基于多智能体协作的复杂任务求解系统设计与实现 - 开题报告

1. 任务描述

1.1 研究背景

单一模型架构在处理多跳推理任务时仍面临三大核心挑战：**(1)推理链断裂，导致逻辑错误；(2)资源消耗与性能之间的权衡困境；(3)领域专业知识的不均衡分布。**

现有多智能体系统在三个关键方面仍存在明显不足：**(1)智能体间通信效率低下**；**(2)缺乏针对轻量模型特性的优化策略**；**(3)评测体系单一，难以全面反映系统性能。**

1.2 研究任务

1. **协作框架设计与实现**：构建一个可扩展的多智能体协作框架，实现基于有向无环图(DAG)的任务分解与基于能力画像的动态角色分配，支持同步/异步混合执行模式。
2. **轻量模型协同机制研究**：探索参数规模在1B-7B之间的轻量化模型协作策略，包括专家混合(Mixture-of-Experts)、知识蒸馏(Knowledge Distillation)和提示词优化(Prompt Engineering)，验证"小模型协作"对性能的提升效果。
3. **多维度评测体系构建**：设计涵盖结果正确性、推理过程质量、资源效率和鲁棒性的综合评测框架，包括精确匹配(Exact Match)、分步赋分(Progress-Supervised)、代码执行验证和对抗样本测试。
4. **系统效率优化**：通过计算图优化、智能缓存、批处理技术和模型量化，在保证性能的前提下降低响应时间与资源消耗，实现边缘设备部署的可能性。

2. 预期目标

- **框架实现**：构建支持4类核心角色（规划者、执行者、检查者、反思者）的多智能体协作框架，实现可视化的交互流程监控和动态调整机制。
- **性能提升**：
  - GSM8K数据集：准确率达到78%以上，较单一轻量模型(Phi-3)提升20个百分点，接近GPT-4性能的90%。
  - HotpotQA数据集：F1分数达到75%以上，较单一模型提升15%，推理步骤正确率提升25%。
  - HumanEval数据集：通过率达到65%以上，较单一7B模型提升20%，代码质量评分提升30%。
- **效率优化**：
  - 系统平均响应时间控制在3秒内（简单任务）至8秒内（复杂任务）
  - Token消耗较单一大模型降低40%，GPU内存峰值降低50%
- 标准化API接口（RESTful/WebSocket/gRPC）

3. 相关工作

1.学习一些轻量化模型的应用，如：

1. **Phi-3**（微软，2024）：最新推出的4B参数模型，采用高质量数据和先进训练技术，在MMLU等常识推理任务中接近7B模型性能，是轻量化协作的理想选择。
2. **TinyLlama**（香港科技大学，2023）：1.1B参数模型，通过3万亿token的高效训练策略实现低成本部署，在简单任务上表现出色。其创新的训练方法和模型结构为小模型协作提供了技术基础。
3. **DeepSeek-Math**（深度求索，2023）：专注数学问题的7B模型，通过数学语料预训练和多轮思维链微调，在GSM8K上表现优于同规模通用模型，为特定领域的专家模型提供了范例。
4. **CodeLlama**（Meta，2023）：针对代码生成优化的模型系列，7B版本在HumanEval上达到38%的通过率，在编程任务上具有较高性价比，支持多种编程语言和长上下文推理。

2.学习一些任务评测方法，如：

1. **Exact Match**：广泛用于数学问题评测的基础指标，但忽略中间步骤正确性，难以反映推理过程质量。近期研究表明，仅依赖此指标可能导致模型优化方向偏离实际应用需求。
2. **HumanEval**（OpenAI，2021）：基于单元测试的代码生成评测标准，覆盖164个Python编程问题，全面评估代码功能正确性。然而，其未考虑代码可读性、效率和安全性等维度。
3. **Progress-Supervised**（DeepMind，2023）：新兴的分步评测方法，对解题过程中的每一步进行评分，更全面反映模型推理能力。该方法通过人类标注的中间步骤作为监督信号，提供了更精细的评测标准。
4. **Robustness Evaluation**（Stanford，2023）：通过对抗样本和输入扰动测试模型鲁棒性，评估系统在非标准输入下的表现。这一方法对多智能体系统尤为重要，可检测协作过程中的错误累积和放大效应。

3.了解多智能体协作的研究，如：

1. **AutoGen**（微软，2023）：提出了基于对话的多智能体框架，支持自定义角色与群组聊天，在代码生成任务上取得显著成果。然而，其通信协议缺乏结构化设计，且未针对轻量模型进行优化，在资源受限环境下效率较低。
2. **MetaGPT**（DeepWisdom，2023）：通过标准化工作流程和软件工程原则提升代码生成任务的效果，引入了基于UML的规划机制。但其主要依赖13B以上大模型，且评测指标单一，难以反映系统在不同任务上的表现差异。
3. **ChatDev**（香港中文大学，2023）：面向软件开发的多智能体系统，模拟产品经理、架构师、程序员等角色协作，在端到端应用开发上表现出色。但其专注于代码生成而非通用任务求解，且缺乏对中间过程的质量评估。
4. **CAMEL**（斯坦福大学，2023）：通过自主智能体间通信实现协作，引入了角色扮演和任务驱动对话。但缺乏明确的角色分工和任务分解机制，在复杂任务上容易出现目标偏移。

4. 技术方案

4.1 系统架构设计

四类智能体：

1. 规划者（Phi-3-mini，3.8B参数）：任务分解与资源分配
2. 执行者（动态路由至DeepSeek-Math-7B（数学）、CodeLlama-7B（代码）或通用模型）：专业任务求解
3. 检查者（TinyLlama-1.1B（量化至INT8））：结果验证
4. 反思者（Qwen1.5-4B（通过API调用））：错误分析优化

4.2 工作流模式：

系统采用基于有向无环图(DAG)的工作流模型，支持以下交互模式：

1. **线性工作流**：用户输入 → 规划者分解 → 执行者求解 → 检查者验证 → 反思者分析 → 结果整合 → 用户输出
2. **分支工作流**：规划者将任务分解为多个并行子任务，由多个执行者同时处理，检查者对各分支结果进行整合验证
3. **迭代工作流**：检查者发现错误后，反思者分析原因，规划者调整方案，重新进入执行阶段
4. **自适应工作流**：系统根据任务复杂度和历史性能动态调整工作流，如简单任务可跳过部分环节

4.3 注意效率与结果优化的平衡

4.4 评测体系设计

1. **结果正确性**：
    - Exact Match：最终答案精确匹配率
    - F1 Score：答案内容重合度
    - Domain-Specific Metrics：领域特定评价指标（如数学问题的数值精度）
2. **效率指标**：
    - 响应时间：从用户输入到系统输出的总时间
    - Token消耗：模型调用的总token数
3. **过程质量评估**：
    - 逐步赋分：按中间步骤匹配度动态评分（0-100分）
    - 推理链完整性：评估推理过程的逻辑连贯性
    - 错误定位准确率：系统自我纠错的准确程度
4. **鲁棒性评估**：
    - 对抗样本测试：在有意干扰的输入下的表现
    - 边界条件处理：极端情况下的系统行为
    - 错误恢复能力：从错误状态恢复的成功率

5. 开发环境

- 编程语言：Python 3.10+
- 核心依赖库：将在后续开发中不断调整
- 本地部署模型：
    - Phi-3-mini-4K (3.8B)：GGUF格式，INT8量化，规划者角色
    - TinyLlama-1.1B：GGUF格式，INT4量化，检查者角色
    - llama.cpp / vLLM 作为推理引擎，支持量化和批处理
- 云API服务：
    - 阿里云DashScope：Qwen1.5-4B/7B，反思者角色
    - Hugging Face Inference API：DeepSeek-Math-7B、CodeLlama-7B，执行者角色

- OpenAI API：GPT-4 Turbo，用于对比实验和复杂任务回退

6. 实施计划

6.1 阶段划分

1. **准备阶段（1周：5.09~5.16）**
   - 相关内容调研：学习使用一些轻量化模型，学习一些测评方法
   - 数据准备：预处理GSM8K、HotpotQA、HumanEval数据集
   - 确定预期目标、开发环境和技术方案
2. **前期开发阶段（2周：5.16~5.30）**
   - 框架实现：构建多智能体协作基础框架，实现四类智能体及其交互协议
   - 确定模块划分和模块基础设计，按照确定的技术方案完成原型系统开发
3. **核心开发阶段阶段（1周：5.30~6.06）**
   - 完成系统的详细设计和功能的实现，给出实验结果
   - 性能优化：实现异步执行、缓存系统，优化提示词模板，调整模型参数
   - 评测系统：构建多维度评测框架
4. **系统优化与文档阶段（1周:6.06~6.13）**
   - 系统优化：逐步完成系统优化，提高系统效率
   - 文档编写：完成技术文档和用户指南

6.2 任务分配和检查点

姓名	任务分工	检查点1	检查点2	检查点3	检查点4
徐焜	结题报告、架构与智能体开发：环境搭建、架构设计、框架选型、规划者/执行者智能体实现	完成环境搭建和基础架构设计	实现规划者智能体核心功能	完成执行者动态路由系统	系统集成与性能调优
童宇捷	中期报告、基线测试设计、检查者/反思者智能体实现、缓存系统开发	定义评测指标和基线测试方案	实现检查者智能体验证机制	完成反思者智能体错误分析	缓存系统实现与性能分析
刘千翔	开题汇报、数据集预处理、API接口设计、前端交互原型、示例应用开发	完成数据集预处理	设计并实现基础API接口	开发前端交互原型	完成示例应用和用户体验测试