Theoretical and Applied Mechanics Letters 15 (2025) 100623

Contents lists available at ScienceDirect

Theoretical and Applied Mechanics Letters

journal homepage: www.elsevier.com/locate/taml

Research Article

A status quo investigation of large-language models for cost-effective computational fluid dynamics automation with OpenFOAMGPT

Wenkang Wang[a,1], Ran Xu[b,1], Jingsen Feng[c], Qingfu Zhang[d], Sandeep Pandey[e], Xu Chu[c,b,*]

[a] International Research Institute for Multidisciplinary Science, Beihang University, Beijing 100191, China
[b] Faculty for Aerospace Engineering and Geodesy, University of Stuttgart, 70569 Stuttgart, Germany
[c] Department of Engineering, University of Exeter, UK
[d] Institute of Fluid Mechanics, Beihang University, Beijing 100191, China
[e] Institute of Thermodynamics and Fluid Mechanics, Technische Universit¨at Ilmenau, Ilmenau D-98684, Germany

ARTICLE INFO

Keywords:
CFD
OpenFOAM
LLM
OpenFOAMGPT

ABSTRACT

We evaluated the performance of OpenFOAMGPT (GPT for generative pretrained transformers), which includes rating multiple large-language models. Some of the present models efficiently manage different computational fluid dynamics (CFD) tasks, such as adjusting boundary conditions, turbulence models, and solver configurations, although their token cost and stability vary. Locally deployed smaller models such as the QwQ-32B (Q4 KM quantized model) struggled with generating valid solver files for complex processes. Zero-shot prompts commonly fail in simulations with intricate settings, even for large models. Challenges with boundary conditions and solver keywords stress the need for expert supervision, indicating that further development is needed to fully automate specialized CFD simulations.

1. Introduction

In recent years, the fluid mechanics community has rapidly embraced data-driven strategies, propelled by the proliferation of high-fidelity simulation data and the remarkable progress in machine learning techniques [1–6]. These methods have shown promise in turbulence modeling, both with and without governing equations [7–12], as well as in solving intricate heat transfer problems [13,14]. Machine learning has also enhanced various experimental fluid mechanics tasks [15] and facilitated scientific discovery, for example, by leveraging causal inference techniques to interpret fluid flow phenomena [16–18].

Simultaneously, large language models (LLMs), such as ChatGPT [19] (GPT for generative pretrained transformers), DeepSeek [20], and Qwen [21], have surged to the forefront of scientific and engineering research, offering unparalleled capabilities in natural language processing [22], automated reasoning [23], and high-level decision-making. Their potential to streamline research pipelines is reflected in diverse applications, from problem-solving and optimization [24–26] to faster scientific breakthroughs [27,28]. Notably, these models can serve as intelligent assistants that both amplify traditional methodologies and introduce novel strategies for data analysis, design, and simulation. These advancements are further complemented by work in multimodal vision-language models, exemplified by Cephalo [29], which merges visual and linguistic data to support complex material design tasks.

Within fluid mechanics, LLMs have proven useful for equation discovery [30] and for streamlining shape optimization [31], demonstrating efficiency in tasks such as identifying governing equations and refining geometric profiles (e.g., airfoils). Recent research has extended LLM-driven insights to microfluidics [32], where LLMs facilitate decision-making for robotic motion planning, as well as unsteady flow prediction [33], combining pretrained transformers and graph neural networks for improved spatial-temporal accuracy. In related engineering fields, Kim et al. [34] explored ChatGPT for automated MATLAB code generation, proving its utility in providing structured starter codes and logic in the expert direction. Similarly, Chen et al. [35] proposed a multiagent LLM system for orchestrating CFD workflows via natural language, making these techniques more accessible through retrieval-augmented generation (RAG). Recently, Chen et al. [36] developed MetaOpenFOAM 2.0, which leverages chain of thought (COT) decomposition and iterative verification to perform complex CFD tasks via natural language, achieving higher executability and cost efficiency. Dong et al. [37] used a domain-adapted LLM that translates natural language into executable CFD configurations via a multiagent system, achieving 88.7% solution accuracy and outperforming larger general-purpose models.

* Corresponding author.
E-mail address: x.chu@exeter.ac.uk (X. Chu).
[1] These authors contributed equally to this work

https://doi.org/10.1016/j.taml.2025.100623
Received 26 May 2025; Received in revised form 18 September 2025; Accepted 19 September 2025
Available online 20 September 2025

Building on this momentum, our recent work introduced OpenFOAMGPT [38], an LLM-based agent for OpenFOAM-centric CFD simulations that integrates GPT-4o and a chain-of-thought-enabled o1 model. While the o1 model has a higher token cost—approximately six times greater than that of GPT-4o—it consistently outperforms the other methods in tasks ranging from zero-shot case setup and boundary condition refinements to turbulence model adjustments and code translation. Through iterative correction loops and a RAG for domain-specific knowledge, the framework adeptly handles a range of flow configurations (including single- and multiphase flow) in just a few iterations, although human oversight remains vital for accuracy and flexibility. This capability not only broadens the applicability of CFD methods to non-specialists, but also signals promising opportunities for adopting LLM-based CFD agents in other solver environments. However, the OpenAI o1 model incurs a relatively high token cost—$15 per million input tokens and $60 per million output tokens—which can become substantial when complex setups requiring iterative corrections are generated. Consequently, identifying a more cost-effective solution with comparable performance is highly beneficial.

In this work, we adapt OpenFOAMGPT to a wider range of LLMs to achieve a more economical agent framework for computational fluid dynamics simulations that leverages two new foundation models—DeepSeek V3 and Qwen 2.5-Max—to achieve cost reductions of up to $O(10^2)$. Our updated approach seeks to further streamline complex CFD workflows without the help of the RAG and scale seamlessly across diverse engineering applications. This approach evaluates the pure zero-shot performance of the OpenFOAMGPT.

2. Methodology

2.1. OpenFOAMGPT

Open-source field operation and manipulation (OpenFOAM) is a widely utilized, open-source CFD solver package [39–41]. Unlike commercial alternatives, OpenFOAM's open architecture enables users to modify existing solvers and develop new physical models that are very convenient to combine with LLM. Its C++ object-oriented design ensures maintainability, extensibility, and parallel efficiency, making it valuable for both academic research and industrial applications. These characteristics position OpenFOAM as an effective platform for integration with LLM-based agents that can guide users through simulation workflows. This study employs the OpenFOAM-v2406 release.

Figure 1 illustrates the hierarchical architecture of our agent OpenFOAMGPT. The workflow begins when a system prompt combines with a user query at the top level. The Builder module then interprets these instructions, consulting the RAG database for domain-specific knowledge when needed, and converts them into a structured execution plan. The executor subsequently manages the workflow by either directing queries to the LLM model for additional reasoning or delegating tasks to the interpreter that translates the output to files needed for simulation operations. The OpenFOAM runner then executes the simulation with the setup files, and the system output and error logs are continuously monitored during simulation; upon failure detection, the error data are appended to the original query and the process cycles again. Otherwise, the workflow terminates successfully. Unlike in our previous study, the RAG function is disabled for the present research to test the pure zero-shot capability.

The following 5 LLMs are currently considered and evaluated.

- ChatGPT-4o is a general-purpose multimodal LLM developed by OpenAI. Trained on a diverse range of internet text, it maintains advanced language understanding and generation capabilities across domains.
- OpenAI o1 is the first reasoning model leveraging a chain-of-thought (CoT) mechanism, enabling superior performance over GPT-4o in complex reasoning, scientific analyses, and programming tasks.
- DeepSeek V3 (671B) is the third-generation LLM from DeepSeek AI and is offered as an open-source alternative to high-end proprietary models. Deepseek V3 is under active update, and the version we tested is V3–0324.
- Qwen2.5-Max is Alibaba's latest LLM, designed as a general-purpose MoE AI system with great performance on various standard tests for LLM.
- Gemini 2.5 Pro is Google DeepMind's latest and most advanced AI model. It excels in multimodal reasoning, supporting text, image, audio, and video inputs.

The comparison of token pricing across LLMs reveals substantial economic differentials between U.S. and Chinese providers (Table 1). The OpenAI o1 model represents the highest-cost option at $15.0 and $60.0 per million tokens for input and output processing, respectively. In contrast, DeepSeek-V3 671B demonstrates remarkable cost efficiency at $0.035 and $0.55 per million tokens—approximately 10^2 less expensive than o1 for equivalent token processing. GPT-4o and Gemini 2.5 pro present an intermediate pricing tier among US-based models ($2.5/1.25 per million input tokens, $10.0 per million output tokens). Gemini 2.5 pro offers a massive 1 million token context window. The Qwen 2.5-Max model maintains competitive pricing at $0.80 and $1.2 per million tokens despite its more limited 32k context capacity. These pricing differentials substantiate our framework's approach of leveraging Chinese models to achieve the $O(10^2)$ cost reduction claimed in our computational fluid dynamics agent implementation while maintaining acceptable performance characteristics.

2.2. A test on locally deployed QwQ-32B on a personal desktop

The local deployment of large language models (LLMs) currently relies on three mainstream frameworks: SGLang [42], vLLM [43], and Ollama [44]. While SGLang and vLLM emphasize multi-GPU parallelization for industrial-scale inference, Ollama's lightweight architecture (v0.4.7) demonstrates superior suitability for single-GPU consumer hardware. We deployed the QwQ-32B Q4 KM quantized model (4-bit precision with medium granularity) on an NVIDIA RTX 4090 GPU (24 GB VRAM), achieving stable operation with 20.3 GB VRAM utilization. Integration into the OpenFOAMGPT workflow was implemented through local API calls (port 11434), enabling direct interaction with CFD simulation templates.

Testing encompassed two benchmark cases: the lid-driven cavity flow (laminar regime) and PitzDaily combustor flow (turbulent reactive case). Both scenarios were evaluated under zero-shot prompting and retrieval-augmented generation (RAG) conditions, with OpenFOAM v2406 documentation used as a supplementary context. Despite 20 maximum inference iterations per trial, neither configuration successfully generated valid solver files adhering to OpenFOAM syntax requirements. Notably, identical failure patterns emerged when the official non-quantized QwQ-32B was queried via the API, eliminating quantization artifacts as the primary cause of failure.

These results align with recent theoretical analyses of domain adaptation thresholds [45], where sub100B parameter general-purpose models exhibit critical knowledge gaps in specialized engineering domains. Error analysis revealed systematic failures in boundary condition specification and turbulence model parameterization [46]. These limitations suggest that effective deployment in computational mechanics workflows requires either scaling to foundation models or domain-specific fine-tuning of smaller architectures.

3. Evaluations

3.1. Zero-shot performance evaluation of different online models

Zero-shot prompting attempts to generate the desired output based on up to limited instructions without any example. As LLMs are trained

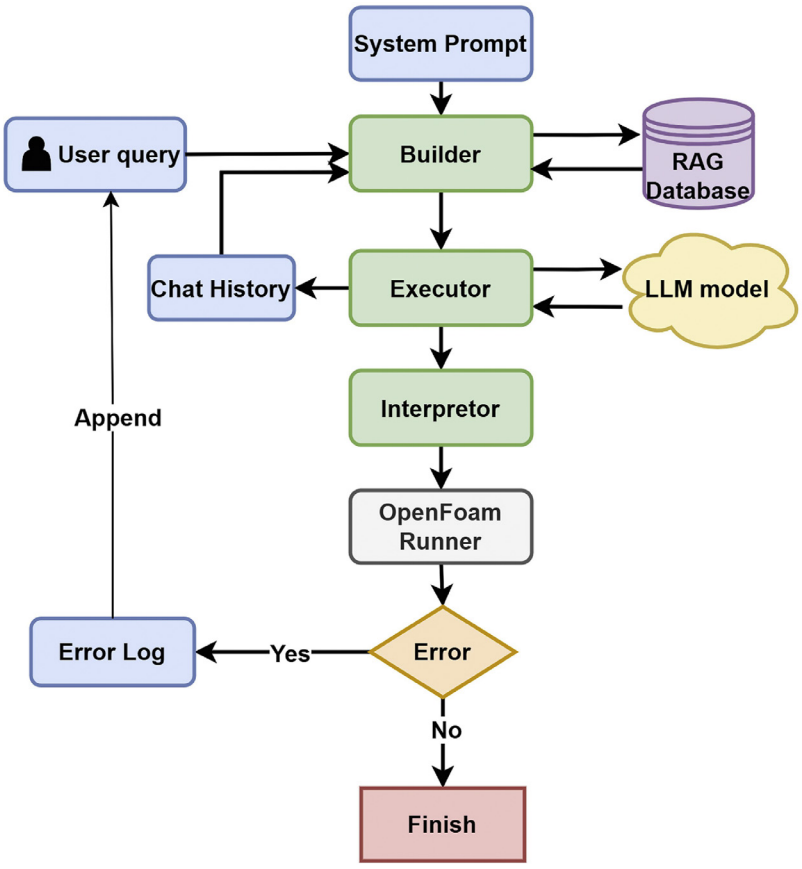

Fig. 1. Design of the agent structure.

Table 1
Comparison of the large language models used.

Model	Input price ($ per million tokens)	Output price ($ per million tokens)	Context length	Country
GPT-4o	2.5	10	128k	USA
OpenAI o1	15	60	200k	USA
DeepSeek-V3***	0.035	0.55	64k	China
Qwen 2.5-Max	0.8	1.2	32k	China
Gemini 2.5 pro	1.25	10	1M	USA

*Prices are approximate and based on public API rates (2025 Q2).
*Context window sizes may vary by implementation.
**Discount price (UTC 16:30–00:30)

Table 2
Promotion template used in the present work.

System Prompt: You are an expert in OpenFOAM, a computational fluid dynamics (CFD) software, with a focus on version v2406. Your task is to assist users in setting up and running OpenFOAM simulations by providing responses in the following strict format:

1. Under the heading **Directory Structure**, the complete directory structure required for the simulation is provided. [Detailed description of the directory structure]
2. Under the heading **File Contents**, provide **detailed, complete content** for each file within the directory structure. [Detailed instruction on the file contents]

User Input: Simulates incompressible, turbulent flow through the...[Discription of the flow setup]
In the last attempt, I encountered the following error. Please correct this error and provide the complete files again ...[Error log of the OpenFOAM runner]
This is your response from last conversation, use as a reference...[Answer generated from last step]

on vast datasets, they often result in the desired outputs. However, if the underlying problem is difficult, then the model might not provide the desired output. In this subsection, we analyze the performance of LLM models with zero-shot prompts.

The prompt template utilized in the present study is detailed in Table 2. This template encompasses several key components of the instructions. Specifically, the system promptly outlines the folder structure pertinent to the output and provides guidance on the contents of the files. The user prompt comprises a detailed account of the CFD configuration for the simulation. In instances where the OpenFOAM runner encounters an error, the error log is appended to the user prompt, together with the historical case files

Table 3
Tasks of alternating initial and boundary conditions. The results are labeled √ (successful), × (failed), and – (skipped). Tests for o1 were conducted only when GPT-4o failed. The numbers after √ are the minimum number of iterative steps needed.

Case	Alternate condition		4o	o1	Qwen	DS	Gemini
Cavity flow	Top wall velocity ($m{\cdot}s^{-1}$)	1 → 2	√ (2)	-	√ (2)	√ (2)	√ (2)
	Top wall velocity ($m{\cdot}s^{-1}$)	1 → 5 sin $2\pi \frac{t}{0.1}$	×	√ (3)	×	×	×
	Mesh resolution	20×20×1 → 15×15×1	√ (2)	-	√ (2)	×	√ (2)
	endTime	3 → 5	√ (2)	-	√ (2)	√ (2)	√ (2)
	Turbulence model	RNGkepsilon	√ (13)	-	×	×	√ (8)
	Turbulence model	kOmegaSST	√ (10)	-	√ (9)	×	√ (6)
	Turbulence model	kkLOmega	×	√ (7)	×	×	√ (6)
	Turbulence model	LRR	√ (7)	-	×	×	√ (6)
PitzDaily	Inlet velocity ($m{\cdot}s^{-1}$)	10 → 20	√ (3)	-	√ (4)	×	×
	Turbulence model	kOmegaSST	√ (2)	-	×	×	×
	Turbulence model	Smagorinsky (LES)	√ (5)	-	√ (6)	×	×
Hotroom	Hot wall temperature (K)	310 → 320	√ (9)	-	√ (8)	×	×
Dambreak	Liquid inside the membrane	water → oil	√ (4)	×	√ (5)	×	×
	Turbulence model	KEpsilon	√ (8)	-	×	×	×
Particle column	Velocity of the fluid/particles ($m{\cdot}s^{-1}$)	1 → 2	√ (3)	-	√ (11)	√ (8)	×
	Type of fluid	Air → CO	×	√ (10)	√ (11)	√ (10)	×
	Turbulence model	KEpsilon	×	×	×	×	×
Mixed vessel	Angular speed of rotation ($rad{\cdot}s^{-1}$)	20 → 15	√ (5)	-	×	×	×
	Turbulence Model	KEpsilon	×	√ (8)	×	×	×

from the preceding step, and the model is instructed to fix the error.

The evaluation encompasses a range of typical CFD engineering tasks, including modifying initial and boundary conditions, adjusting turbulence models, and updating thermophysical properties, as listed in Table 3. For each task, we performed more than five repeated tests. Note that for all the tests, we set the temperature parameter, which controls the randomness and creativity of the generated text, to 0. Therefore, the test outputs are nearly deterministic and replicable, ensuring the robustness and repeatability of the current results.

- Cavity flow: Simulates laminar, isothermal, incompressible flow in a square cavity via icoFoam. The top wall moves horizontally at 1 $m{\cdot}s^{-1}$; the other walls are stationary.
- PitzDaily: Models incompressible turbulent flow through a two-dimensional sudden expansion channel via the k-ϵ turbulence model and simpleFoam solver.
- Hotroom: Simulates turbulent natural convection in a tall rectangular cavity via the k-ϵ model and buoyantBoussinesqSimpleFoam. The bottom wall is heated, the top wall is cooled, and the side walls are adiabatic.
- Dambreak: Represents a simplified laminar dam break via the VOF-based interFoam solver. A water column collapses into a square tank containing a central rectangular obstacle, creating complex flow patterns and trapped air pockets.
- Particle column: MPPICFoam is used to simulate particle dynamics and fluid flow in a vertical rectangular column. Fluid motion is described by the Navier–Stokes equations, and particles are tracked via the Lagrangian approach, which considers drag, collisions, and gravity.
- Mixed vessel: This vessel simulates fluid mixing in a rotary agitator via pimpleFoam. The geometry features a cylindrical domain with rotating inner walls, stationary outer walls, and rectangular barriers to enhance mixing.

Table 3 shows that the OpenAI models have a better success rate than the other models do. However, Qwen2.5-Max delivers performance comparable to that of OpenAI models while dramatically reducing token costs. Owing to its high cost and strict token rate limit, GPT-o1 was tested solely on instances where GPT-4o was unsuccessful, as it is not an ideal option for tasks demanding multistep, large-scale text input and output.

Notably, the performance of DeepSeek V3 is largely affected by the unstable official API connection from DeepSeek, which makes it difficult to complete the iteration loop. This is also the main reason why Deepseek-R1 is not included in the current comparison. The reasoning models such as Deepseek-R1 have a lengthy reasoning process before producing the case file text, thus requiring a more stable API. However, the current official deepseek API faces frequent disconnection, leading to failed attempts that require many iterative steps.

Although third-party Deepseek APIs, such as those from the AI cloud service of Alibaba, Bytedance, Tencent, etc., are available, the performance of third-party models is evidently lower than that of official models. One of the notable errors shared by all the third-party models is that they often have garbage characters appearing in the generated case setup files, making it almost impossible to finish the tests. This likely results from the open-source model not being updated to match the version used by the official API. Therefore, the current test results are only based on the unstable deepseek API, which may not represent the full capability of deepseek models.

Gemini 2.5 Pro is the latest released among all the tested models. While it has a high success rate for the simple cavity flow case, it fails in the remaining cases, which involves more complicated physics. In many cases, Gemini 2.5 Pro understands the error during the iterative steps very well but cannot correctly modify the files accordingly.

3.2. Extensive zero-shot evaluations with Qwen 2.5-Max

We further evaluated the ability of Qwen 2.5-Max to generate and debug simulations, encompassing a series of classical single- and multiphase scenarios.

- 2D rising bubble (Fig. 2(a)): The setup consists of a rectangular tank filled with water, measuring 30 mm in width and 100 mm in height. Initially, a bubble with a diameter of 10 mm is positioned centrally at the bottom of the domain. Buoyancy-driven motion induces the bubble to rise, deform, and interact dynamically with the surrounding fluid.

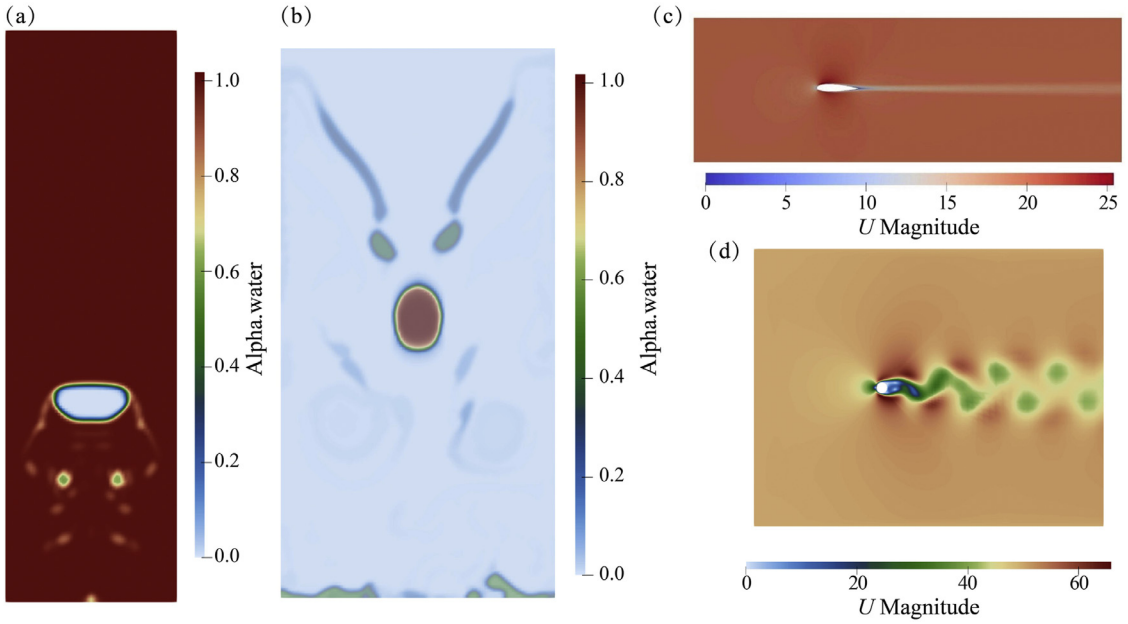

Fig. 2. Case simulation results. (a) 2D rising bubble. (b) 2D falling droplet. (c) AirFoil2D. (d) Cylinder.

Table 4
Evaluations with Qwen2.5-Max.

Case	File provided	Iterations	Result	Total tokens (k token)	Token cost
Bubble	blockMeshDict, setFieldsDict	8	√	71	$0.25
Droplet	blockMeshDict, setFieldsDict	20	√	195	$0.67
AirFoil	polyMesh	2	√	15	$0.056
MotorBike	polyMesh	10	Patch type 'patch' not constraint type 'empty'	66	$0.23
Cylinder	polyMesh	3	√	15	$0.05
Nozzle	blockMeshDict	20	'Smoother' not found in "fvSolution"	127	$0.37

- 2D falling droplet (Fig. 2(b)): This case examines the dynamics of a single water droplet falling under gravity within a two-dimensional rectangular tank filled with air, employing the volume of fluid (VoF) method. Initially, at $t = 0$ s, the droplet is positioned centrally at the tank's upper boundary and subsequently descends.
- 2D airfoil (Fig. 2(c)): This case investigates the aerodynamic performance of a two-dimensional NACA 0012 airfoil positioned at a 5° angle of attack within a computational wind tunnel. The domain dimensions are 2000 mm in length and 1000 mm in width. At the beginning of the simulation ($t = 0$ s), a uniform airflow at 20 m·s^{-1} enters the domain, initiating steady-state conditions driven by pressure gradients and viscous forces interacting with the airfoil surface.
- 3D MotorBike: This case investigates the aerodynamic and turbulent flow characteristics around a simplified motorBike geometry. The transient airflow interaction is modeled for a motorcycle body with overall dimensions of 2.1 m × 0.8 m × 1.2 m. The three-dimensional computational domain extends 20 vehicle lengths upstream and downstream, placing the motorcycle 10 m downstream from the inlet boundary.
- 2D cylinder (Fig. 2(d)): This case examines aerodynamic and vortex-induced phenomena around a two-dimensional circular cylinder. The cylinder is centered at the origin within a rectangular computational domain featuring clearly defined boundaries: the inlet (left side), outlet (right side), and walls (top and bottom). Initially, at $t = 0$ s, a uniform freestream velocity of 1 m·s^{-1} is imposed.
- 2D nozzleFlow2D: This case investigates axisymmetric high-speed fuel injection. The computational domain includes a 3 mm diameter inlet connected to a gradually expanding throat. At the initial time $t = 0$ s, diesel fuel is injected at a velocity of 460 m·s^{-1} into a low-pressure gas environment maintained at atmospheric conditions. The simulation employs the volume of fluid (VoF) method coupled with a large eddy simulation (LES) turbulence model.

Table 4 shows the results of the computational experiments. The results confirm that Qwen 2.5-Max can handle selected CFD cases without needing RAG support. When geometric models and mesh files are given, the LLM can generate and debug simulations for rising bubble, falling droplet, airfoil, and cylinder cases. More complex cases, such as motorBike and nozzleFlow, revealed additional challenges. In the motorBike simulation, we supplied the full mesh and an explicit list of required field files, yet Qwen-2.5-Max wrote every face as a patch, producing the error "patch type 'patch' not constraint type 'empty'". Reprompting that asked only to convert the six free-stream faces to empty faces triggered a global string substitution that also altered the outlet boundaries, so the same error resurfaced in every iteration. For nozzleFlow, the error pattern depended on what the prompt contained. If the prompt does not include the pcorrFinal block, OpenFOAM is terminated with a "missing smoother keyword" message. When pcorrFinal was added, the solver advanced past that point but then failed because the model still did not create the mandatory 0/p rgh field, even though the prompt explicitly

asked for it. The failures related to geometry definition and physics consistency are also the main reasons for the failure of the other models in Table 3, highlighting that the LLM still struggles with complex geometry and advanced physics models.

4. Conclusion, limitations and outlook

We extended OpenFOAMGPT by integrating two significantly more affordable large language models, DeepSeek V3 and Qwen 2.5-Max, achieving cost savings of up to two orders of magnitude compared with OpenAI o1. These models are capable of handling common CFD tasks—such as modifying boundary conditions, turbulence models, and solver configurations—across a variety of test cases. We also explored a locally deployed QwQ-32B model (Q4 KM quantized model) running on a single desktop GPU, although it struggled to produce fully correct solver files for specialized engineering tasks, suggesting that smaller models may need domain-specific training or fine-tuning to handle complex scenarios.

Despite these encouraging results, certain challenges persist. In more intricate simulations—for instance, cases with complex geometry setups—zero-shot prompting alone often fell short. Repeated boundary-condition errors and missing solver keywords highlight the need for human oversight or additional AI guidance, especially when dealing with less-documented features of OpenFOAM. Although the less expensive models successfully reduce token costs, they sometimes suffer from narrower context windows and struggle with multistep error correction, indicating room for improvement in handling elaborate CFD workflows.

In the future, several avenues seem promising. First, smaller or mid-sized models could be fine-tuned via specialized CFD corpora to increase accuracy while keeping inference costs low. Second, bridging textual instructions with geometry and mesh data remains a hurdle—multimodal approaches and more sophisticated prompt engineering strategies could help LLMs interpret problem setups in a more intuitive way. Third, fully combining zero-shot techniques with retrieval-augmented generation may offer a practical blend of lower costs and more reliable outcomes. Finally, improving local deployment on consumer-grade GPUs—whether by scaling up model sizes or refining quantization—could reduce dependence on external APIs. By pursuing these directions, we can inch closer to creating robust, flexible, and truly cost-effective LLM-driven CFD solutions for both research and industry.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Xu Chu reports financial support was provided by The Royal Society. Xu Chu reports a relationship with The Royal Society that includes: funding grants. No conflict of Interests If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Wenkang Wang: Methodology, Investigation, Conceptualization. **Ran Xu:** Validation, Investigation. **Jingsen Feng:** Writing – original draft, Validation. **Qingfu Zhang:** Investigation. **Sandeep Pandey:** Software. **Xu Chu:** Writing – original draft, Validation, Software, Conceptualization.

Acknowledgments

This work was supported by the Royal Society (Grant No. RG\R1\251236), and the Fundamental Research Funds for the Central Universities of China (Grant No. JKF-2025055317102).

References

[1] R. Vinuesa, S.L. Brunton, Enhancing computational fluid dynamics with machine learning, Nat. Comput. Sci. 2 (6) (2022) 358–366.
[2] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid. Mech. 51 (2019) 357–377.
[3] A. Cremades, S. Hoyas, R. Vinuesa, Additive-feature-attribution methods: a review on explainable artificial intelligence for fluid dynamics and heat transfer, Int. J. Heat. Fluid. Flow. 112 (2025) 109662.
[4] G. Yang, R. Xu, Y. Tian, S. Guo, J. Wu, X. Chu, Data-driven methods for flow and transport in porous media: a review, Int. J. Heat. Mass Transf. 235 (2024) 126149.
[5] S. Pandey, J. Schumacher, K.R. Sreenivasan, A perspective on machine learning in turbulent flows, J. Turbul. 21 (9-10) (2020) 567–584.
[6] W. Wang and X. Chu. Optimized flow control based on automatic differentiation in compressible turbulent channel flows. *arXiv preprint arXiv:2410.23415*, 2024.
[7] J. Wu, H. Xiao, E. Paterson, Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework, Phys. Rev. Fluid. 3 (7) (2018) 074602.
[8] J. Ling, A. Kurzawski, J. Templeton, Reynolds averaged turbulence modelmodelling using deep neural networks with embedded invariance, J. Fluid. Mech. 807 (2016) 155–166.
[9] A. Beck, D. Flad, C. Munz, Deep neural networks for data-driven LES closure models, J. Comput. Phys. 398 (2019) 108910.
[10] X. Chu, S. Pandey, Non-intrusive, transferable model for coupled turbulent channel–porous media flow based upon neural networks, Phys. Fluid. 36 (2) (2024) 025112.
[11] X.I.A. Yang, S. Zafar, J.-X. Wang, H. Xiao, Predictive large-eddy-simulation wall modeling via physics-informed neural networks, Phys. Rev. Fluid. 4 (3) (2019) 034602.
[12] A. Beck, M. Kurz, Toward discretization-consistent closure schemes for large eddy simula- tion using reinforcement learning, Phys. Fluid. 35 (12) (2023).
[13] W. Chang, X. Chu, A. Fareed, S. Pandey, J. Luo, B. Weigand, E. Laurien, Heat transfer prediction of supercritical water with artificial neural networks, Appl. Therm. Eng. 131 (2018) 815–824.
[14] X. Chu, W. Chang, S. Pandey, J. Luo, B. Weigand, E. Laurien, A computationally light data-driven approach for heat transfer and hydraulic characteristics modeling of supercritical fluids: From dns to dnn, Int. J. Heat. Mass Transf. 123 (2018) 629–636.
[15] R. Vinuesa, S.L. Brunton, B.J. McKeon, The transformative potential of machine learning for experiments in fluid mechanics, Nat. Rev. Phys. 5 (9) (2023) 536–545.
[16] W. Wang, X. Chu, A. Lozano-Dur´an, R. Helmig, B. Weigand, Information transfer between turbu- lent boundary layer and porous media, J. Fluid. Mech. 920 (2021) A21.
[17] W. Wang, A. Lozano-Dur´an, R. Helmig, X. Chu, Spatial and spectral characteristics of information flux between turbulent boundary layers and porous media, J. Fluid. Mech. 949 (2022) A16.
[18] Y. Liu, W. Wang, G. Yang, H. Nemati, X. Chu, The interfacial modes and modal causality in a dispersed bubbly turbulent flow, Phys. Fluid. 35 (8) (2023).
[19] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. Leoni Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
[20] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.
[21] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
[22] B. Min, H. Ross, E. Sulem, A.P.B. Veyseh, T.H. Nguyen, O. Sainz, E. Agirre, I. Heintz, D. Roth, Recent advances in natural language processing via large pre-trained language models: a survey, ACM Comput. Surv. 56 (2) (2023) 1–40.
[23] P. Ma, T.-H. Wang, M. Guo, Z. Sun, J.B. Tenenbaum, D. Rus, C. Gan, and W. Matusik. Llm and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery. *arXiv preprint arXiv:2405.09783*, 2024.
[24] L. Song, C. Zhang, L. Zhao, and J. Bian. Pre-trained large language models for industrial control. *arXiv preprint arXiv:2308.03028*, 2023.
[25] Y.-J. Wang, B. Zhang, J. Chen, and K. Sreenath. Prompt a robot to walk with large language models. *arXiv preprint arXiv:2309.09969*, 2023.
[26] K. Huang, Y. Qu, H. Cousins, W.A. Johnson, D. Yin, M. Shah, D. Zhou, R. Altman, M. Wang, and L. Cong. Crispr-gpt: An llm agent for automated design of gene-editing experiments. *arXiv preprint arXiv:2404.18021*, 2024.
[27] K. Chibwe, D. Mantilla-Calderon, F. Ling, Evaluating gpt models for automated literature screening in wastewater-based epidemiology, ACS Environ. Au (2024).
[28] M.C. Ramos, C. Collison, A.D. White, A review of large language models and autonomous agents in chemistry, Chem. Sci. (2024).
[29] M.J. Buehler, Cephalo: Multi-modal vision-language models for bio-inspired materials analysis and design, Adv. Funct. Mater. (2024) 2409531.
[30] M. Du, Y. Chen, Z. Wang, L. Nie, D. Zhang, Large language models for automatic equation discovery of nonlinear dynamics, Phys. Fluid. 36 (9) (2024).
[31] X. Zhang, Z. Xu, G. Zhu, C.M.J. Tay, Y. Cui, B.C. Khoo, L. Zhu, Using large language models for parametric shape optimization, 2024. URL https://arxiv.org/abs/2412.08072.
[32] Z. Xu, L. Zhu, Training microrobots to swim by a large language model, 2024. URL https://arxiv.org/abs/2402.00044.
[33] M. Zhu, A. Bazaga, and P. Li'o. Fluid-llm: Learning computational fluid dynamics with spatiotemporal-aware large language models. *arXiv preprint arXiv:2406.04501*, 2024.
[34] D. Kim, T. Kim, Y. Kim, Y.-H. Byun, T.S. Yun, A chatgpt-matlab framework for numerical modeling in geotechnical engineering applications, Comput. Geotech. 169 (2024) 106237.

[35] Y. Chen, X. Zhu, H. Zhou, and Z. Ren. Metaopenfoam: a llm-based multi-agent framework for cfd. *arXiv preprint arXiv:2407.21320*, 2024.
[36] Y. Chen, X. Zhu, H. Zhou, Z. Ren, Metaopenfoam 2.0: Large language model driven chain of thought for automating cfd simulation and post-processing, 2025. URL https://arxiv.org/abs/2502.00498.
[37] Z. Dong, Z. Lu, Y. Yang, Fine-tuning a large language model for automating computational fluid dynamics simulations, Theoret. Appl. Mech. Lett. 0 (2025) 20250424, doi:10.1016/j.taml.2025.100594. 1ISSN 2095-0349URL. http://taml.cstam.org.cn//article/id/d9c81dde-935f-43c2-bcef-bfa93458091c.
[38] S. Pandey, R. Xu, W. Wang, and X. Chu. Openfoamgpt: a rag-augmented llm agent for openfoam-based computational fluid dynamics. *arXiv preprint arXiv:2501.06327*, 2025.
[39] H.G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Comput. Phys. 12 (6) (1998) 620–631.
[40] S. Pandey, X. Chu, E. Laurien, B. Weigand, Buoyancy induced turbulence modulation in pipe flow at supercritical pressure under cooling conditions, Phys. Fluid. 30 (6) (2018) 065105.
[41] Y. Liu, X. Chu, G. Yang, B. Weigand, Simulation and analytical modeling of high--speed droplet impact onto a surface, Phys. Fluid. 36 (1) (2024).
[42] SGLang ContributorsSglang documentation, GitHub repository, 2023. https://github.com-s-1gl-project-s-1glang.
[43] W. Kwon, Z. Li, Y. Zhuang, et al., Efficient memory management for large lan- guage model serving with pagedattention, in: Proceedings of the 29th Symposium on Operating Systems Principles (SOSP), 2023, pp. 611–626.
[44] Ollama TeamOllama, 2024 URL https://ollama.ai/.Open-source-framework .
[45] A. Chowdhery, S. Narang, J. Devlin, et al., Palm: Scaling language modeling with pathways, J. Mach. Learn. Res. 24 (240) (2023) 1–113.
[46] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, Annu. Rev. Fluid. Mech. 52 (1) (2020) 477–508.