SwarmFoam：基于多种类型大语言模型的 OpenFOAM 多智能体系统

多种大语言模型</numerusform

Chunwei Yang[a]，Yankai Wang[a]，Jianxiang Tang[a]，Haojie Qu[a]，Ziqiang Zou[b]，YuLiu[b]，Chunrui Deng[b]，Zhifang Qiu[b, *]，Ming Ding[a, *]

[a]黑龙江省核电系统与设备重点实验室，哈尔滨工程大学，中国哈尔滨，150001[b]先进核能技术国家重点实验室，中国核动力研究设计院，中国成都，610041

摘要

数值模拟是科学研究中的主流方法之一，通常由专业工程师执行。随着多智能体技术的发展，利用协作智能体复现人类行为，在实现智能计算流体力学（CFD）仿真方面展现出巨大潜力。目前已有一些基于大语言模型的多智能体系统被提出。然而，它们在处理复杂几何结构时存在明显局限。本文介绍了一种新的多智能体仿真框架 SwarmFoam。SwarmFoam 集成了多模态感知、智能纠错和检索增强生成等功能，旨在通过对图像和高级指令进行双重解析，实现更复杂的仿真。实验结果表明，SwarmFoam 对不同模态的仿真输入具有良好的适应性。在 25 个测试案例中，总体通过率为 84%；自然语言和多模态输入案例的通过率分别为 80% 和 86.7%。SwarmFoam 的这项工作将进一步推动面向 CFD 的智能体方法发展。

关键词：多智能体系统、大语言模型、OpenFOAM、计算流体力学、多模态模型

1. 引言

计算流体力学（CFD）是一门利用数值分析方法，通过计算机求解流体动力学方程，从而实现物理现象仿真的学科。早期计算流体力学研究主要聚焦于建立基础理论，例如有限元法、有限差分法和格子玻尔兹曼方法。随着计算机性能的提升，计算流体力学已广泛应用于航空航天工程（Ferro et al., 2023年; Lusher et al., 2025年）、能源与动力工程（Jansasithorn et al., 2025年; Xin et al., 2025年）和气象预报（Suárez-Vázquez et al., 2025年; Xie et al., 2025年）等领域，推动了相关数值方法的显著进步。随着人工智能（AI）技术日趋成熟，人工智能与数值计算的跨学科融合已成为计算流体力学领域最新的研究课题之一。

2024年之前，关于人工智能在计算流体力学（CFD）中应用的研究主要集中于求解偏微分

方程组，包括数据驱动方法（Lu 等，2021；Mao 等，2023年）和基于物理的方法（Raissi 等，2019）。尽管有些研究采用深度学习方法构建自动化智能体，但这些工作的研究方法仅限于湍流建模（Novati 等，2021）和网格生成（Kim 等，2025年）等特定任务。在应用方面，这些研究仅适用于单一场景（Gbadago 等，2025年），无法用于范围广泛的 CFD 模拟。

随着生成式人工智能技术日趋成熟，大语言模型（LLM）为计算流体力学开辟了新的研究范式。多智能体系统通过在同一工作环境中协调不同的智能体角色，按照预定义工作流智能调用外部工具，以完成特定任务。这种方法在自动驾驶（M. Zhang et al., 2025; P.Zhang et al., 2025）、临床诊断（Liu et al., 2026; Qiu et al., 2024）和建筑设计（Chen & Bao, 2025; Y. Dong et al., 2025）等复杂场景中已取得显著成效。一些学者已开始探索利用大语言模型或多智能体系统为计算流体力学软件 OpenFOAM 创建智能体，旨在仅依靠自然语言或高级指令实现流体力学仿真。

OpenFOAM 是功能最强大、应用最广泛的开源CFD平台之一。与商业计算流体力学软件不同，OpenFOAM 依靠配置文件驱动流体动力学模拟的所有阶段，且不依赖用户交互界面，因此是基于智能体的自动化的理想目标。然而，由于几何构建和网格生成过程复杂，且文件格式受到严格限制，要实现完全自动化的智能计算流体力学工作流仍面临独特挑战。因此，目前相关技术工作仍处于竞争激烈的初步研发阶段。

2024年，清华大学发布了全球首个面向计算流体力学的开源多智能体系统 MetaOpenFOAM-1（Chen 等，2024）。该系统采用 MetaGPT（Hong 等，2023）多智能体框架，支持自定义智能体角色和行为。MetaOpenFOAM-1 定义了多个智能体角色，分别代理数值计算中的四个步骤：算例分析、预处理、执行计算和错误修正。每个智能体都配备了通过 LangChain 编程框架（Jeong 等，2024）构建的问答接口，可与包含 OpenFOAM帮助文档的本地知识库交互。2025年，研究人员进一步提出了 MetaOpenFOAM-2（Chen 等，2025）。新版本新增了任务分解功能（将用户需求拆分为仿真、后处理和模拟优化等子任务），并增强了后处理功能。

同年，英国埃克塞特大学提出了另一种智能仿真系统——OpenFOAMGPT（Pandey 等，2025）。OpenFOAMGPT 利用大语言模型的原生能力，在不采用多智能体框架的情况下实现了智能 OpenFOAM 工作流。随后，研究人员开发了 OpenFOAMGPT-2（Feng 等，2025），该系统同样引入了多智能体技术，通过设计专门负责预处理、提示生成、软件代理和后处理的智能体。与 1.0 版本相比，OpenFOAMGPT-2 显著提高了仿真任务的成功率。

借鉴 MetaOpenFOAM 和 OpenFOAMGPT 的开发经验，伦斯勒理工学院提出了第三个主流 FOAM 多智能体系统：*Foam-Agent*（Yue 等，2025a；Yue 等，2025b）。其工作流与 MetaOpenFOAM 系列相似，因为二者都利用 LangChain 实现本地知识库问答接口。二者的区别在于计算流体力学工作流设计：*FOAM-Agent*采用了另一种流行的多智能体

框架 LangGraph（Wang & Duan，2024年）。与 MetaGPT 相比，LangGraph 在智能体开发方面提供了更大的灵活性，但开发复杂度也更高。Foam-Agent 的核心在于提出了一种更科学的多文件依赖关系感知与编写方法：模型在生成当前文件时，会参考相关的配置内容。这种文件生成方法更符合 OpenFOAM 的格式特征，从而实现了更高的计算成功率。

然而，现有的计算流体力学智能体系统完全依赖自然语言来驱动计算流体力学模拟。当模拟对象的几何体和物理条件变得更加复杂时，仅依靠自然语言输入很难准确描述仿真问题。为解决这一问题，SwarmFoam 首次引入了观察者智能体。基于我们设计的多模态感知机制，观察者智能体能够同时理解输入中的图像和文本模态所表达的仿真任务，这与真实的仿真过程更加相似。此外，SwarmFoam 还包含负责代理计算流体力学模拟各个阶段的其他智能体，包括预处理、运行、错误修正和后处理。检索增强生成（RAG）系统提升了模拟配置文件的质量，智能体可以从中获取参考算例、求解器使用帮助以及文件命名规范等信息。本文的主要贡献总结如下：

- 多模态感知机制：在实际工作中，仿真信息通常同时以图像和文本形式传递。为了有效处理模拟图像，我们提出了一种针对计算流体力学模拟设计的多模态感知机制，并引入了新的观察者智能体。该智能体结合自然语言输入，解析图像中的几何与物理信息，用于支持网格文件的创建。

- 首错优先智能纠错机制：在仿真错误修正过程中，我们采用首错优先策略。这使多个智能体能够分析并修正最可能导致当前模拟轮次失败的文件，从而避免分析由单一根本原因引发的次生错误。在处理复杂仿真任务时，该策略展现出显著的成本优势，因为它能有效减少令牌用量。

- 消融研究与错误分析：我们对关键机制进行了消融研究
  并对提出的两种多模态感知机制进行了比较分析，突显了图像预解析在提升整体仿真性能方面的关键作用。我们还讨论了禁用智能纠错机制时的仿真性能。我们对迭代过程中遇到的所有仿真错误进行了统计分析，分析了主要错误原因，并讨论了大语言模型在处理计算流体力学模拟任务时存在的一些局限性。

本文的主要结构如下。在第2.1节中，我们概述了*SwarmFoam*的智能模拟工作流架构，并展示了各智能体的角色划分。在第2.2节中，我们介绍了检索增强生成系统的详细内容。在第2.3节中，我们详细说明了*SwarmFoam*中各智能体的角色与动作设计，以及信息交互机制。在第2.4节中，我们展示了在不同输入模态下进行智能模拟的完整过程。第3节介绍了用于评估*SwarmFoam*性能的实验，包括数据集设置与结果讨论。在第4节中，我们总结了整个研究内容；在第5节中，我们提出了未来研究的建议。

面向计算流体力学的多智能体系统开发。

2. 方法论

2.1 智能模拟工作流架构

SwarmFoam的框架如图1所示。为有效实现计算流体力学任务的自动化，*SwarmFoam*设计了六种智能体角色：观察者、架构师、输入编写器、运行器、审查者和ParaMaster。具体而言，

(1) 观察者智能体：解析自然语言和算例图像输入。(2) 架构师智能体：根据观察者智能体解析的信息，规划完成当前仿真任务所需的文件结构。(3) 输入编写器代理：生成初始的 OpenFOAM 配置文件，或修正错误的配置文件。(4) 运行代理：执行仿真命令并捕获潜在的错误信息。(5) 审查代理：分析错误信息并确定错误配置文件的路径。

(6) ParaMaster 代理：编写代码以调用 ParaView 后处理器来输出图像。

在接收到用户的仿真需求后，上述多智能体系统中的各个智能体将根据其设计的角色、操作和工作流，自主执行 OpenFOAM 仿真。这些仿真智能体通过与环境中的信息进行交互来协作完成仿真任务，并输出仿真结果。

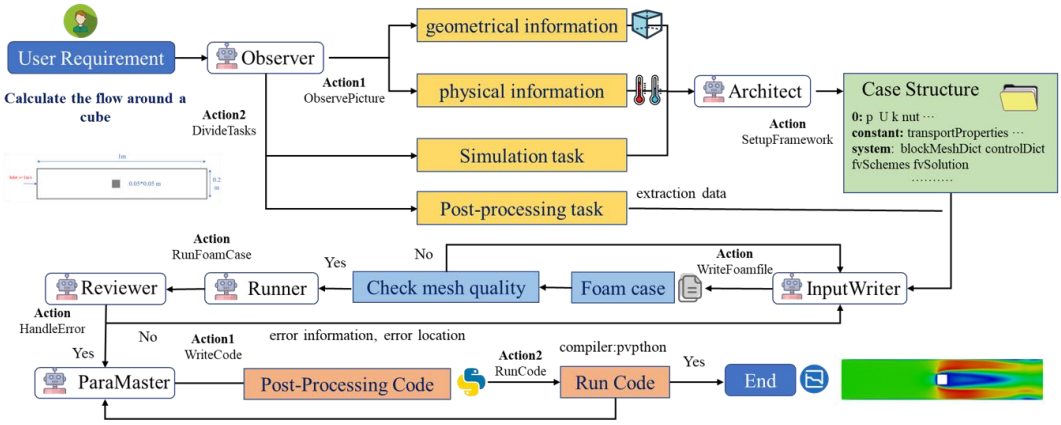


图 1. SwarmFoam 的智能仿真工作流。

2.2 检索增强生成系统

SwarmFoam 支持主流文本处理大语言模型，例如 Deepseek-R1、Deepseek- V3、gemini-2.5-pro、gpt-5-pro 和 gpt-4o，以及多模态大语言模型（MM- LLMs），包括 gemini-2.5-flash、通义千问系列和 Llama-4系列。为高效处理不同类别的任务，SwarmFoam 配置了两类大语言模型问答接口。第一类接口调用文本大语言模型，根据自然语言输入生成响应。第二类接口调用多模态大语言模型，根据多模态输入进行问答。

如表1所示， *SwarmFoam*总共使用了 6 份格式化帮助文档。前四类文档为参考算例文件："openfoam_allrun_files"包含 214

个 OpenFOAM-9 算例执行文件；"openfoam_cases_struct"包含 227 种算例结构；"openfoam_commands"包含 OpenFOAM-9 支持的 84 条仿真命令，例如网格生成命令（blockMesh）和网格质量检查命令（checkMesh）；"Openfoam_input_files"包含 3727 个算例配置文件。其余文档包含求解器相关内容。在 OpenFOAM 中，求解器用于求解控制方程，对仿真执行至关重要。"solver_describe"简要介绍了 OpenFOAM-9 中的各类求解器，而"solver_help"则提供了各求解器的使用建议，例如物理量的维度、算例结构、适用的物理问题和文件命名规范。

检索增强生成的目的是检索上述本地专家知识，并将其与上下文一并提供给大语言模型，从而确保模拟配置文件更加合理、规范（Luo 等人，2026）。检索增强生成技术的基本原理如图 2所示。整个检索增强生成流程分为两个主要步骤：索引和查询。在索引阶段，仿真帮助文档被拆分为多个文本块。这些文本块经由加载器、向量模型和数据转换器处理，随后以向量形式存储在多个本地容器中。在查询阶段，向量数据库会检索与查询输入最相似的文本片段。然后，根据具体任务需求，将这些检索到的文本片段嵌入类型 1 或类型 2 问答界面的提示词中。

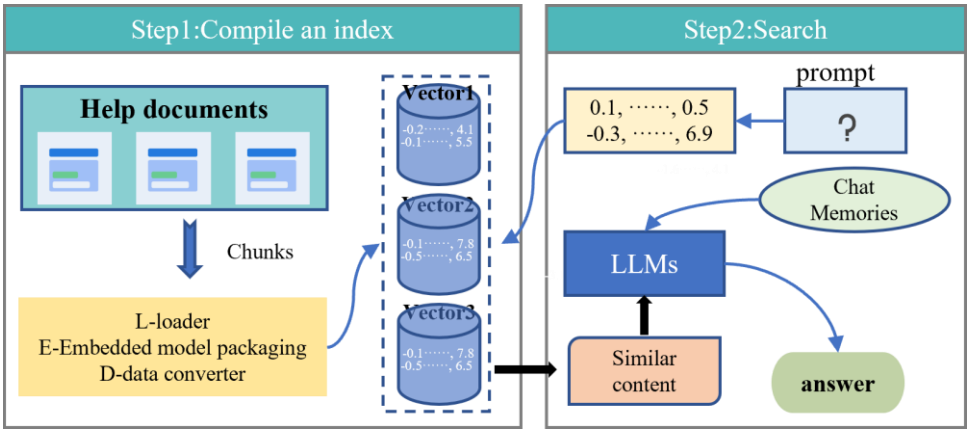


图2. 检索增强生成的基本原理。

表1 本地帮助文档列表。

文件名	内容
openfoam_allrun_flies.txt	执行文件参考
openfoam_cases_struct.txt	算例文件夹结构参考
openfoam_commands.txt	OpenFOAM-9 支持的仿真命令
openfoam_input_files.txt	本地算例配置文件
solver_describe.yaml	OpenFOAM-9 各类求解器简介
solver_help.txt	OpenFOAM-9 求解器用户指南

2.3 智能体设计

2.3.1 观察者智能体

观察者智能体仅订阅来自环境的“用户需求”类型消息，并发布解析后的模拟信息。当观察者智能体收到用户发布的信息时，该智能体便会被触发。观察者智能体依次执行两个动作：ObservePicture 和任务划分。ObservePicture 动作用于从算例图像中解析物理信息（例如边界条件、工作流体类型、初始条件）和几何信息（例如几何形状描述、具体尺寸、顶点位置）。任务划分动作将用户需求拆分为两个独立的子任务：仿真和后处理。如果用户输入中未包含具体的后处理需求，观察者智能体将不会发布后处理任务。

ObservePicture 动作依赖一种面向仿真的多模态感知机制。SwarmFOAM 最初设计了两种处理多模态信息的方法。第一种方法是，先由观察者对图像进行深入解析，以提取其中蕴含的仿真信息，然后将其发送给输入编写器以生成配置文件。在这一过程中，图像解析与几何构建彼此解耦；图像信息会先转换为文本信息。第二种方法是，将图像与用于创建网格文件 "blockMeshDict" 的提示词一同直接提交给输入编写器，从而跳过图像到文本的转换过程。

第一种多模态感知方法的工作流程如图 3所示。仿真需求以两种模态作为输入：文本和图像。以气泡上升算例为例，自然语言输入为：“计算一个涉及二维气泡上升的问题。总时长设为 2 秒。”自然语言输入未包含具体的几何和物理条件，因为这些信息体现在图像中。对于输入提示词，分词器和向量编码器会将其转换为特定维度的嵌入向量。输入图像则由图像编码器转换为相同维度的嵌入向量。由于这些向量的维度相同，因此可以直接拼接并传递给多模态大语言模型，从而实现对图像和自然语言的解析。

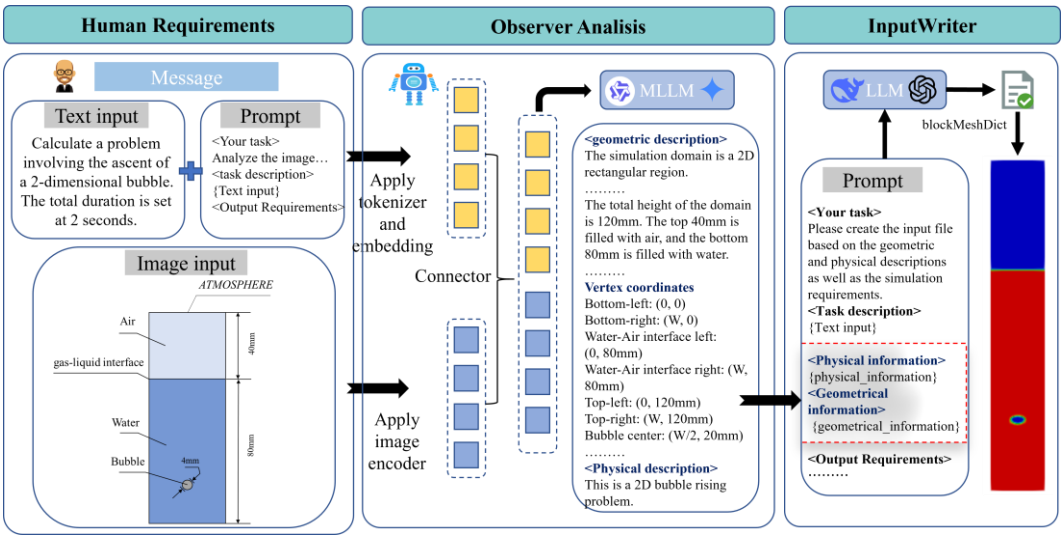


图 3. 多模态信息预解析流程图。

解析完成后，观察者会准确描述当前仿真任务，包括问题维度、工作流体、几何尺寸、顶点坐标和物理条件等细节。随后，这些信息会嵌入输入编写器的提示词中，用于生成“blockMeshDict”文件。

第二种多模态感知方法的工作流程如图4所示。在该方法中，观察者仅负责解析自然语言信息，而图像则直接提供给输入编写器。随后，多模态大语言模型同时执行图像理解、仿真需求理解以及网格文件（"blockMeshDict"）生成，无需进行任何预解析。该方法的优点是简化了图像处理工作流，但可能导致更多几何与物理信息的丢失。

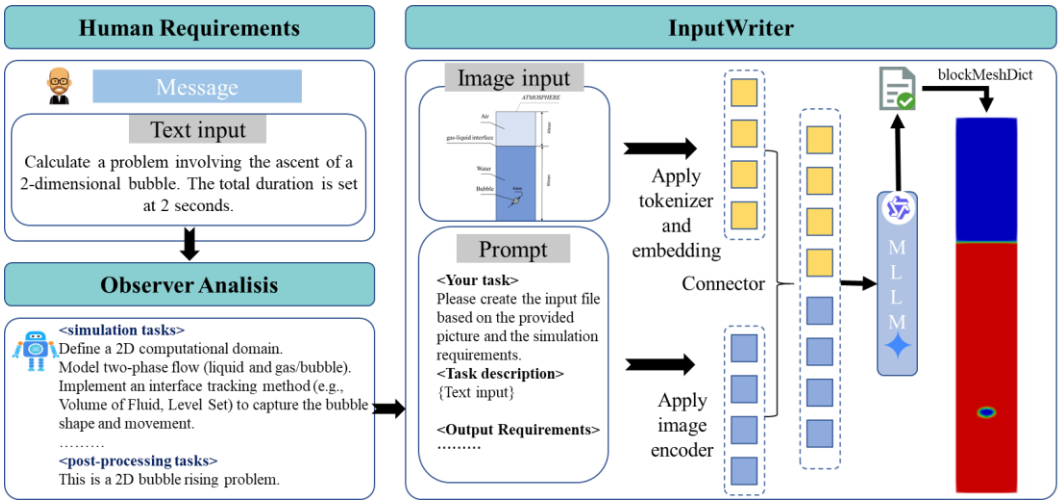


图4. 多模态信息直接利用流程图

在第3.2.3节所进行的消融分析中，讨论了预解析对整体仿真性能的影响。由于性能差异显著，*SwarmFoam*最终采用了第一种多模态感知方法。

多模态解析系统还可用于其他需要图像解析的场景，例如解读输出图像以评估仿真精度。

2.3.2 架构师智能体

架构师智能体订阅观察者智能体发布的消息，并发布一系列文件生成指令。接收到观察者智能体的消息后，架构师智能体执行SetupFramework操作。首先，智能体将当前仿真需求解析为一段包含五个要素的结构化信息：算例名称（例如Cavity）、算例领域（例如不可压缩）、算例求解器（适用于当前问题的求解器，例如pisoFoam）、算例类别（问题类型，例如雷诺平均模拟）以及求解器描述（对求解器的简要介绍）。

获得这些结构化信息后，架构师智能体会从本地知识库中检索最相似的算例结构。结合检索结果与仿真需求，架构师智能体定义完成当前任务所需的 OpenFOAM 算例文件结构。基础算例文件结构如表2所示。实际应用中，通常还需要许多

其他配置文件，具体取决于所要解决的问题。 .

表2 OpenFOAM 基础算例文件结构

文件夹	文件名	功能
0	p	定义压力场的初始条件和边界条件
	U	定义流速的初始条件和边界条件 字段
常量 transportProperties		定义输运参数（黏度）
系统	blockMeshDict	定义网格绘制参数
	控制字典	定义示例计算的控制参数
	有限体积格式字典	参数离散格式的定义
	fvSolution	定义参数的求解方法
root	Allrun	定义案例指令的执行

鉴于 OpenFOAM 配置文件之间存在相互依赖关系，架构师智能体会依次发布文件生成指令。对于不同目录中的文件，子任务优先级从高到低依次为：「system」、「constant」、「0」和「Allrun」。在「system」目录中，子任务优先级依次为「blockMeshDict」、「controlDict」、「fvSchemes」、「fvSolution」，然后是其他文件。在「constant」目录中，优先处理「transportProperties」，然后是其他文件。「0」目录中的所有文件优先级相同。

2.3.3 输入编写器代理

输入编写器代理订阅架构师智能体和审查代理（来自 HandleError 操作）发布的消息，并向环境发布仿真执行命令。输入编写器代理的操作 WriteFoamfile 设计为两个子任务：FirstWrite 和 CorrectFile。输入编写器代理会根据收到的消息来源，选择要执行的子任务类型。

如果当前消息由架构师智能体发布，表明仿真工作流处于初始配置文件生成阶段，则输入编写器代理会执行 FirstWrite 子任务。FirstWrite 负责生成表2中列出的一系列配置文件。每个文件的生成都包含四个步骤：从本地知识库检索、检查依赖文件、生成配置文件以及文件清理。在从本地知识库检索这一步中，智能体会搜索相似案例和求解器帮助文档。检查依赖文件旨在确保不同配置文件中的边界条件、边界命名和单位设置保持一致。文件清理这一步旨在移除大语言模型输出中的残留标识符（例如文件开头的 bash、Foam）。这些不符合 OpenFOAM 文件规范的标识符，是导致仿真失败的一个重要原因。

如果当前消息由审查代理发布，输入编写器代理便会执行 CorrectFile 子任务。该子任务根据收到的错误文件路径和错误描述，修正错误文件或创建缺失文件。此错误纠正过程同样遵循 FirstWrite 的四个步骤。

2.3.4 运行代理

运行代理订阅输入编写器代理发布的消息，并相应地向环境发布成功标志或错误内容。它是一个单动作智能体，负责执行 RunFoamCase 操作。触发后，运行代理会执行“Allrun”文件以启动算例模拟，并记录仿真命令的执行顺序。运行结束后，算例目录中会生成一系列日志文件。如果仿真失败，运行代理会依次从各个仿真日志文件中提取错误信息。

2.3.5 审查代理

审查代理订阅运行代理发送的消息，负责发布解析后的错误信息及其对应的文件路径。该智能体包含两个动作：HandleError 和 EndMark（算法1）。如果仿真成功完成且需要后处理，审查代理将执行 EndMark 动作，并直接向 ParaMaster 代理发布消息。如果仿真成功且无需后处理，仿真工作流即告终止。

Algorithm 1 Error Diagnosis and Correction

Require: Message history H, Configuration C
Ensure: Error diagnosis D, Error file path F

$(checkMeshfailed) \leftarrow$ Parse_Input(H, C) ▷ Initialize paths and data
$LLM_Agent \leftarrow$ Initialize_Agent() ▷ Setup LLM interface
if $Simulation$.successful **then** ▷ Check for no-error signal
  return Next Step, No Error
end if
▷ — Tier 1: Handle Specific, Known Errors —
if NOT Exists($task_path$ + /log.blockMesh) **then**
  $(D, F) \leftarrow$ Handle_BlockMesh_Error($task_path$)
  return D, F
end if
if Exists($Timeprecisionerror$) **then**
  $(D, F) \leftarrow$ Handle_TimeFormat_Error($task_path$)
  return D, F
end if
▷ — Tier 2: General Runtime Error Diagnosis —
Restore_State($files, task_path$) ▷ Reset to pre-run state
$first_err \leftarrow errors[0]$
$(D_{parsed}, F_{parsed}) \leftarrow$ Parse_Error_Log($first_err$) ▷ Initial error parsing
$error_type \leftarrow$ Classify_Error_Type($LLM_Agent, first_err, files$) ▷ Use LLM to classify
if $error_type$ = format error **then**
  $(D, F) \leftarrow$ Diagnose_Format_Error($LLM_Agent, \ldots$)
else if $error_type$ = Missing file **then**
  $(D, F) \leftarrow$ Diagnose_Missing_File($LLM_Agent, \ldots$)
else
  $(D, F) \leftarrow$ (Unknown type error)
end if
return D, F ▷ Return final diagnosis

如果仿真失败，智能体将执行 HandleError 操作。智能体首先验证几何体是否已成功创建，以及网格质量是否可接受。如果这些

要求未得到满足，它就会发布指令，要求重新生成 blockMeshDict 文件。在 OpenFOAM 仿真过程中，几何创建和求解器执行等命令会依次执行，每条命令都会生成一个日志文件。如果几何体已成功创建，检查员智能体会删除运行期间生成的文件，然后分析收到的错误日志内容。

与 Foam-Agent 和 MetaOpenFOAM 不同，SwarmFoam 采用了截然不同的错误修正原则。在现有系统中，审查代理会汇总日志文件中的所有错误，一并提交给大语言模型进行分析，随后向输入编写器代理发送一系列修改指令。然而，SwarmFoam 采用“首错优先”原则。

在每次修正迭代中，智能体会假定遇到的首个错误是导致当前仿真失败的主要原因，并且仅提供该错误对应的原因和路径。这一原则避免了对初始故障引发的派生错误进行不必要的分析。在此分析过程中，仿真错误分为两大类：文件缺失和格式错误。分析完成后，审查代理会向输入编写器代理发送消息。

由于表示仿真成功和仿真失败的消息来自不同的动作类型，因此审查代理完成任务后，只会触发一个特定的下游智能体。

2.3.6 ParaMaster 代理

ParaMaster 代理订阅由审查代理发布的消息（源自 EndMark 动作）。SwarmFoam 通过编写 Python 代码来调用 ParaView 程序，实现后处理功能。ParaView 是广泛用于可视化 OpenFOAM 算例的程序，并提供可调用的 Python 接口 pvpython。因此，SwarmFoam 中的 ParaMaster 代理本质上是专用于 ParaView 的代码生成器。

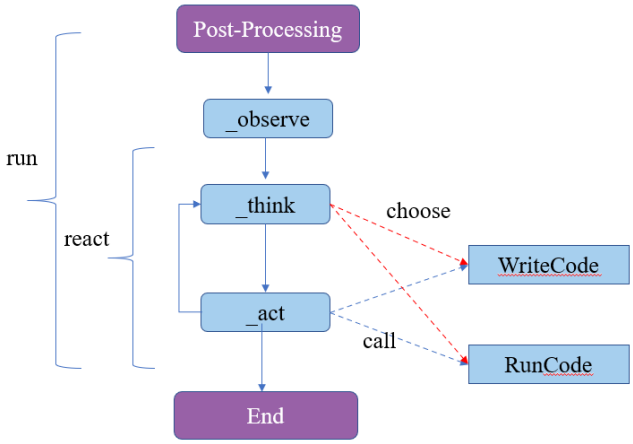


图 5. ParaMaster 代理工作流程

如图 5所示，它包含两种操作：WriteCode 和 RunCode。WriteCode 负责编写和修正代码，RunCode 用于执行代码。后处理智能体利用推理机制来决定在每次迭代中

执行哪种类型的操作。如果智能体在推理过程中判断无需采取进一步操作，则整个后处理工作流结束，并输出生成的图像。后处理完成后，整个基于智能体的仿真过程随即终止，因为环境中没有其他可供订阅的消息。

2.4 模拟演示

2.4.1 自然语言输入

图 6展示了一个多组分燃烧案例（甲烷射流）的基于智能体的仿真，其中仿真需求通过自然语言输入提供。用户的自然语言输入简要描述了计算条件（例如，基础几何体、喷射口、流动方向和工作流体类型）。由于输入不包含图像，观察者智能体会跳过图像解析过程；因此，“几何描述”和“物理描述”字段均设为“None”。

根据用户输入，观察者智能体对当前仿真需求（称为“仿真任务”）进行详细分析，其中包括求解器选择、初始条件以及边界条件的应用。随后，观察者智能体将解析后的仿真信息发送给架构师智能体。架构师智能体进行初步规划，以确定模拟多组分燃烧问题所需的配置文件。这些文件包括初始质量分数场（CH4、N2、O2）的文件；初始压力（p）、温度（T）和速度（U）场的文件；以及物理属性、化学反应、热物理性质、网格和求解器设置的文件。

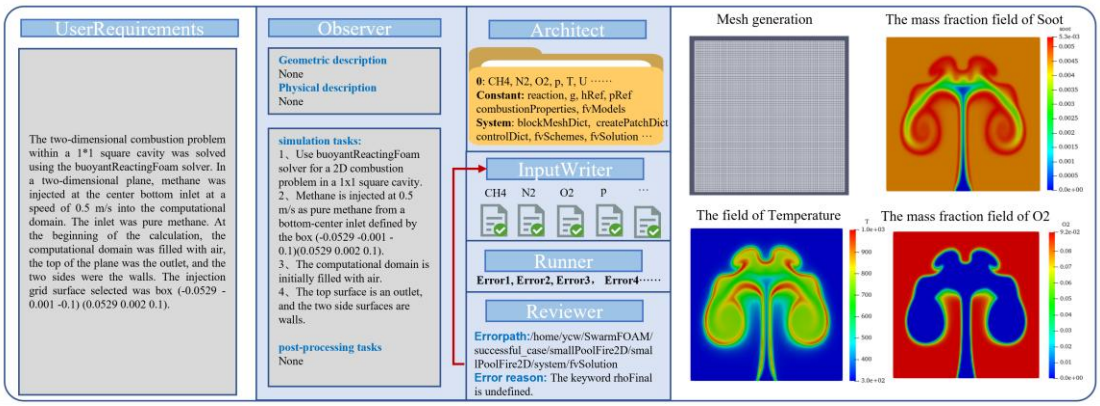


图 6. 自然语言输入的模拟演示

建立算例架构后，输入编写器智能体会依次生成配置文件，并将其放置在相应路径下。运行器智能体执行“Allrun”文件以运行仿真案例，并依次从终端和日志文件中捕获一系列错误。审查者智能体会分析第一个错误。如图所示，该智能体判定此次仿真失败的关键原因是压力方程设置中缺少“rhoFinal”参数，并指出导致此错误的文件路径（fvSolution）。此错误信息会发送至环境。在下一轮迭代中，输入编写器智能体将重写“fvSolution”文件中的“rhoFinal”关键字设置。

通过迭代式错误纠正，SwarmFoam 成功模拟了多组分 t

二维腔室中的射流燃烧。在 t = 0.9s 时，结果显示，烟炱质量分数、温度和氧气浓度呈现相似的空间分布。随着甲烷不断注入，燃烧过程逐渐加剧，促使火焰传播并产生火焰涡旋。在燃烧区，氧气浓度降低，温度升高，并生成烟炱。

2.4.2 多模态输入

图7展示了一个使用多模态输入（自然语言和图像）提供需求的案例模拟演示。图中描绘了气流流经障碍物的情形。在解析图像和自然语言后，观察智能体提供了仿真所需的几何与物理信息。

在几何信息方面，观察智能体将计算域识别为一个长 1200 mm、高 200 mm 的二维矩形。计算域的底部边界上放置了一个障碍物，其几何尺寸和位置均得到了准确描述。在对整体几何体进行初步解析后，观察者进一步详细检查了每个边界，并提供了其坐标范围和类型。如图所示，观察者将计算域的左侧边界识别为“入口”，右侧边界识别为“出口”，其余边界识别为“壁面”。这些识别出的边界条件与输入案例图像一致。

在物理信息方面，观察者将该问题识别为一个二维、等温、湍流、不可压缩流案例，并将使用 pisoFoam 求解器进行仿真。此外，观察者还提供了工作流体类型、流速、温度和压力等详细信息。

观察阶段之后的智能体工作流与图 6所示类似。*SwarmFoam*规划了所需的文件类型并生成了配置文件。执行后，智能体报告了错误原因及其对应路径。通过迭代式错误纠正，*SwarmFoam*成功模拟了障碍物流动过程。根据后处理要求，ParaMaster 代理生成了速度场和压力场分布图。从结果可以看出，由于流动横截面积减小，障碍物下游上方区域出现了显著的加速效应，而下方区域则形成了低速区。

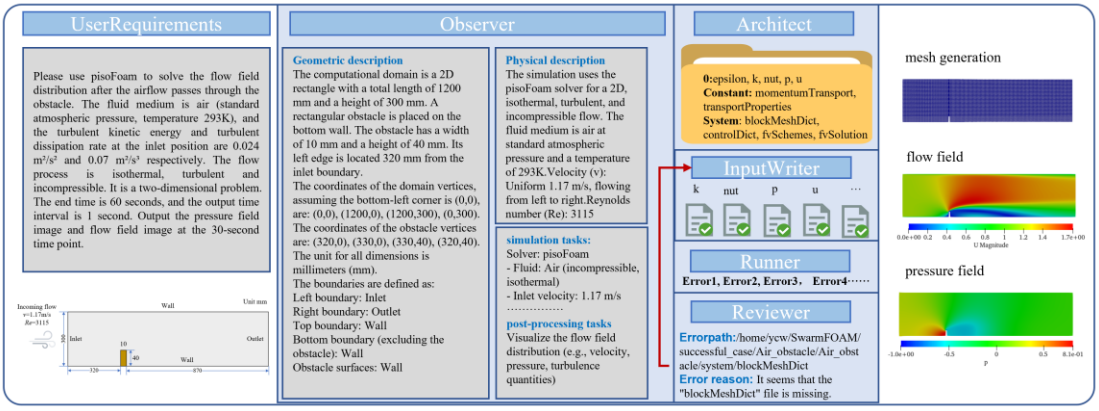


图 7. 多模态输入的模拟演示

3. 实验测试与应用

3.1 实验设置

3.1.1 数据集

如表3所示，我们建立了25个独立测试案例，用于评估*SwarmFoam*的仿真性能和图像解析能力。该数据集包括10个自然语言输入案例和15个多模态输入案例，涵盖计算流体力学中的多种物理问题，包括单相流、两相流、磁流体力学、燃烧化学反应等。

表3 测试数据集

输入模态	问题类型	求解器名称	Num
自然语言	不可压缩	simpleFoam	1
		pisoFoam	2
		非牛顿流体 icoFoam	1
	燃烧	反应流 foam	2
	基础	laplacianFoam	1
	金融	金融 foam	1
	电磁学	磁流体动力学 foam	1
	DNS	直接数值模拟 foam	1
多模态	不可压缩	pisoFoam	2
		simpleFoam	3
		interFoam	1
		solidDisplacementFoam	1
		icoFoam	2
		rhoPimpleFoam	1
	多相	interFoam	2
		multiphaseEulerFoam	1
	可压缩	buoyantPimpleFoam	1
	基础	laplacianFoam	1

3.1.2 评估指标

多智能体系统的性能通过四项指标进行评估：迭代次数、令牌用量、通过率和成本。具体而言，迭代次数是多智能体系统成功完成一个仿真案例所需错误修正次数的平均值，用于评估智能纠错系统的错误修正能力。令牌用量是平均令牌开销，包括三部分：输入令牌、思考令牌和输出令牌。通过率表示成功完成的智能体仿真测试占总测试数的比例，反映系统完成仿真任务的能力。成本是完成一个案例的平均开销；我们根据文献中的公式进行计算(Gebreab等，2026)。成本与令牌用量成正比，并与大语言模型的定价模型有关，反映了系统的经济效率。四项指标的具体计算方法如下：

$$\text{Iterations} = \frac{\sum_{i=1}^{n} k_i + m \bullet k_{max}}{m+n} \tag{1}$$

$$\text{Token usage} = \frac{\sum_{i=1}^{m+n} (T_{in,i} + T_{think,i} + T_{out,i})}{m+n} \tag{2}$$

$$\text{Pass rate} = \frac{n}{m+n} \tag{3}$$

$$\text{Cost} = \frac{\sum_{i=1}^{m+n} (T_{in,i} \bullet P_{in} + T_{think,i} \bullet P_{think} + T_{out,i} \bullet P_{out})}{10000(m+n)} \tag{4}$$

其中，k_i 是第 i 个成功运行案例的迭代次数；m 和 n 分别是失败测试和成功测试的数量；k_{max} 是允许的最大迭代次数，在实验中设为 20；$T_{in,\ i}$ 是第 i 个案例的输入令牌成本；$T_{think,\ i}$ 是第 i 个案例的思考令牌成本；$T_{out,\ i}$ 是第 i 个案例的输出令牌成本；P_{in} 是大语言模型输入令牌的单价；P_{think} 是大语言模型思考令牌的单价；P_{out} 是大语言模型输出令牌的单价。

3.1.3 模型设置

在统一测试中，SwarmFoam 使用 Gemini-2.5-flash 执行文本生成和视觉理解任务。构建本地知识库时，选用 HuggingfaceEmbeddings 作为嵌入模型。SwarmFoam 使用的 OpenFOAM 版本为 OpenFOAM-9。由于 OpenFOAM 对格式有严格要求，因此将大语言模型的温度参数设为 0.01（值越低，随机性越小）。

在针对自然语言输入案例的对比测试中，SwarmFoam 和 Foam-Agent（基线）均使用 DeepSeek-Chat 作为大语言模型，因为 Foam-Agent 不支持 Gemini系列模型。

3.2 结果与讨论

3.2.1 总体性能

SwarmFoam 在数据集中的 25 个仿真案例上进行了评估，结果见 图8。所有案例的总体通过率为 84%（使用 Gemini-2.5-flash）。其中，自然语言输入案例的通过率为 80%（Gemini-2.5-flash），多模态输入案例的通过率为 86.67%（Gemini-2.5-flash）。由于处理多模态输入需要额外进行图像解析，多模态案例的令牌消耗和迭代次数明显高于自然语言案例。这些结果表明，*SwarmFoam* 对不同模态的仿真输入表现出出色的适应性。

由于 Foam-Agent 仅接受自然语言输入，因此使用数据集中的 10 个自然语言案例进行了对比测试（两个系统均使用 DeepSeek-Chat）。尽管与 Foam-Agent 相比，SwarmFoam 的通过率没有显著差异，但其平均令牌消耗大幅降低了 83.85%。这归因于“首错

”校正机制，该机制在第2.3.5节中引入。在每次迭代中，*SwarmFoam*会识别并修正最有可能导致失败的单个文件。虽然这可能会增加平均迭代次数，但能够有效减少校正阶段被重写的文件数量，尤其是在处理复杂任务时。

对使用不同大语言模型处理自然语言案例时的智能体仿真性能进行比较后发现，Gemini-2.5-flash（具备推理机制）在平均迭代次数（减少 50.79%）和仿真通过率（+30 %）方面显著优于 DeepSeek-Chat（不具备推理机制）。由于增加了推理过程，前者的令牌用量高于后者（+20.86%）。总体而言，这些发现表明，具备推理机制的大语言模型更适合处理 CFD智能体仿真任务。

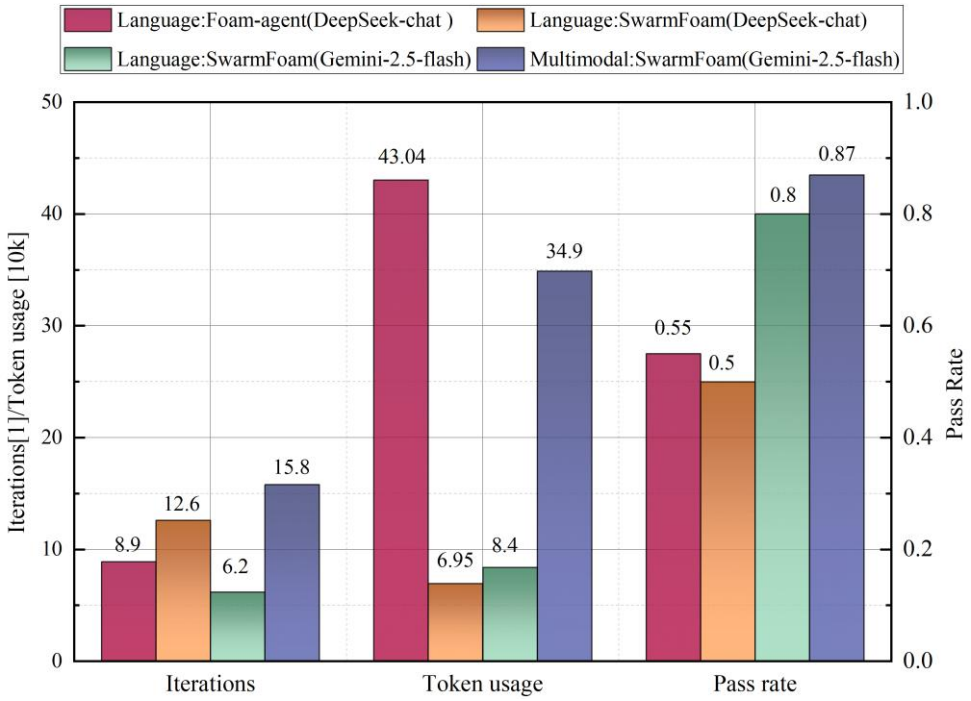


图8. 整体仿真性能比较。

3.2.2 案例验证

为验证*SwarmFoam*智能体仿真的准确性，选取了三个典型 CFD案例进行讨论：Cavity、DamBreak 和 T形交叉管流量分布。这三个案例的自然语言输入如表4所示。特别是，空腔算例采用了两种不同的几何形状和求解器（icoFoam 和 pisoFoam），以展示 *SwarmFoam* 的适应性。

图9 和 图10 展示了空腔和截切空腔在 t=0.5s 时的图像输入和流场分布。可以观察到，对于完整空腔，由于上盖运动，整个空腔内形成了明显的循环流。对于 CavityClipped 算例，由于左下角缺失，循环流主要集中在空腔上部。 图11展示了 DamBreak 算例的图像输入，以及 t = 9s 时的水相分数分布。仿真结果显示，水柱坍塌后撞击底部障碍物和罐壁，形成复杂的流动

结构。 图12 展示了 T形交叉管流量分布算例的图像输入，以及 t = 100s 时的速度场分布。经过一段时间后，管道内的流速达到稳定状态。主流集中在初始速度方向上，较小部分的流体则从右侧出口流出。

这三个算例的测试结果表明，SwarmFoam 的仿真结果与实际结果具有良好的一致性。SwarmFoam 准确理解了用户需求和算例图像，正确构建了几何体和网格，并成功复现了各种物理现象。

表4 自然语言输入

算例名称	仿真输入
Cavity	使用 icoFoam 对不可压缩空腔流进行二维 RANS模拟
CavityClipped	使用 pisoFoam 软件进行二维 RANS 仿真，模拟不可压缩的 3/4 立方腔体流动。所用的 RANS 模型为 Kepsilon，速度为每秒 1 米。
DamBreak	使用 interFoam 求解简化的二维溃坝问题。在水箱底部有一个小障碍物。在 t = 0s 时，水柱允许其自由流动，随后将发生水柱坍塌。
T形接头	使用 simpleFoam 求解器计算内部的二维流动分布问题。管道内的流动介质为空气，其密度为 1 kg/m3，动力黏度为 0.003333 kg/m-s。

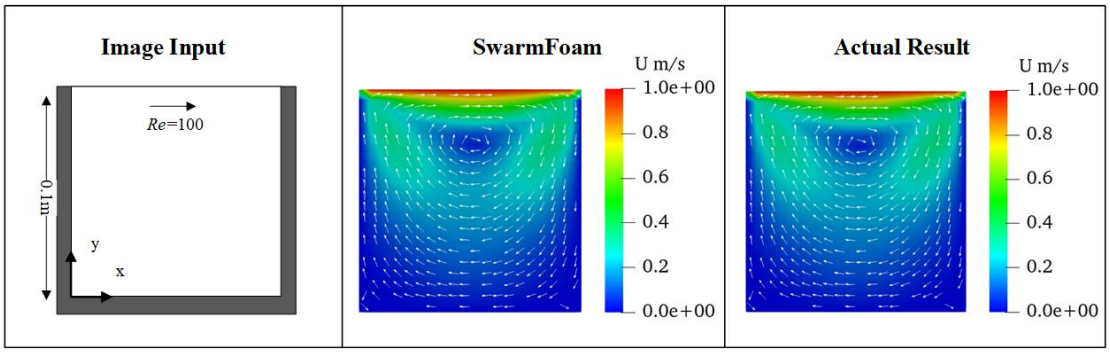


图 9. 空腔算例（icoFoam）

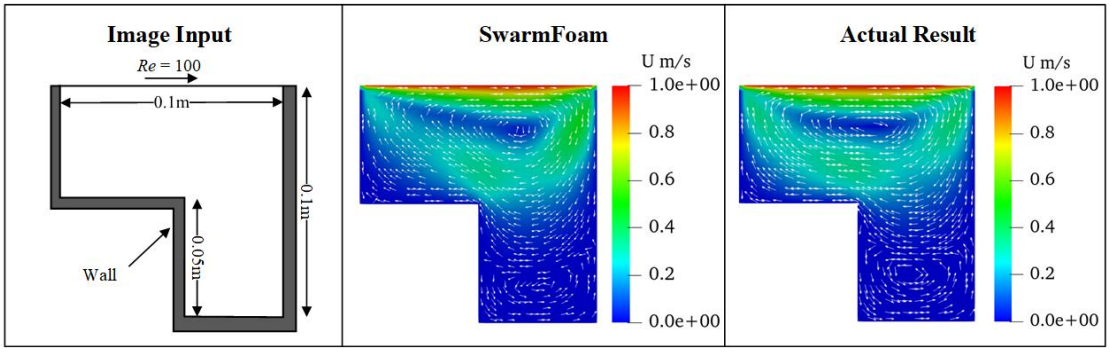


图 10. 截切空腔算例（pisoFoam）

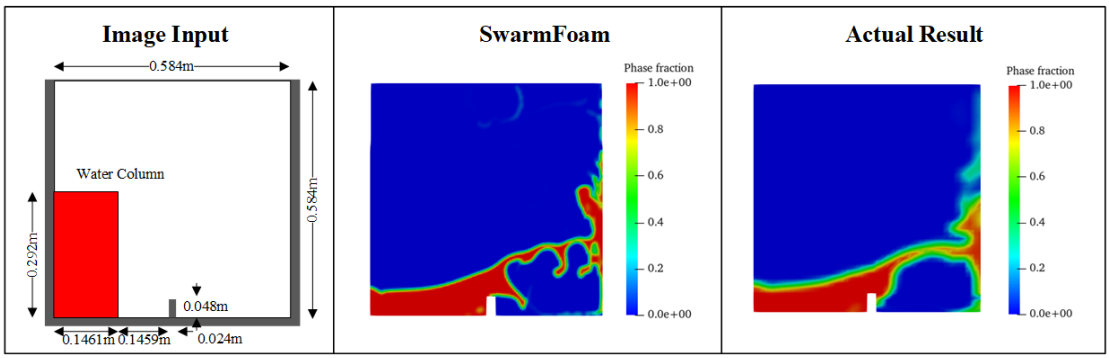


图11. 溃坝算例 (interFoam)

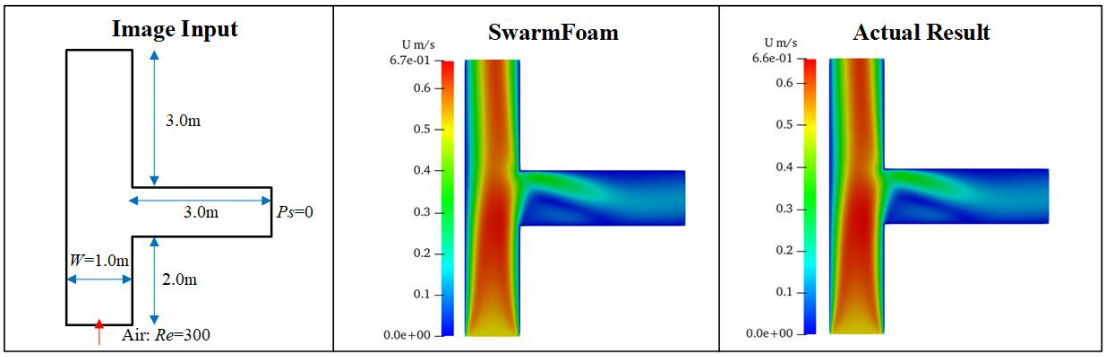


图12. T形结算例 (simpleFoam)

3.2.3 消融研究

消融研究的目的是（Gong等，2024）通过移除或替换特定的关键设计，评估其对*SwarmFoam*整体仿真性能的贡献。在本小节中，我们针对多模态感知机制和智能纠错机制开展消融研究。

1)不使用多模态感知机制

多模态感知机制的消融研究结果如图13所示。方法1采用图像解析与网格文件生成分别进行（即解耦）的方式（使用 ObservePicture）。方法2则采用图像解析与网格文件生成同步进行的方式（不使用 ObservePicture）。

由于方法2不会预先解析图像，因此其提示词更短。观察智能体仅执行DivideTask动作，而跳过ObserverPicture动作，因此与方法1相比，该阶段的大语言模型调用次数更少。从理论上说，方法2应当更经济。然而，实际测试结果表明，方法2在网格构建过程中丢失了大量几何与物理信息。这导致其平均迭代次数（+150.79%）显著高于方法1，而且大多数任务在达到最大纠错上限（20次迭代）时被强制终止。如图13所示，方法2较低的通过率（-54.02%）反而导致其令牌用量和平均成本接近翻倍。这些发现表明，对于多模态算例，ObserverPicture动作在提升 *SwarmFoam* 的基于智能体的仿真性能方面发挥着关键作用。

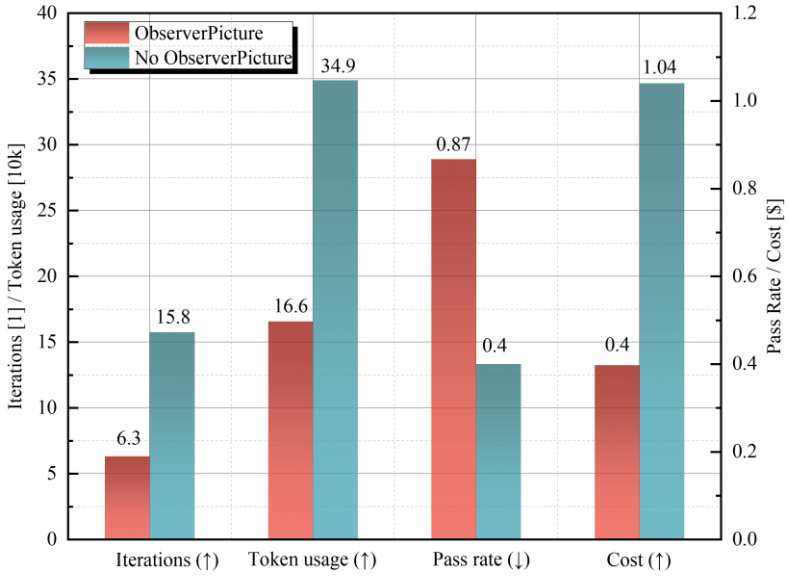


图 13. 使用不同多模态信息解析方法时的仿真性能比较。

2)无错误纠正机制

智能纠错机制对仿真性能的影响如图14所示。禁用审查者智能体后，系统不再进行多轮迭代，因此令牌用量降低（-30.95%），大语言模型调用成本也有所减少（- 40.63%）。然而，这也使通过率降低了60%。这表明，受模型性能和知识范围限制，大语言模型难以在首次尝试时完成复杂任务的智能体仿真，因此需要可靠的错误纠正机制。

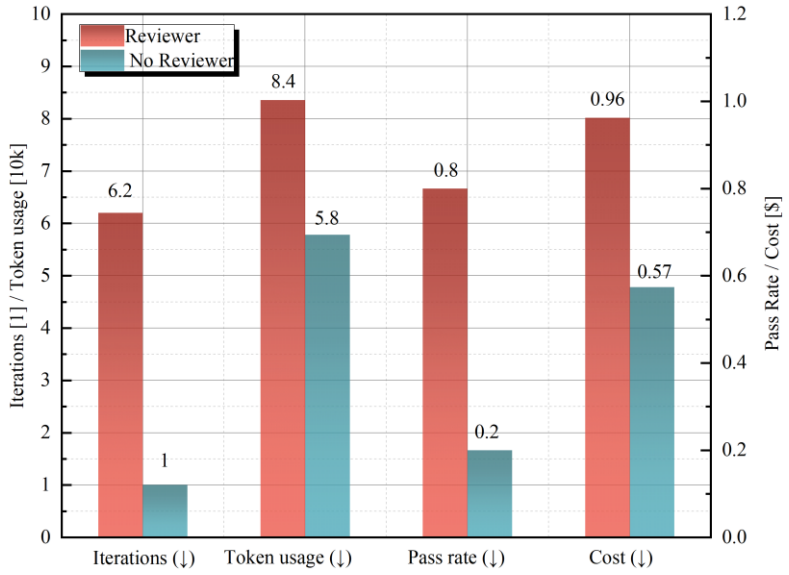


图 14. 错误纠正机制对仿真性能的影响。

3.2.4 错误分析

如第3.2.3节所述，与人类专家类似，大语言模型必须经历一系列 f

迭代式错误纠正过程，才能完成仿真任务。为便于今后对智能体系统进行有针对性的优化和升级，同时评估当前通用大语言模型在处理计算流体力学任务时的局限性，我们对SwarmFoam在算例迭代过程中捕获的仿真错误进行了统计分析。结果如图15所示。为便于讨论，仿真错误分为五类：

1)配置错误：求解器、初始条件、边界条件及物理属性相关文件的配置不正确。如图所示，配置错误是多模态输入案例（71.1%）和自然语言输入案例（44.9%）中最主要的错误类型。这类错误主要源于大语言模型（LLM）的能力局限，以及依赖文件所提供的信息不足，例如缺少湍流模型的关键参数设置，或物理量的量纲不匹配。

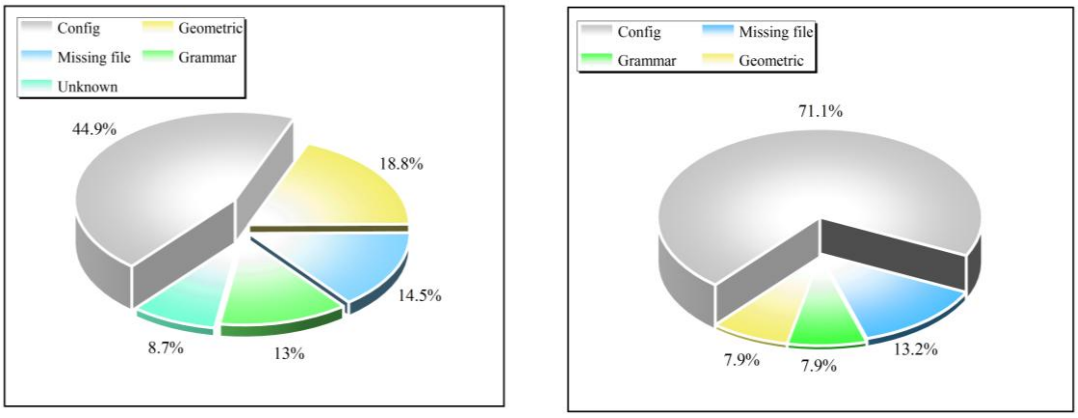
2)几何错误： 三维几何或网格生成中的错误，导致无法通过网格质量检查。由于多模态输入案例涉及的几何形状更加复杂，几何错误的比例高出5.1%。OpenFOAM 的网格生成工具完全依靠顶点坐标的组合来定义几何条件并生成网格。仅依靠自然语言描述时，大语言模型难以生成可靠的“blockMeshDict”文件。

3)文件缺失： 完成仿真所必需的文件缺失。此错误主要由两个原因导致。最主要的原因是架构师智能体规划的算例文件结构不当。第二个原因是未能有效清理大语言模型输出中的残留内容，导致 OpenFOAM 程序将某些文件识别为空文件。

4) 语法错误： 配置文件的语法不符合 OpenFOAM规范。这主要是由于大语言模型的输出遗漏了部分 OpenFOAM结构符号。

5)未知类型： 检查员智能体掌握的信息不足以进行错误修正，包括无法确定错误原因或路径。

通过整合检索增强、智能纠错和文件清理等机制，SwarmFoam成功解决了上述大多数仿真错误。



(a)多模态输入算例　　(b)自然语言输入算例

图 15. 仿真错误的统计分析。

4. 结论

计算流体力学是研究人员理解物理现象和评估设备性能的重要手段，涉及几何构建、网格生成和模型配置等复杂流程。由于这些特点，计算流体力学任务长期以来依赖经过专业培训、在流体力学和计算机科学方面具备丰富知识的人类专家。在本研究中，仿真完全由多个智能代理协作完成。仿真需求以图像或高级指令的形式输入，并由多模态感知机制进行解析。基于对仿真任务的理解，智能体生成的仿真文件质量通过检索本地专家知识的检索增强生成系统得到提升。为评估所设计多智能体系统的综合性能，我们使用涵盖多模态输入并涉及多种物理问题的仿真数据集对代理系统进行了测试，其中包括单相流、多相流和磁流体力学。本文得出的主要结论如下：

1）基于多类型大语言模型的计算流体力学多智能体框架（SwarmFoam）能够有效提取案例图像信息，并准确理解用户的仿真需求，成功实现智能CFD代理。通过结合检索增强生成、多模态感知和智能错误校正算法，SwarmFoam 在 25 个独立测试案例中的总体通过率达到 84%。

2）在生成配置文件之前预解析图像，以提取几何体和物理信息，对于提升仿真性能具有重要意义。执行图像预解析操作（ObserverPicture）后，多模态算例的通过率提高了 54.02%，成本降低了 61.54%。3）配置错误是迭代次数增加、成本上升的主要原因。在多模态和自然语言案例测试中，配置错误分别占错误总数的 71.1% 和 44.9%。这凸显了计算流体力学多智能体代理系统具备稳健的依赖文件生成机制和检索增强生成机制的必要性。

5. 未来工作

SwarmFoam 的研究展示了多智能体系统和多模态大语言模型在实现智能仿真方面的巨大潜力，其方法也可扩展到数值天气预报、计算结构力学和计算生物力学等其他科学计算领域。为进一步推动计算流体力学智能体智能的发展，并改进系统，我们提出以下建议：

• 改进网格生成机制。通用大语言模型在构建复杂几何结构和生成网格时仍不够可靠，尤其难以处理曲面边界和非结构化网格。这一不足可能源于大语言模型对计算流体力学原理和 OpenFOAM特定约定缺乏足够的专业知识。• 增强多模态感知能力。应提升多模态感知系统的性能，使其能够解析实际工程图纸。这可能需要设计更复杂的多模态解析策略，并训练专用的视觉理解模型。• 改进智能体对仿真结果质量的评估能力。这需要探索一套仿真验证标准来评估结果的可靠性，而不只是

根据日志错误判断是否需要修正配置。

致谢

本研究得到中国核工业集团有限公司引领创新基金（项目编号：CNNC-LCKY-2023-047）、四川省科技计划（项目编号：2025ZNSFSC0082）、中国核动力研究设计院先进核能技术国家重点实验室（项目编号：YNSW-0224-0202-04-01）以及中国核工业集团有限公司（CNNC）“青年人才计划”项目的资助。该项目题为“医用同位素试验堆燃料溶液中的气泡输运机制与两相瞬态现象研究”。

附录

A.1 提示模板

所有智能体均使用标准化的提示模板。每个智能体会根据其任务的具体要求对该模板进行一定程度的修改。下图展示了 SwarmFoam 的部分提示词。

ObserverPicture Prompt

<Your task>
Based on the description of the simulation object and the pictures, analyze the geometric and physical information.

<task description>
{cfd_example_describe}

<Reference information>
The relevant content for completing the current task

<Output requirement>
Return ```The information of the example picture is as follows:
Geometric description: This section contains three pieces of information. The first part is the geometric description. The second part is the vertex coordinates. The third part contains the coordinate information of edges or faces.
Physical description: This part includes the physical information in the analyzed picture, such as flow velocity, flow direction, temperature, pressure, etc.```
if no picture, both return "None"

Keep "Geometric description:" and "Physical description:" in your return.
Do not output any other content or identifier.
Do not include special symbols in the reply.
Do not have any other identifiers in the begin and end of the output file.

DivideTasks Prompt

<Your task>
According to the cfd_example_describe in the task description, the user requirements need to be decomposed into simulation tasks and post-processing tasks.

<task description>
{cfd_example_describe}

<Reference information>
The relevant content for completing the current task

<Output requirement>
Return The tasks is as follows:```
simulation tasks: The actual simulation requirements you have obtained.
post-processing tasks: The post-processing requirements in the description.If the description does not require the post-processing task，content after "post-processing tasks:" to be "None".```

Do not output any other content or identifier.
Do not include special symbols in the reply.
Do not have any other identifiers in the begin and end of the output file.

HandleError Prompt

<Your task>
Based on the error message from OpenFOAM, determine the type of error (note) error type must be one of the follows:format error, Missing file. The "Missing file" function only considers the files within the "0", "system" and "constant" folders, as well as the "Allrun" file.

<error message>
{errors}

<Existing files>
{file_list}

<Error type explain>
Format error: Some configuration files do not comply with the OpenFOAM standards (such as having incorrect formats) or are improperly configured, or missing file keyword.
Missing error: A certain file is missing in the example directory, which don't in the existing files.

<Output requirement>
Return error type. Do not output any other content.

WriteFoamFile Prompt

<Your task>

For input requirement in user requirement, If there are similar input files, reference the similar examples (for reference only) to create the input files.

If there is no reference, complete the creation directly based on the description for OpenFOAM-9.

Simulation task information is in task description. The physical information of the simulation task is in Physical information.

The geometric information of the simulation object is in Geometrical information or base the picture.

The dimension of physical quantities are consistent with those in the "similar file".

If the reference example uses the same solver as the current one, the dimensions should be consistent.

The current version of OpenFOAM is 9.

<User requirement>

{requirement}

<task description>

{CFD_task}

<Physical information>

{physical_information}

<Geometrical information>

{geometrical_information}

<solver help>

Some dimension or naming suggestions

{solver}

<similar file>

for reference only

{similar_file}

<dependent file>

The "dependent file" refers to the files that are dependencies of the target file.

Ensure that the boundary settings of the current file and the dependent files are consistent, and that the parameter settings are also matched.

{associated_file}

<Output requirement>

Return ```The input file is:The content you have written```

Do not output any other content or identifier.

Do not include special symbols in the reply.

Do not have any other identifiers in the begin and end of the output file.

参考文献

Chen, J., & Bao, Y.（2025年）。用于钢筋混凝土结构符合规范的自动化设计的多智能体大语言模型框架。《Automation in Construction》，177，106331。https://doi.org/https://doi.org/10.1016/j.autcon.2025.106331

Chen, Y., Zhu, X., Zhou, H., & Ren, Z.（2024年）。MetaOpenFOAM：一种用于计算流体力学的基于大语言模型的多智能体框架。https://doi.org/10.48550/arXiv.2407.21320Chen, Y., Zhu, X., Zhou, H., & Ren, Z.（2025年）。MetaOpenFOAM 2.0：用于自动化计算流体力学模拟与后处理的大语言模型驱动的思维链。arXiv，abs/2502.00498。Dong, Y., Zhan, Z., Hu, Y., Doe, D. M., & Han, Z.（2025年，2025/12/01）。利用大语言模型驱动的多智能体系统，面向非专业人士在建筑设计中的交互的人工智能建筑信息模型协调员。《Automation in Construction》，180，106563。https://doi.org/https://doi.org/10.1016/j.autcon.2025.106563Feng, J., Xu, R., & Chu, X.（2025年）。OpenFOAMGPT 2.0：面向计算流体力学的端到端可信自动化。

Ferro, C. G., Pietrangelo, F., & Maggiore, P.（2023年，2023年/10/01/）。采用计算流体力学分析评估创新航空航天防冰系统格栅面板内部的换热性能。《Aerospace Science and Technology》，141，108565。https://doi.org/https://doi.org/10.1016/j.ast.2023.108565

Gbadago, D. Q., Go, S., & Hwang, S.（2025年，2025年/01/01/）。化学过程设计的重大进展：推出集成式人工智能与计算流体力学模拟自动化框架。《Computers & Chemical Engineering》，192，108906。https://doi.org/https://doi.org/10.1016/j.compchemeng.2024.108906

Gebreab, S., Musamih, A., Salah, K., & Jayaraman, R.（2026年，2026年/03/01/）。基于大语言模型的实时与历史区块链数据探索与分析。《应用专家系统》，298，129851。https://doi.org/https://doi.org/10.1016/j.eswa.2025.129851

Gong, H., Xing, H., Yu, Y., & Liang, Y.（2024年，2024年10月15日）。一种用于原油期货价格和波动率预测的二次分解组合模型：基于比较实验和消融实验的分析。《应用专家系统》, 252, 124196.https://doi.org/https://doi.org/10.1016/j.eswa.2024.124196

Hong, S., Zheng, X., Chen, J. P., Cheng, Y., Zhang, C., Wang, Z., Yau, S. K. S., Lin, Z. H., Zhou, L., Ran, C., Xiao, L., & Wu, C.（2023年）。MetaGPT：面向多智能体协作框架的元编程。arXiv，abs/2308.00352。

Jansasithorn, A., Ma, L., Ingham, D., & Pourkashanian, M.（2025年，2025年10月30日）。使用田口方法构建稳健的计算流体力学模型，以仿真垂直轴风力涡轮机的动态被动自启动。《Energy》, 335, 137965.https://doi.org/https://doi.org/10.1016/j.energy.2025.137965

Jeong, S., Baek, J., Cho, S., Hwang, S. J., & Park, J.（2024年，六月）。Adaptive-RAG：通过问题复杂度学习适应检索增强大语言模型。载于 K。

Duh, H. Gomez, & S. Bethard, 《2024年计算语言学协会北美分会年会论文集：人类语言技术》（第 1 卷：长篇论文），墨西哥城，墨西哥。

Kim, I., Chae, J., & You, D.（2025年，2025/11/05/）。利用深度强化学习为叶片通道的最优计算流体力学生成自动网格。《计算物理学杂志》，541，114306。https://doi.org/https://doi.org/10.1016/j.jcp.2025.114306

Liu, B., Nie, Y., Song, H., Lin, Y., Li, J., Weng, X., Su, Z., Suo, Y., Lv, T., Zhao, X., & Yang, J.（2026年，2026/02/01/）。Magic：一种基于大语言模型的多智能体激活式图推理智能协作模型，用于肝病诊断。《信息融合》，126，103557。https://doi.org/https://doi.org/10.1016/j.inffus.2025.103557

Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E.（2021，2021/03/01）。基于算子通用逼近定理，通过 DeepONet 学习非线性算子。《自然·机器智能》，3（3），218-229。https://doi.org/10.1038/s42256-021-00302-5

Luo, S., Xu, J., Zhang, X., Wang, L., Liu, S., Hou, H., & Song, L.（2026年，2026年/02/01/）。RALLRec+：基于推理的检索增强型大语言模型推荐。《应用专家系统》，297，129508。https://doi.org/https://doi.org/10.1016/j.eswa.2025.129508

Lusher, D. J., Sansica, A., Sandham, N. D., Meng, J., Siklósi, B., & Hashimoto, A.（2025年，2025年/02/01/）。OpenSBLI v3.0：利用图形处理器上的领域专用语言进行高保真多块跨声速翼型计算流体力学模拟。《计算物理通讯》，307，109406。https://doi.org/https://doi.org/10.1016/j.cpc.2024.109406

Mao, R., Lin, M., Zhang, Y., Zhang, T., Xu, Z.-Q. J., & Chen, Z. X.（2023年，2023年/10/01/）。DeepFlame：由深度学习赋能的反应流模拟开源平台。《计算物理通讯》，291，108842。https://doi.org/https://doi.org/10.1016/j.cpc.2023.108842

Novati, G., de Laroussilhe, H. L., & Koumoutsakos, P. (2021, 2021/01/01). 通过多智能体强化学习实现湍流建模自动化。《自然·机器智能》, 3(1), 87-96.https://doi.org/10.1038/s42256-020-00272-0

Pandey, S., Xu, R., Wang, W., & Chu, X. (2025年). OpenFOAMGPT：一种用于基于 OpenFOAM 的计算流体力学的大语言模型（LLM）智能体，采用检索增强技术。《流体物理》, 37(3), 035120. https://doi.org/10.1063/5.0257555

Qiu, J., Lam, K., Li, G., Acharya, A., Wong, T. Y., Darzi, A., Yuan, W., & Topol, E. J. (2024年, 2024/12/01). 医学与医疗保健领域基于大语言模型的智能体系统。《自然 · 机器智能》，6(12), 1418-1420. https://doi.org/10.1038/s42256-024-00944-1

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019, 2019/02/01/). 物理信息神经网络：用于求解涉及非线性偏微分方程的正问题和逆问题的深度学习框架。《计算物理学杂志》, 378, 686-707.https://doi.org/https://doi.org/10.1016/j.jcp.2018.10.045

Suárez-V ázquez, M., Ballesta, S. V., Otero-Cacho, A., Muñuzuri, A. P. 和 Mira, J.（2025年，

2025年/12/01/）。城市计算流体力学模拟的一种快速自动化方法：利用气象预测进行验证及其在无人机飞行中的应用。https://doi.org/https://doi.org/10.1016/j.uclim.2025.102664Wang, J., & Duan, Z. (2024). 借助 LangGraph 实现智能体人工智能：利用大语言模型提升机器翻译的模块化框架。arXiv, abs/2412.03801。Xie, J., Feng, X., Gao, T., Dong, C., Yin, B., & Wu, C. (2025, 2025/10/01/). 使用 CUDA C 加速隐式海洋模型。《应用海洋研究》, 163, 104740。https://doi.org/https://doi.org/10.1016/j.apor.2025.104740Xin, J., Wang, X., Malik, A., Tian, Z., & Lu, H. (2025). 上端壁边界层抽吸对钍熔盐堆中高负荷氦压缩机性能的影响。《流体物理》, 37(9)。 https://doi.org/10.1063/5.0282786Yue, L., Somasekharan, N., Cao, Y., & Pan, S. (2025). Foam-Agent：迈向自动化智能计算流体力学工作流。arXiv, abs/2505.04997。Yue, L., Somasekharan, N., Zhang, T., Cao, Y., & Pan, S. (2025). Foam-Agent：用于自动化 OpenFOAM 计算流体力学模拟的端到端可组合多智能体框架。arXiv 预印本 arXiv:2509.18178。Zhang, M., Fang, Z., Wang, T., Lu, S., Wang, X., & Shi, T. (2025, 2025/07/05/). CCMA：利用大语言模型实现自动驾驶汇入场景中级联协作多智能体的框架。《应用专家系统》, 282, 127717。https://doi.org/https://doi.org/10.1016/j.eswa.2025.127717Zhang, P., Lin, K., Li, D., Fu, Z., Cai, Y., Li, B., Yu, H., & Li, M. (2025, 2025/11/01/). DAPlanner：结合多模态大语言模型的自动驾驶运动规划双智能体框架。《应用软计算》, 183, 113625。https://doi.org/https://doi.org/10.1016/j.asoc.2025.113625