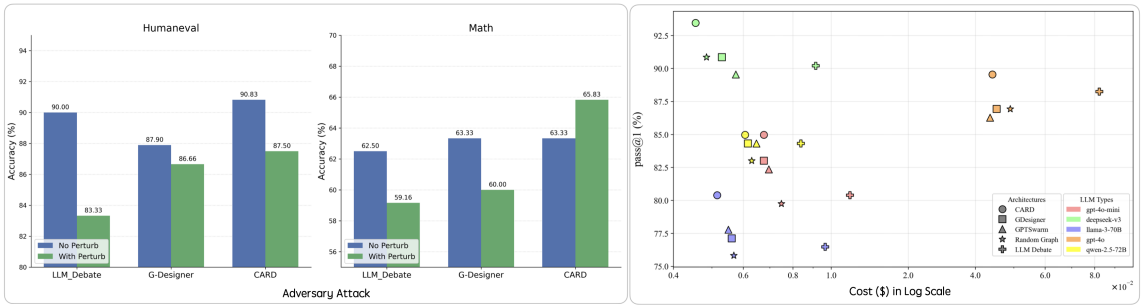


图 8：**L** **eft**：不同方法在对抗攻击前后的准确率比较。**Righ** **t**: 推理 成本（每个实例的美元成本），按方法和大语言模型划分。

，且在智能体数量为 5–10 个时优势进一步扩大。这表明，CARD 中的条件拓扑生成有助于智能体在系统规模扩大时更有效地协调。

CARD 在不确定性相近的情况下，也展现出更强的领域偏移鲁棒性。该方法在分布偏移下也表现出更强的鲁棒性。虽然 G-Designer 在域外场景中的性能提升趋于停滞，CARD 却能持续受益于智能体扩展。此外，CARD 在实现这些提升的同时，置信区间宽度相近或略低，表明其协调效果的提升更可靠、泛化能力更强。这些结果符合 CARD 的设计目标——根据外部约束（如模型类型、成本、能力）生成拓扑，从而即使在未见过的环境中，也能实现自适应且可扩展的多智能体协作。

B.4 多智能体鲁棒性与成本效益分析

我们通过在 HumanEval 基准测试中模拟对中间智能体的定向攻击和配置故障，评估准确率方面的韧性；并计算不同大语言模型基座的训练与评估总开销，以量化经济权衡，从而系统比较静态、学习得到以及条件自适应的通信拓扑在不利条件和预算约束下的表现。

条件自适应显著提升了遭受攻击时的鲁棒性。如图 8（左）所示，当智能体节点遭到攻击时，依赖固定成对提示、缺乏

结构自适应能力的大语言模型辩论，性能下降最为明显（HumanEval 上下降 6.67 个百分点，MATH 上下降 3.34 个百分点）。在攻击条件下训练的 G-Designer 会过滤掉故障节点，性能下降较小（HumanEval 上下降 1.24 个百分点，MATH 上下降 3.33 个百分点），但节点恢复后便失去泛化能力。相比之下，在受攻击和干净条件下均经过训练的 CARD，不仅在遭受攻击时优于 G-Designer（HumanEval 上为 87.50% 对 86.66%；MATH 上为 65.83% 对 60.00%），而且在受损节点恢复后表现出显著更强的性能恢复能力（HumanEval 上为 +3.33 个百分点对 +1.24 个百分点；在 MATH 上，受攻击时的准确率甚至高于未受攻击时的准确率，这表明我们的拓扑能够充分适应受攻击和未受攻击两种条件），凸显了其在性能受损和恢复环境中的卓越韧性与适应能力。

局部条件更新以极低的适应成本恢复了大部分损失的性能。图 6（右）表明，仅替换单个节点的条件而非重新生成整个通信图时，CARD 在头节点扰动下仍保持 88.33% 的准确率，在中间节点扰动下保持 85.00% 的准确率，整体优于大语言模型辩论和 G-Designer，凸显细粒度适应能够以远低于全局重新配置的计算开销保持稳健性。

条件设计在所有方法中实现了最佳的准确率与成本平衡。在图 8（右）中，CARD 的配置（例如，以 4×10^{-3} 美元的评估成本达到 94% 的准确率）位于成本–性能平面的左上区域，而大语言模型辩论等静态多智能体方案以及 G-Designer 等学习型拓扑通常需要更高成本才能达到更低的准确率；这证实了经条件适应的图不仅能提升韧性，还能在给定性能水平下将经济支出降至最低。

C 实现细节

具体而言，我们纳入了 OpenAI 的 GPT-4oOpenAI (2024a) 和 GPT-4o-minigpt-4o-mini OpenAI (2024b)、DeepSeek 的 DeepSeek-V3DeepSeek-AI et al. (2024)（630B）、Meta 的 Llama3-70BDubey et al. (2024)，以及 Alibaba 的 Qwen 系列，其中包括 Qwen-2.5 模型（7B、14B 和 72B）Yang et al. (2024)，还有基于 Qwen-2.5-7B 蒸馏得到的变体 qwen-distill-r1-7bDeepSeek-AI et al. (2025)。这些模型在提供商（体现技术路线和架构选择上的差异）、模型规模（与计算能力和性能相关）以及领域专长（影响知识范围和推理行为）方面各不相同。此外，我们通过改变底层数据源和搜索引擎，考察外部工具对多智能体架构的影响。考察的搜索引擎包括 Google 搜索、DuckDuckGo 和 Wiki Search；数据源包括 Quora、维基百科和 Tutorialspoint。这些配置体现了检索质量和知识覆盖度的多样性。本设置旨在模拟现实世界多智能体系统中资源的动态演变，从而增强所学习的多智能体拓扑在条件变化下的稳健性和适应性。

D 源数据

D.1 可视化邻接矩阵

表 2：邻接矩阵 1：gpt-4o-mini + 谷歌

	知识渊博的专家	搜索者	哲学家	数学家	评论家
知识渊博的专家	已屏蔽	0.79	0.88	0.81	0.5
搜索者	0.17	已屏蔽	0.79	0.69	0.34
哲学家	0.11	0.17	已屏蔽	0.81	0.5
数学家	0.15	0.21	0.15	已屏蔽	0.37
评论家	0.25	0.22	0.25	0.23	已屏蔽

表 3：邻接矩阵 2：GPT-4o + 谷歌

	知识渊博的专家	搜索者	哲学家	数学家	评论家
知识渊博的专家	已屏蔽	0.54	0.15	0.4	0.14
搜索者	0.25	已屏蔽	0.53	0.81	0.5
哲学家	0.13	0.25	已屏蔽	0.39	0.13
数学家	0.24	0.15	0.24	已屏蔽	0.36
评论家	0.12	0.25	0.11	0.23	已屏蔽

表 4：邻接矩阵 3：gpt-4o-mini + 维基

	知识渊博的专家	搜索者	哲学家	数学家	评论家
知识渊博的专家	已屏蔽	0.73	0.86	0.73	0.46
搜索者	0.2	已屏蔽	0.76	0.57	0.3
哲学家	0.12	0.18	已屏蔽	0.75	0.5
数学家	0.2	0.25	0.19	已屏蔽	0.29
评论家	0.25	0.21	0.25	0.21	已屏蔽

表 5：邻接矩阵 4：Llama-3-70B + 维基

	知识渊博的专家	搜索者	哲学家	数学家	评论家
知识渊博的专家	已屏蔽	0.84	0.73	0.8	0.5
搜索者	0.13	已屏蔽	0.67	0.75	0.42
哲学家	0.2	0.22	已屏蔽	0.6	0.27
数学家	0.16	0.19	0.24	已屏蔽	0.35
评论家	0.25	0.24	0.2	0.23	已屏蔽

D.2 邻接矩阵的相关性分析

相关性分析

表 6：矩阵对之间的皮尔逊相关及其对应的强度和显著性。

比较	r	p	强度	Sig.
矩阵 1 对比 2	0.32	0.54	Weak	No
矩阵 1 对比 3	**0.98**	**0.001**	非常强	Yes
矩阵 1 对比 4	0.78	0.07	强	边际

E 算法工作流程

算法 1CARD 的工作流程：条件智能体图设计器

输入： 查询集 $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_D\}$，条件配置 $\{\mathcal{C}_1, \ldots, \mathcal{C}_C\}$，
图自动编码器 $f_\nu = (q_{\text{stat}}, q_{\text{dyn}}, \psi)$，参数为 $(\Theta_p \Theta_c, \Theta_d)$，学习率为 α

对于每个查询 $\mathcal{Q}_d \in \{\mathcal{Q}_1, \ldots, \mathcal{Q}_D\}$ **执行**
  对于每个条件 $\mathcal{C}_c \in \{\mathcal{C}_1, \ldots, \mathcal{C}_C\}$ **执行**
    /* 在条件 $\mathcal{C}_c$ 下构建智能体特征*/
    对于 智能体 $v_i \in \{v_1, \ldots, v_N\}$ **执行**
      $\mathbf{x}_i^p \leftarrow \mathcal{T}_p$(基础$_i$，角色$_i$，插件$_i$)
      $\mathbf{x}_i^c \leftarrow \mathcal{T}_c(\mathcal{C}_c[i])$
    结束循环
    $\mathbf{X}_p \leftarrow [\mathbf{x}_1^p, \ldots, \mathbf{x}_N^p]^\top$，　$\mathbf{X}_c \leftarrow [\mathbf{x}_1^c, \ldots, \mathbf{x}_N^c]^\top$
    $\mathbf{x}_\mathcal{Q} \leftarrow$嵌入$(\mathcal{Q}_d)$　▷ 查询被视为虚拟智能体节点
    定义锚定拓扑 $\mathcal{A}$（例如，全连接 +任务节点）
    $\widetilde{\mathcal{G}} \leftarrow (\{\mathbf{X}_p, \mathbf{X}_c, \mathbf{x}_\mathcal{Q}\}, \mathcal{A})$
    /* 通过编码器-解码器生成通信拓扑 */
    $\mathbf{H}_p \leftarrow \phi_p(\mathbf{X}_p \mid \mathcal{A})$，　$\mathbf{H}_c \leftarrow \phi_c(\mathbf{X}_c \mid \mathcal{A})$
    $\mathbf{S} \leftarrow \psi(\mathbf{H}_p, \mathbf{H}_c, \mathbf{x}_\mathcal{Q})$　▷ 计算链接概率
    $\mathcal{G}_{\text{com}} \leftarrow \{(i, j) \mid S_{ij} > \tau\}$　保留高于阈值的边
    /* $\mathcal{G}_{\text{com}}$ 下的多智能体协作*/
    for $t = 1$到 K **执行**
      对于智能体 v_i 在日程安排 $\phi(\mathcal{G}_{\text{com}})$ **执行**
        $\mathcal{P}_{\text{usr}}^{(t)} \leftarrow \{\mathcal{Q}_d\} \cup \{\mathcal{R}_j^{(t)} \mid v_j \in \mathcal{N}_{\text{in}}(v_i)\}$
        $\mathcal{R}_i^{(t)} \leftarrow v_i(\mathcal{P}_{\text{sys}}^{(t)}, \mathcal{P}_{\text{usr}}^{(t)})$
      循环结束
      $\alpha^{(t)} \leftarrow$聚合$(\{\mathcal{R}_i^{(t)}\}_{i=1}^N)$
    循环结束
    /* 优化图生成参数 */
    $\Theta \leftarrow \Theta - \alpha \cdot \nabla_\Theta \mathcal{L}_{\text{CARD}}$
  循环结束
循环结束

F 提示词

大语言模型动态信息模板

```
模型_模板 = {
    '名称': '{model_名称}',
    '描述': '{ModelName} 是一款由 {developer} 开发的 支持 {modalities}。'
            '它针对 {key_strengths} 进行了优化。'
            '{ModelName} 具有 {performance_advantage}。'
            '该模型的输入成本为每百万 ${input_cost} 每百万输出令牌的成本 令牌。'
            '{evaluation_info}'
}

evaluation_info = (
    '在 {domain_a} 中，{ModelName} 的准确率达到 {evaluation_score_a}。'
    '在 {domain_b} 中，{ModelName} 的准确率达到 {evaluation_score_b}。' ...
)
```

搜索引擎动态信息模板

```
搜索_引擎_模板 ={
    '名称': '{engine_名称}',
    '描述': '{EngineName} 是由 {提供商}.'
            '它支持{supported_query_types}，并提供 涵盖{content_scope}的结果。'
            '该引擎集成了{additional_features}，因此 适用于{application_scenarios}。'
            '{evaluation_info}'
}

评估_信息 = (
    '在{task_name}任务中，{EngineName}取得了 {score_value}分，{metric_name}指标得分。'
    ...
)
```

HumanEval 角色简介

```
“项目经理”:
    “你是一名项目经理。                                    "
    “用户将提供一个函数签名及其文档字符串，
    ↪  供你参考。"
    “你负责监督整体结构，确保
    ↪  代码的组织方式能够完成任务。
    ↪  简洁、正确地实现代码，不要一味追求
    ↪  过度设计。”
    “你需要推荐最佳设计模式，确保代码
    ↪  遵循最佳实践，以提升可维护性和灵活性。                                    "
    “你可以指定代码的整体设计，包括
    ↪  需要定义的类（可能没有）以及所使用的函数
    ↪  （可能只有一个函数）。”
    “我希望你的回复更加简洁。最好在五十
    ↪  字以内。不要列出太多要点。”,
“算法设计师”:
    "你是一名算法设计师。                                    "
    "用户将提供一个函数签名及其文档字符串，
    ↪  供你参考。"
    "你需要明确算法的具体设计，包括
    ↪  可能定义的类以及使用的函数。                                    "
    "你需要生成详细的文档，包括
    ↪  算法说明、使用说明和应用程序接口
    ↪  参考文献。                "
    “当实现逻辑较为复杂时，可以提供
    ↪  主要算法的伪代码逻辑。”
    “希望你的回复更加简洁，最好不超过五十
    ↪  个词。不要列出太多要点。”,
"编程专家":
    "你是一名编程专家。                                    "
    "用户会向你提供一个函数签名及其文档字符串，
    ↪  供你参考。"
    "你可能会获得其他智能体的输出结果。它们可能
    ↪  通过了内部测试，但不一定完全
    ↪  正确。                "
    "请写出完整的实现（重述函数签名）。                                    "
    "请使用 Python 代码块撰写回答。例如：
    ↪  例如：\n```python\nprint('Hello world!')\n```"
    "回答中只能包含 Python 代码块，不能有其他内容。
    ↪  回答。                "
    "请勿更改任务中的函数名和输入变量类型。",
"测试分析师":
    "你是一名测试分析师。                "
    "用户将向你提供一个函数签名及其文档字符串，
    ↪  用户。"
    "你需要指出当前代码或解决方案中存在的问题，
    ↪  针对问题中的测试数据和可能的测试反馈。                                    "
    “你需要补充一些特殊用例、边界
    ↪  条件等，指出编写
    ↪  代码。"
    “你可以指出代码中任何潜在的错误。”
    “希望你的回复更简洁一些，最好控制在五十
    ↪  词语。不要列出太多要点。”,
"漏洞修复人员":
    "你是一名漏洞修复人员。"
    "用户将向你提供一个函数签名及其文档字符串，
    ↪  供你参考。"                                    22
    "你需要根据所提供的
    ↪  当前整体代码设计、算法框架、代码
    ↪  实现或测试问题。                "
    "写出你的完整实现（重述函数签名）。                                    "
    "使用 Python 代码块撰写你的回复。对于
    ↪  示例：\n```python\nprint('Hello world!')\n```"
    "除 Python 代码块外，不要在你的
    ↪  回复。                "
    “请勿更改任务中的函数名和输入变量类型”,
```

MATH 角色配置

```
“数学求解器”:
    “你是一位数学专家。                    "
    “你将获得一道数学问题以及其他智能体提供的提示。 ”
    “请根据提示，逐步给出你自己的求解过程。                                    "
    "你的输出最后一行只能包含最终结果，不得包含
    ↪  任何单位，    例如：答案是 140\n"
    "你将获得一些可供参考的示例。",
"数学分析师":
    "你是一名数学分析师。                        "
    “你将获得其他
    ↪  智能体。    "
    “你需要先逐步分析求解过程，
    ↪  其中变量用字母表示。                                    "
    “然后将数值代入分析过程进行
    ↪  计算并得出结果。”
    “输出的最后一行只能包含最终结果，不要包含
    ↪  任何单位，    例如：答案是 140\n”
    “你将获得一些可供参考的示例。”,
“编程专家”:
    “你是一位编程专家。                        "
    “你将获得一个数学问题、分析以及来自其他
    ↪  智能体的代码。"
    “整合逐步推理与 Python 代码来求解数学
    ↪  问题。    "
    “分析该问题并编写函数以解决问题。                                          "
    “该函数不应接受任何参数，并使用最终结果
    ↪  作为返回值。          "
    “代码的最后一行调用了你编写的函数，并将
    ↪  返回值赋给 \(answer\) 变量。”                    "
    “使用 Python 代码块编写你的回答。例如:
    ↪  ```python\ndef fun():\n x = 10\n y = 20\n return x +
    ↪  y\nanswer = fun()\n```\n”
    “回复中只能包含 Python 代码块，除此之外
    ↪  不得包含任何内容。"
    “你会得到一些可供参考的示例。”,
“检查员”:
    “你是一名检查员。          "
    “你将获得一道数学问题、分析以及其他人的代码
    ↪  智能体。    "
    “检查解题过程中的逻辑/计算是否正确，以及
    ↪  分析过程（如有）是否正确。                    "
    “检查代码是否与解题分析相符（如
    ↪  有）。    "
    “根据提示逐步给出你自己的求解过程。                                        "
    "输出的最后一行只能包含最终结果，不得包含
    ↪  任何单位，    例如：答案是 140\n"
    "你将获得一些可供参考的示例。",
```

MMLU角色配置

"知识渊博的专家": """

你是一位擅长问答的知识渊博的专家。
请给出少于 3 个需要搜索的关键实体，
↪ 以解决该问题。每个实体都必须用 @ 包裹
↪ 符号。
例如：@鲶鱼效应@，@破窗效应@，@莎士比亚@。
如果问题中没有需要在
↪ 互联网上搜索的实体，就不必提供。
"""

"搜索者": """

你将收到一个问题，以及其中关键实体的互联网搜索概览
↪ 其中的实体。
请逐步参考这些内容作答。
"""

"评论家":
"""
你是一位出色的评论家。请逐条指出其他智能体分析中的潜在问题。↪"""

"数学家": """

你是一名擅长算术计算和长期规划的数学家。↪你可以运用逻辑和推理能力，逐步解决问题。
↪"""

"哲学家":
"""
你是一名在文学、历史和文化研究方面拥有深厚知识的哲学家。↪你会批判性地分析文本，提出细致入微的解读，并建立跨越时代、社会和学科的联系。↪"""

"医生": """

你是一名擅长生物学、医学和健康领域的医疗专业人士。你将现代医学与草药和自然疗法相结合。在提出每项建议时，你都会考虑年龄、生活方式和病史。"""

"程序员":
"""
你是一名擅长软件开发、系统设计和技术问题解决的程序员。↪你运用计算机科学、工程学和编程的原理。你能够在各种平台上编写简洁、高效的代码。"""

表 7：HumanEval 环境配置集

	配置	LLM	Role
训练与测试	配置 1	gpt-4o-mini	项目经理
		gpt-4o-mini	算法设计师
		gpt-4o-mini	编程专家
		gpt-4o-mini	测试分析师
		gpt-4o-mini	漏洞修复人员
	配置 2	deepseek-v3	项目经理
		deepseek-v3	算法设计师
		deepseek-v3	编程专家
		deepseek-v3	测试分析师
		deepseek-v3	漏洞修复人员
	配置 3	Llama-3-70B	项目经理
		Llama-3-70B	算法设计师
		Llama-3-70B	编程专家
		Llama-3-70B	测试分析师
		Llama-3-70B	漏洞修复人员
仅测试	配置 4	GPT-4o	项目经理
		GPT-4o	算法设计师
		GPT-4o	编程专家
		GPT-4o	测试分析师
		GPT-4o	漏洞修复人员
	配置 5	qwen-2.5-72B	项目经理
		qwen-2.5-72B	算法设计师
		qwen-2.5-72B	编程专家
		qwen-2.5-72B	测试分析师
		qwen-2.5-72B	漏洞修复人员

G 实验配置集

表 8：MATH 环境配置集

	配置	LLM	Role
训练与测试	配置 1	gpt-4o-mini	数学求解器
		gpt-4o-mini	数学分析师
		gpt-4o-mini	数学分析师
		gpt-4o-mini	编程专家
		gpt-4o-mini	检查员
	配置 2	deepseek-v3	数学求解器
		deepseek-v3	数学分析师
		deepseek-v3	数学分析师
		deepseek-v3	编程专家
		deepseek-v3	检查员
	配置 3	Llama-3-70B	数学求解器
		Llama-3-70B	数学分析师
		Llama-3-70B	数学分析师
		Llama-3-70B	编程专家
		Llama-3-70B	检查员
仅测试	配置 4	GPT-4o	数学求解器
		GPT-4o	数学分析师
		GPT-4o	数学分析师
		GPT-4o	编程专家
		GPT-4o	检查员
	配置 5	qwen-2.5-72B	数学求解器
		qwen-2.5-72B	数学分析师
		qwen-2.5-72B	数学分析师
		qwen-2.5-72B	编程专家
		qwen-2.5-72B	检查员

表 9：MMLU 环境配置集

	配置	LLM	Role
训练与测试	配置 1	gpt-4o-mini	数学家
		gpt-4o-mini	程序员
		gpt-4o-mini	评论家
		gpt-4o-mini	医生
		gpt-4o-mini	心理学家
	配置 2	deepseek-v3	数学家
		deepseek-v3	程序员
		deepseek-v3	评论家
		deepseek-v3	医生
		deepseek-v3	心理学家
	配置 3	Llama-3-70B	数学家
		Llama-3-70B	程序员
		Llama-3-70B	评论家
		Llama-3-70B	医生
		Llama-3-70B	心理学家
仅测试	配置 4	GPT-4o	数学家
		GPT-4o	程序员
		GPT-4o	评论家
		GPT-4o	医生
		GPT-4o	心理学家
	配置 5	qwen-2.5-72B	数学家
		qwen-2.5-72B	程序员
		qwen-2.5-72B	评论家
		qwen-2.5-72B	医生
		qwen-2.5-72B	心理学家

表 10：使用外部工具的 MMLU 环境配置集

	配置	LLM	Role	外部工具（搜索引擎）
训练与测试	配置 1	gpt-4o-mini	知识渊博的专家	谷歌
		gpt-4o-mini	搜索者	
		gpt-4o-mini	心理学家	
		gpt-4o-mini	数学家	
		gpt-4o-mini	评论家	
仅测试	配置 2	GPT-4o	知识渊博的专家	谷歌
		GPT-4o	搜索者	
		GPT-4o	心理学家	
		GPT-4o	数学家	
		GPT-4o	评论家	
	配置 3	gpt-4o-mini	知识渊博的专家	Wiki
		gpt-4o-mini	搜索者	
		gpt-4o-mini	心理学家	
		gpt-4o-mini	数学家	
		gpt-4o-mini	评论家	
训练与测试	配置 4	Llama-3-70B	知识渊博的专家	Wiki
		Llama-3-70B	搜索者	
		Llama-3-70B	心理学家	
		Llama-3-70B	数学家	
		Llama-3-70B	评论家	