CoE-Ops: Collaboration of LLM-based Experts for AIOps Question Answering

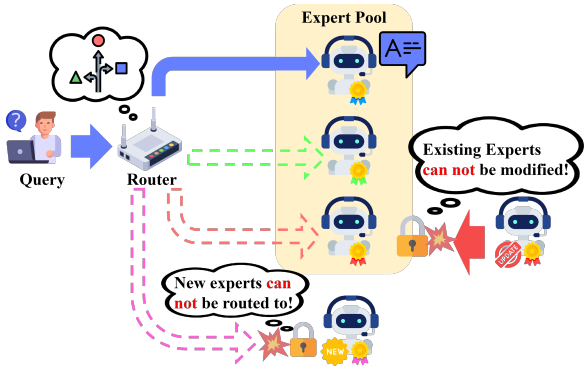
Anonymous ACL submission

Abstract

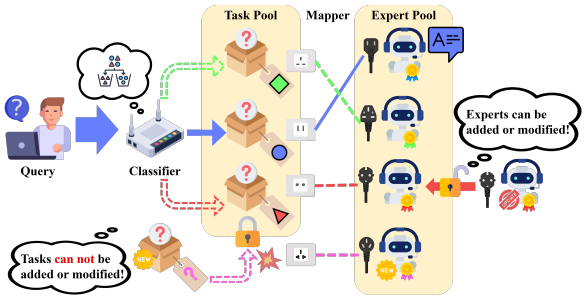
While Large Language Models (LLMs) have advanced the paradigm of AIOps, a single monolithic model struggles to cover the comprehensive DevOps lifecycle—ranging from low-level fault analysis to high-level release planning—due to domain knowledge constraints. Although ensemble learning offers a potential solution, existing approaches often lack the scalability to adapt to dynamic task shifts. To address these challenges, we propose CoE-Ops, a Collaboration-of-Experts framework designed for complex AIOps Question-Answering (QA). CoE-Ops incorporates a training-free, general-purpose LLM as a task classifier, augmented by Retrieval-Augmented Generation (RAG) to precisely route queries across heterogeneous expert models without fine-tuning. This mechanism enables robust handling of both concrete (e.g., anomaly detection) and abstract (e.g., operation) tasks. Extensive evaluations on the DevOps-Eval benchmark demonstrate that CoE-Ops significantly outperforms state-of-the-art baselines: it achieves a 72% improvement in routing accuracy for high-level tasks compared to existing CoE methods, delivers an 8% accuracy gain over the best standalone experts, and surpasses large-scale Mixture-of-Experts (MoE) models by up to 14% in overall accuracy with fewer parameters.

1 Introduction

DevOps has established itself as a critical methodology for bridging software development and IT operations, emphasizing continuous delivery across an eight-phase lifecycle(Jabbari et al., 2016). With the advent of artificial intelligence, this paradigm has evolved into AIOps, which leverages machine learning to automate complex operational workflows(Dang et al., 2019; Notaro et al., 2021). Recently, the emergence of Large Language Models (LLMs) has further revolutionized this field, offering powerful reasoning capabilities for diverse tasks. However, given the heterogeneity of DevOps subtasks—ranging from code generation to anomaly detection—relying on a single monolithic model often leads to inadequate coverage and deployment failures in specialized scenarios(Diaz-De-Arcaya et al., 2023; Khan et al., 2025). Consequently, research is shifting towards *Collaboration-of-Experts (CoE)* paradigms, which orchestrate multiple domain-specific LLMs to address multifaceted requirements.



(a) Analyzing Model Scalability Limitations in One-Stage CoE via End-to-end Routing.



(b) Analyzing Task Scalability Limitations in CoE via Two-Stage Expert Routing.

Figure 1: Model and Task Scalability Analysis in Collaboration of Experts.

Despite the promise of multi-agent collaboration, existing CoE frameworks in AIOps face severe scalability limitations, categorized here into **Model Scalability** and **Task Scalability** (as illus-

trated in Fig. 1). First, regarding *Model Scalability* (Fig. 1a), traditional CoE frameworks utilizing **end-to-end routing** suffer from rigid coupling between the router and expert models. Incorporating newly released AIOps LLMs or replacing outdated ones necessitates computationally expensive retraining of the entire router, as the system cannot dynamically adapt to changes in the expert pool. Second, while some approaches adopt **two-stage routing** (Classifier-Mapper) to mitigate model rigidity, they introduce *Task Scalability* bottlenecks (Fig. 1b). These frameworks typically employ discriminative classifiers with fixed output dimensions tailored to specific datasets. When task contexts evolve or new task categories emerge (changing from N to M classes), such classifiers fail to generalize without structural modification and retraining on in-domain data, relying heavily on parametric knowledge rather than adaptive reasoning.

To address these challenges, we propose **CoE-Ops**, a scalable and adaptive question answering framework designed for the comprehensive DevOps lifecycle. To resolve *Model Scalability*, CoE-Ops employs a refined two-stage mechanism that decouples task classification from expert selection, enabling the dynamic composition and role-switching of expert models through flexible mapping updates without retraining. Crucially, to overcome *Task Scalability* limitations, we replace rigid discriminative classifiers with a **Retrieval-Augmented Generation (RAG) enhanced classifier**. By retrieving relevant contextual knowledge and integrating it into the prompt, our framework transcends fixed-category constraints, allowing it to adaptively interpret and route high-level, abstract AIOps tasks across unfamiliar scenarios. We validate CoE-Ops on the DevOps-Eval benchmark, demonstrating that it effectively balances diverse expert capabilities and achieves superior performance in complex operational environments.

Our key contributions are summarized as follows:

- A Collaboration-of-Expert framework CoE-Ops based on two-stage expert routing and a general-purpose large language model as task classifier, enabling dynamic switching across diverse AIOps task domains and LLM ensembles.
- An enhanced task classifier empowered by retrieval-augmented generation technology, specifically designed to address high-level task representations inherent in DevOps scenarios.
- Comprehensive empirical validation on DevOps-EVAL benchmarks with multiple task-expert configurations and over a dozen AIOps expert models, systematically validating CoE-Ops's dual scalability in task scalability and model scalability.

2 Related Work

2.1 Development and Operations

DevOps is a collaborative, cross-domain methodology centered on automating the continuous delivery of software updates (Leite et al., 2019). The integration of Artificial Intelligence into this paradigm has catalyzed the emergence of AIOps (Brahmandam, 2025), which aims to enhance operational efficiency through automated reasoning and data-driven decision-making.

AIOps leverages AI and ML technologies to efficiently build and operate large-scale online services and applications in software engineering(Dang et al., 2019). Most existing AIOps implementations rely on data from a limited number of domains(Notaro et al., 2021) and predominantly employ supervised learning techniques(Mondru et al., 2024). Consequently, their proposed models are often confined to specific DevOps subdomains rather than being deployable across the entire ecosystem(Khan et al., 2025). A critical challenge for AIOps lies in selecting and integrating appropriate machine learning models(Hua, 2021)(Mulongo, 2024) to ensure adaptability to diverse use cases while fulfilling heterogeneous(Krishnamurthy and Neelanath, 2025) and evolving requirements(Brahmandam, 2025).

2.2 Ensemble Learning with Large Language Models

Ensemble learning with large language models involves the systematic utilization of multiple LLMs, each designed to handle user queries during downstream inference to capitalize on their individual strengths(Chen et al., 2025)(Varangot-Reille et al., 2025). Depending on the strategy for model integration, ensemble learning can be categorized into two paradigms: Mixture-of-Experts (MoE) and Collaboration-of-Experts (CoE).

In recent years, MoE models have become a primary choice for foundation models(Liu et al., 2024)(Yang et al., 2025) due to their computational

efficiency and strong generalization capabilities. In MoE systems, different expert modules possess distinct strengths, making efficient utilization a key challenge. FrugalGPT(Chen et al., 2023) and LLM-Blender(Jiang et al., 2023) aggregate outputs from various experts to generate final results, while others adopt voting strategies to select the optimal output(Sukhbaatar et al., 2024)(Si et al., 2023)(Li et al., 2024). However, these expert modules cannot complete tasks independently, and the selection and generation processes lack interpretability. As a result, the Collaboration-of-Experts framework has increasingly drawn attention from researchers.

CoE primarily facilitates synergistic interactions among experts by selecting one or several optimal experts for a given input. Early efforts explored the use of sub-networks as expert models(Zhang et al., 2021)(Huang et al., 2024). With the proliferation of large-scale models, CoE has shifted focus toward incorporating diverse performance metrics, such as answer accuracy(Shnitzer et al., 2023)(Maurya et al., 2025), inference cost(Šakota et al., 2024)(Stripelis et al., 2024a)(Stripelis et al., 2024b), and problem difficulty(Ong et al., 2024)(Ding et al., 2024). A core research direction in CoE involves the design of routing algorithms for large models(Shnitzer et al., 2023). For instance, cascading networks(Hu et al., 2024)(Yue et al., 2025) have been proposed, or large models are represented as nodes(Guha et al., 2024)(Feng et al., 2024) or vector embeddings(Jitkrittum et al., 2025), with probabilistic methods(Zhang et al., 2025) employed to predict routing outcomes. Recent studies further integrate reinforcement learning(Lu et al., 2024)(Nguyen et al., 2024)(Zhao et al., 2024)(Wang et al., 2025) to refine expert routing strategies and introduce hardware-aware optimizations(Prabhakar et al., 2024)(Suo et al., 2025) for efficient expert model loading. To address the lack of interpretability in routing decisions, a two-stage expert routing framework(Jain et al., 2024)(Wang et al., 2024) has been developed (as shown in Fig 1b). This framework first categorizes input problems and then selects the most suitable expert for each category, thereby enhancing both the explainability of routing decisions and the scalability of the overall system.

3 Methodology

The framework of our proposed CoE-Ops is shown in Fig. 2. It consists of a two-stage expert routing mechanism which replaces discriminative models with general-purpose LLMs enhanced by retrieval-augmented generation capabilities.

3.1 Two-stage Expert Routing

CoE-Ops primarily improves upon the two-stage expert routing mechanism proposed in seminal works including Composition of Experts(Jain et al., 2024) and Bench-CoE(Wang et al., 2024). During the original process of two-stage expert routing, the AIOps user's query is first classified by a pretrained or fine-tuned classifier to determine its task type. The query is then routed to the best-in-domain model for processing based on this label, as shown in Fig. 1b.

The task classifier in the two-stage expert routing can be abstracted as (1) shows.

$$\hat{T} = \underset{T \in \{T_1, \ldots, T_n\}}{\arg\max} \; P(T|X, \mathcal{C}) \tag{1}$$

where T represents the AIOps task, $\mathcal{C}$ represents the classification model, and X denotes the current input from user.

In particular, within the two-stage expert routing architecture of the Collaboration of Experts, the cardinality of candidate AIOps experts should adhere to the bounds specified in (2), since each AIOps expert model demonstrates expertise in a minimum of one specialized AIOps domain.

$$2 \leq N_{\text{expert}} \leq N_{\text{task}}, \tag{2}$$

where N_{task} denotes the number of AIOps tasks.

Following AIOps task categorization by the classifier, input AIOps queries are dynamically routed to domain-specialized expert models through a "task-expert" allocation mechanism, as mathematically formalized in (3).

$$f : \mathcal{T} \rightarrow \mathcal{E}, \tag{3}$$

where $\mathcal{T} = \{T_1, T_2, \ldots, T_M\}$ denotes the set of AIOps tasks, $\mathcal{E} = \{E_1, E_2, \ldots, E_N\}$ denotes the set of AIOps experts, M indicates the count of AIOps tasks, and N indicates the count of AIOps experts.

When developing the "task-expert" allocation mechanism, it is necessary to establish a metric for evaluating the capability of each expert model across different task domains. For AIOps queries involving multiple-choice questions and question-and-answer formats, the answer accuracy of the

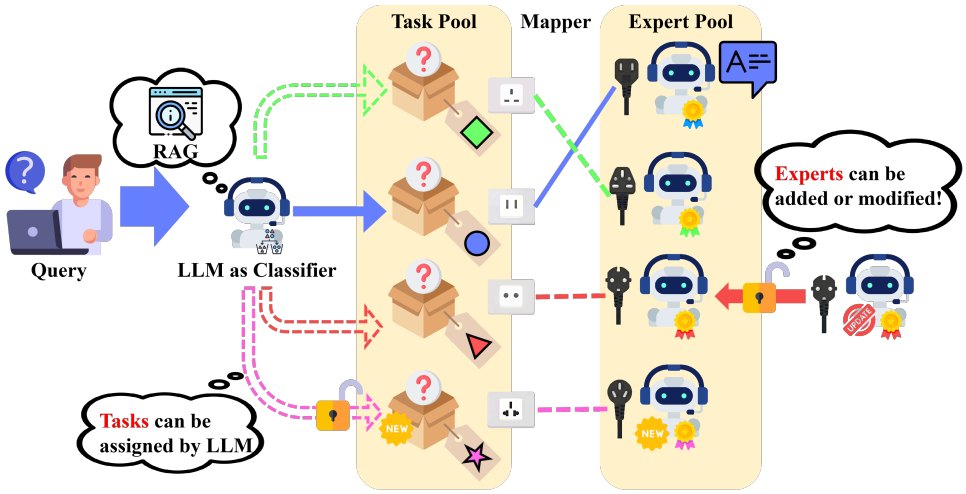


Figure 2: Overview of the CoE-Ops framework. CoE-Ops improves the CoE based on Two-Stage Expert Routing by modifying its process. First, the discriminative model-based classifier is replaced with an LLM-based one. Subsequently, the prompt is enhanced by extracting a task list from benchmark datasets and employing Retrieval-Augmented Generation (RAG) to retrieve relevant context for the current input, thereby assisting the LLM-based classifier in classification.

expert model can serve as a suitable evaluation metric. This accuracy measurement, as shown in (4), provides a quantitative basis for assessing model performance.

$$\mathrm{Acc}(M, T_i) = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbb{I}(M(\mathcal{X}_{ij}) = \mathcal{A}_{ij}) \quad (4)$$

where N_i denotes the number of AIOps queries in the AIOps task T_i, M represents the expert model, $\mathcal{X}_i$ stands for the AIOps queries in the AIOps task T_i, and $\mathcal{A}_{ij}$ indicates the correct answer to the AIOps query $\mathcal{X}_{ij}$.

Upon construction of the capability assessment leaderboard, the expert model demonstrating superior accuracy within each task domain is designated as the optimal solution for the "task-expert" allocation, with formal validation provided in (5).

$$M_i^* = \arg\max_{M \in \mathcal{M}} \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbb{I}(M(\mathcal{X}_{ij}) = \mathcal{A}_{ij}) \quad (5)$$

where M_i^* denotes the best AIOps model on task T_i.

3.2 Classifier with General-purpose LLM

To overcome the limitations inherent in conventional two-stage expert routing CoE frameworks, particularly their dependence on repeated classifier fine-tuning or retraining across distinct task scenarios, we implement a dual enhancement strategy. First, the classifier component is replaced by a general-purpose LLM operating in zero-shot mode, thereby eliminating fine-tuning requirements. Second, a structured task-list prompting mechanism (see Prompt 1 in A) is integrated to ensure task scalability of the optimized architecture.

The enhanced framework enables dynamic adaptation to shifting task scenarios through prompt-based task list modification, eliminating the need for classifier pretraining or fine-tuning. This architectural innovation substantially reduces computational overhead while maintaining task scalability within the CoE paradigm.

The classification architecture of our framework, enhanced through the integration of prompt engineering and a general-purpose LLM, achieves formal abstraction as mathematically characterized in (6).

$$\hat{T} = \arg\max_{T \in \{T_1, T_2, \ldots, T_n\}} P(T | X, P, \mathcal{L}_{\text{General}}), \quad (6)$$

where P denotes the prompt with the task list, $\mathcal{L}_{\text{General}}$ represents the general-purpose LLM.

Notably, unlike fine-tuned classifiers, using a general-purpose LLM as a classifier may yield an "unknown" class result. This reflects the LLM's effort to reduce hallucination by refusing to force-classify ambiguous inputs. Thus, after incorporating prompts and a general-purpose LLM, an additional "unknown" class is needed. Consequently, the number of output task classes is modified as shown in (7).

$$N_{\text{predict task}} = N_{\text{task}} + N_{\text{unk}}, \quad (7)$$

where N_{unk} denotes the number of tasks of unknown types (typically equals 1).

In this case, we need to select an extra expert model for the "unknown" class. Our selection strategy, as shown in (8), is to choose the expert model with the highest average capability in all task domains to handle the "unknown" AIOps input.

$$M^*_{\text{unk}} = \arg\max_{M \in \mathcal{M}} \frac{1}{|\mathcal{T}|} \sum_{T_i \in \mathcal{T}} \text{Acc}(M, T_i) \quad (8)$$

where M^*_{unk} denotes the best AIOps model on "unknown" task.

For the expert models, we also avoid fine-tuning. Instead, we use prompts with chain of thought as the input. The prompt template is shown in Prompt 2 in Appendix A. In the multiple-choice setting, to assess expert capabilities via answer accuracy, we ask the model to return answers in a fixed format.

3.3 LLM Classifier Enhanced with RAG

Simply replacing the classifier in the two-stage expert router with a general-purpose LLM carries risks. In AIOps domains with abstract or high-level task (like plan, build, code, etc.), the LLM may struggle to link inputs to tasks due to limited information. To address this, context needs to be introduced to help the LLM better understand the AIOps inputs, establish task-input connections, and improve AIOps task prediction.

In this condition, we integrated retrieval-augmented generation into the two-stage LLM routing. The prompt template is shown in Prompt 3 in Appendix A. By retrieving similar questions and their categories to the input question, RAG aids the general-purpose LLM in determining the input's task category. This led to the improvement of the CoE-Ops framework in the scenarios with high-level AIOps tasks. Similar to other RAG approaches, the RAG process in our CoE-Ops can be divided into two sub-phases: Off-line and On-line, as abstractly shown in (9).

$$P(o|q) = \sum_{c \in \mathcal{C}} P(a|q, c) P(c|q), \quad (9)$$

where q denotes the encoded vector of the query, c represents the encoded vector of the context, and o denotes the output of the LLM classifier.

During the Off-line stage, existing textual data is encoded, as shown in (10).

$$c = \text{Encoder}_{\text{RAG}}(C), \quad (10)$$

where C denotes the context data.

In the On-line stage, the input AIOps query is first encoded into a vector by the encoder, as shown in (11).

$$q = \text{Encoder}_{\text{RAG}}(Q), \quad (11)$$

where Q denotes the query data.

After obtaining the input AIOps query vector and knowledge base vectors, we perform retrieval to find the knowledge base vectors most similar to the input vector. The retrieval process is described by (12).

$$P(c|q) = \frac{\exp(\sin(q, c))}{\sum_{c \in \mathcal{C}} \exp(\sin(q, c))}. \quad (12)$$

The formula for the Retriever's similarity calculation is shown in (13).

$$\text{sim}(q, c) = q \cdot c. \quad (13)$$

4 Experimental Setup

To verify the effectiveness of CoE-Ops in complex AIOps scenarios, we conduct evaluations on the DevOps-Eval[1] benchmark. DevOps-Eval comprises a vast collection of multiple-choice questions categorized into English and Chinese subsets. Specifically, the English subset focuses on low-level AIOps tasks, while the Chinese subset covers the comprehensive DevOps lifecycle (high-level tasks). Detailed statistics are provided in Tab. 4 of Appendix B.

We establish a task-expert mapping (Tab. 5, Appendix C) covering low-level (Set A) and high-level (Set B) AIOps tasks. While smaller models (Sets 1 and 3) are locally hosted, larger models (Sets 2 and 4) are accessed via APIs due to their scale. Notably, CoE-Ops demonstrates high extensibility: switching between sets only requires updating prompts and mappings without structural changes or fine-tuning. We evaluate performance using Accuracy, Precision, Recall, and F1-score, further leveraging confusion matrices and radar charts to analyze model capabilities across diverse domains.

5 Experiment

5.1 CoE-Ops Effectiveness Evaluation

To evaluate whether CoE-Ops effectively harmonizes diverse expert models via ensemble learning,

[1] https://hf-mirror.com/datasets/codefuse-ai/CodeFuse-DevOps-Eval

Models	Accuracy(%)					Average(%)			
	LP	RCA	TSAD	TSC	TSF	Acc	Prec	Rec	F1
Model Size $\leq$ 8B									
Internlm-7B	47.71	20.40	27.00	**42.50**	35.62	35.07	37.05	35.07	34.36
Internlm-chat-7B	**61.71**	1.20	22.33	33.50	**49.38**	35.99	39.47	35.99	35.42
CodeFuse-DevOps-Model-7B-Base	32.00	20.40	29.67	29.50	27.81	28.17	29.57	28.17	25.39
CodeFuse-DevOps-Model-7B-Chat	38.86	24.40	25.33	35.00	28.44	30.56	31.71	30.56	30.36
CoE-Ops(DeepSeek-R1-Distill-Qwen-7B)	60.86	19.20	25.00	37.50	**49.38**	40.07	42.40	40.07	39.60
CoE-Ops(DeepSeek-V3)	**61.71**	**29.20**	**31.33**	**42.50**	49.38	**44.08**	**46.82**	**44.08**	**43.58**
Model Size $\geq$ 8B									
Glm-4-flash	89.43	53.60	**42.00**	32.00	78.44	62.54	64.50	62.54	63.16
Codegeex-4	82.29	56.40	31.00	44.50	50.62	54.44	63.84	54.44	58.65
Ministral-8b	**90.86**	87.60	40.00	28.00	80.62	68.38	69.07	68.38	68.70
Random-CoE	86.57	62.40	34.67	31.50	66.88	59.15	62.63	59.15	60.84
Mixtral-8x7b-instruct	80.29	38.00	34.33	**48.50**	66.56	55.56	61.15	55.56	57.99
Bench-CoE	90.29	87.60	41.33	29.50	81.56	68.94	70.30	68.94	69.58
CoE-Ops(DeepSeek-R1-Distill-Qwen-7B)	87.14	**88.80**	38.00	42.50	80.00	69.15	71.13	69.15	70.10
CoE-Ops(DeepSeek-V3)	**90.86**	86.00	37.67	44.50	**83.12**	**70.49**	**72.29**	**70.49**	**71.31**

Table 1: Performance Comparison on DevOps-Eval English Subset. "LP" denotes Log Parser, "RCL" denotes Root Cause Analysis,"TSAD" denotes Time Series Anomaly Detection, "TSC" denotes Time Series Classification, "TSF" denotes Time Series Forecasting.

we conduct experiments across Expert Sets 1–4 on Task Sets A and B (see Tab. 5 in Appendix C). We employ DeepSeek-R1-Distill-Qwen-7B (local) and DeepSeek-V3 (API) as classifiers. For RAG, we utilize the DevOps-Eval eval split as context, encoded by all-MiniLM-L6-v2. Inputs are dynamically routed to experts based on classification results, with performance measured via Accuracy, Precision, Recall, and F1-score.

Performance Gains in Low-level Tasks. As shown in Tab. 1, CoE-Ops consistently outperforms all standalone experts on the English subset. Notably, with the DeepSeek-V3 classifier, CoE-Ops achieves 44.08% Accuracy, a significant improvement of 4% to 8% over the best-performing individual models in both the $\leq$8B and $\geq$8B categories. This superiority is further visualized in the capability radar charts (Fig. 5 and Fig. 6 in Appendix D).

Scalability and Robustness in High-level Tasks. On the more challenging Task Set B (Chinese subset, Tab. 2), CoE-Ops maintains its advantage despite increased classification complexity. It achieves a peak average Accuracy of 75.6%, outperforming individual experts across most DevOps stages. This balanced capability enhancement is further evidenced by the radar charts in the Appendix (Fig. 7 and Fig. 8 in Appendix D); while standalone models often exhibit "capability gaps" in specific stages (e.g., Operate or Plan), CoE-Ops expands the coverage area across all dimensions, demonstrating superior robustness and model-agnostic scalability in complex, high-level AIOps scenarios.

Summary. Across diverse expert configurations and linguistic domains, CoE-Ops proves highly scalable. By only adjusting prompts and task-expert mappings, the framework achieves consistent performance gains without model retraining, establishing its efficiency as a model-agnostic ensemble solution for AIOps.

5.2 Classifier Scalability and RAG Impact

To assess the scalability and robustness of our task routing mechanism, we conduct an ablation study on the Classifier component across Task Sets A and B. We compare our training-free classifiers (Classifier 1 and 2) against two baselines: (1) a version without RAG enhancement, and (2) Bench-CoE, which utilizes a fine-tuned classifier. The results are summarized in Tab. 3 and visualized via confusion matrices in Fig. 3 and Fig. 4.

Efficiency in Low-level Tasks (Set A). In the English subset, both CoE-Ops classifiers demonstrate superior generalization without any fine-tuning. Specifically, Classifier 2 (DeepSeek-V3) achieves a perfect 100% accuracy, while Classifier 1 (R1-Distill-7B) significantly outperforms the fine-tuned

Models	Accuracy(%)								Average(%)			
	Build	Code	Deploy	Monitor	Operate	Plan	Release	Test	Acc	Prec	Rec	F1
Model Size $\leq$ 8B												
Internlm-chat-7b	76.61	59.95	78.43	60.65	40.27	59.09	67.45	77.19	54.20	53.63	54.20	53.56
Mathstral-7B-v0.1	73.39	67.15	74.51	61.57	54.78	51.52	75.00	78.07	62.74	62.77	62.74	62.47
Qwen2-7B-Instruct	75.69	68.05	75.69	58.80	55.9	**63.64**	70.75	78.95	63.57	64.44	63.57	63.32
CoE-Ops(DeepSeek-R1-Distill-Qwen-7B)	77.98	69.04	80.39	62.96	56.49	56.06	68.40	79.82	64.52	64.93	64.52	64.24
CoE-Ops(DeepSeek-V3)	75.69	68.89	78.04	63.89	55.85	57.58	67.92	82.89	64.14	64.44	64.14	63.86
Model Size $\geq$ 8B												
Doubao-1.5-lite-32k	82.11	**77.90**	**82.35**	65.74	67.32	54.55	80.66	**85.53**	73.21	73.73	73.21	73.47
Gemma-2-27b-it	**89.91**	73.35	**82.35**	**68.06**	71.09	56.06	**84.43**	84.65	74.22	74.13	74.22	74.14
Glm-4-flash	80.28	74.41	81.18	63.43	59.87	59.09	81.60	83.33	68.60	68.23	68.60	68.26
mixtral-8x7b-instruct	81.19	68.51	78.04	64.81	56.79	57.58	78.30	83.33	65.26	66.89	65.26	65.94
CoE-Ops(DeepSeek-R1-Distill-Qwen-7B)	81.65	75.62	80.78	64.35	71.93	57.58	78.77	83.33	74.28	74.79	74.28	74.52
CoE-Ops(DeepSeek-V3)	84.4	**77.90**	78.82	**68.06**	**72.42**	**63.64**	82.08	83.33	**75.60**	**75.91**	**75.60**	**75.75**

Table 2: Performance comparison on DevOps-Eval Chinese Subset.

Classifiers	**Task Set A (English)**				**Task Set B (Chinese)**			
	Acc (%)	Prec (%)	Rec (%)	F1 (%)	Acc (%)	Prec (%)	Rec (%)	F1 (%)
Random Select	20.00	–	–	–	12.50	–	–	–
Bench-CoE	62.46	52.69	62.46	55.35	4.94	11.86	4.94	0.83
DeepSeek-R1-Distill-Qwen-7B w/o RAG	77.11	87.66	77.11	81.52	13.91	32.65	13.91	14.66
DeepSeek-R1-Distill-Qwen-7B	80.92	95.62	80.92	87.51	43.84	71.47	43.84	50.43
DeepSeek-V3 w/o RAG	100.0	100.0	100.0	100.0	24.93	41.67	24.93	26.54
DeepSeek-V3	**100.0**	**100.0**	**100.0**	**100.0**	**77.22**	**79.79**	**77.22**	**77.22**

Table 3: Classification Performance comparison on DevOps-Eval English Subset (Task Set A) and DevOps-Eval Chinese Subset (Task Set B).

Bench-CoE by nearly 20% in F1-score. The concentration of high-value diagonals in the heatmaps (Fig. 3) further confirms the precision of our zero-shot routing in less complex domains.

Robustness and Recovery in High-level Tasks (Set B). The advantage of CoE-Ops becomes more pronounced as task complexity scales. While the fine-tuned Bench-CoE experiences a catastrophic performance drop (Accuracy falling to 4.94%) when shifting to the comprehensive DevOps lifecycle, our framework maintains substantial lead. Notably, RAG integration plays a critical role in complex scenarios: for Classifier 2, RAG augmentation boosts Accuracy from 24.93% to 77.22%. This "performance recovery" is visually evident in the clearer cluster separations in Fig. 4 compared to non-RAG baselines.

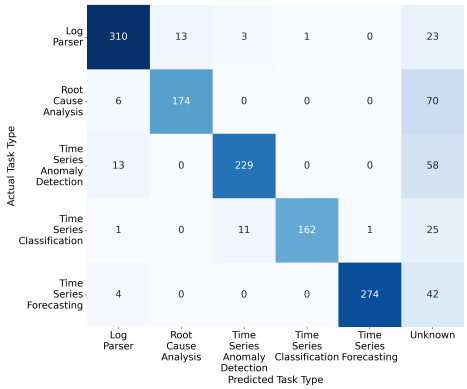
Summary. These results demonstrate that while traditional fine-tuning-based classifiers struggle with domain shift and task scaling, the CoE-Ops framework, empowered by RAG and advanced LLMs, exhibits exceptional task scalability. It effectively handles both increased task categories and hierarchical complexity in AIOps without the need for task-specific retraining.

5.3 Efficiency and Comparative Analysis

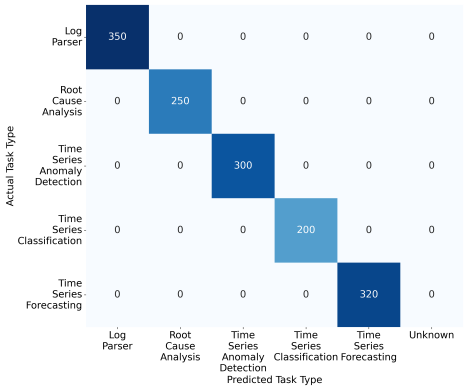
To evaluate the efficiency of CoE-Ops, we compare it against state-of-the-art Mixture-of-Experts (MoE) and CoE frameworks. A key distinction lies in model scale: while Mixtral-8x7B-Instruct reaches approximately 56B parameters, our framework's largest expert configuration utilizes only 27B parameters. We conduct evaluations on Task Sets A and B, using Random-CoE (random routing) and Bench-CoE as baselines. Note that Bench-CoE is excluded from Task Set B due to its suboptimal classification performance. Results are detailed in Tab. 1 and Tab.2, with multi-dimensional capabilities visualized in Fig. 9 and Fig.10.

Superior Parameter Efficiency. As indicated in Tab. 1, CoE-Ops consistently outperforms both MoE and conventional CoE baselines in the AIOps domain. Notably, our framework, leveraging an ensemble of significantly smaller models, surpasses the 56B Mixtral in overall accuracy. This suggests that dynamic, RAG-enhanced routing among specialized experts is more effective than the static routing or dense computation of larger unified models.

Conclusion on Robustness. The radar charts (Fig. 9 and Fig. 10) further corroborate that CoE-

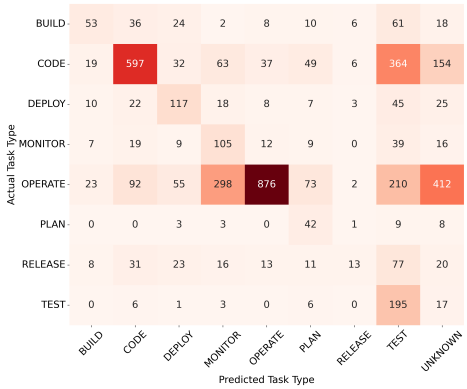


(a) DeepSeek-R1-Distill-Qwen-7B (Classifier 1) with RAG

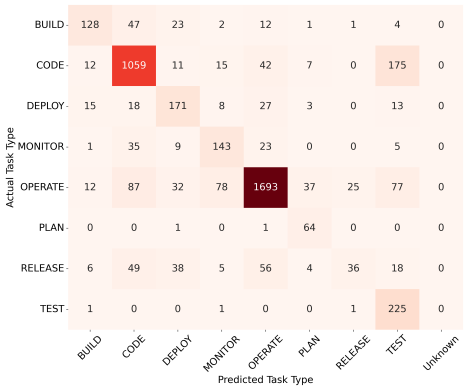


(b) DeepSeek-V3 (Classifier 2) with RAG

Figure 3: Heatmap visualization of confusion matrices on DevOps-EVAL English (Task Set A).



(a) DeepSeek-R1-Distill-Qwen-7B (Classifier 1) with RAG



(b) DeepSeek-V3 (Classifier 2) with RAG

Figure 4: Heatmap visualization of confusion matrices for two classifiers on DevOps-EVAL Chinese (Task Set B).

Ops achieves a more balanced and expansive capability profile. By surpassing larger-scale models with fewer total parameters, CoE-Ops demonstrates superior parameter efficiency and robustness, establishing itself as a viable solution for resource-constrained AIOps environments.

6 Conclusion

This paper presents CoE-Ops, a scalable framework that resolves the "capability-scalability" dilemma in AIOps QA by harmonizing heterogeneous expert models. By employing a two-stage routing mechanism driven by a RAG-enhanced general LLM, CoE-Ops effectively decouples task classification from expert selection. This design not only circumvents the need for costly retraining during task scenario transitions—thereby ensuring superior task scalability—but also significantly mitigates hallucinations in high-level, abstract scenarios. Empirical results confirm that our approach outperforms massive MoE models through efficient collaboration among smaller, specialized experts. For future work, we plan to explore the automated construction of expert capability rankings to facilitate fully autonomous collaboration. Furthermore, we aim to integrate CoE-Ops with multi-agent systems, establishing a multi-tiered collaboration paradigm to address increasingly complex, interactive AIOps workflows.

Limitations

Despite its effectiveness, CoE-Ops has three primary limitations. First, its dependence on external APIs introduces risks of service instability and network latency, although checkpoint mechanisms partially mitigate these. Second, while RAG significantly reduces classification hallucinations, misrouting risks persist under extreme out-of-distribution (OOD) scenarios. Finally, although the framework is training-free, the inference cost of maintaining multiple expert models (local or remote) remains a consideration for resource-constrained environments.

Ethical Considerations

We address potential ethical concerns as follows. Data Privacy: Using remote APIs involves transmitting telemetry data to external servers; we provide local deployment options to mitigate these privacy risks for sensitive AIOps environments. Content Safety: API calls are subject to content filtering, which may occasionally trigger invocation failures on sensitive test inputs. AI Stewardship: CoE-Ops is designed as a decision-support tool to enhance operator efficiency, not to replace human oversight in critical DevOps lifecycles.

References

Balajee Asish Brahmandam. 2025. Beyond devops: The evolution toward intelligent it operations with aiops and mlops.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Zhijun Chen, Jingzheng Li, Pengpeng Chen, Zhuoran Li, Kai Sun, Yuankai Luo, Qianren Mao, Ming Li, Likang Xiao, Dingqi Yang, and 1 others. 2025. Harnessing multiple large language models: A survey on llm ensemble. *arXiv preprint arXiv:2502.18036*.

Yingnong Dang, Qingwei Lin, and Peng Huang. 2019. Aiops: real-world challenges and research innovations. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 4–5. IEEE.

Josu Diaz-De-Arcaya, Ana I Torre-Bastida, Gorka Zárate, Raúl Miñón, and Aitor Almeida. 2023. A joint study of the challenges, opportunities, and roadmap of mlops and aiops: A systematic survey. *ACM Computing Surveys*, 56(4):1–30.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*.

Tao Feng, Yanzhen Shen, and Jiaxuan You. 2024. Graphrouter: A graph-based router for llm selections. *arXiv preprint arXiv:2410.03834*.

Neel Guha, Mayee Chen, Trevor Chow, Ishan Khare, and Christopher Re. 2024. Smoothie: Label free language model routing. *Advances in Neural Information Processing Systems*, 37:127645–127672.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*.

Yunke Hua. 2021. *A systems approach to effective aiops implementation*. Ph.D. thesis, Massachusetts Institute of Technology.

Shaomang Huang, Jianfeng Pan, and Hanzhong Zheng. 2024. Ccoe: A compact llm with collaboration of experts. *arXiv e-prints*, pages arXiv–2407.

Ramtin Jabbari, Nauman Bin Ali, Kai Petersen, and Binish Tanveer. 2016. What is devops? a systematic mapping study on definitions and practices. In *Proceedings of the scientific workshop proceedings of XP2016*, pages 1–11.

Swayambhoo Jain, Ravi Raju, Bo Li, Zoltan Csaki, Jonathan Li, Kaizhao Liang, Guoyao Feng, Urmish Thakkar, Anand Sampat, Raghu Prabhakar, and 1 others. 2024. Composition of experts: A modular compound ai system leveraging large language models. *arXiv preprint arXiv:2412.01868*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, and 1 others. 2025. Universal model routing for efficient llm inference. *arXiv preprint arXiv:2502.08773*.

Ahmad Faraz Khan, Azal Ahmad Khan, Anas Mohamed, Haider Ali, Suchithra Moolinti, Sabaat Haroon, Usman Tahir, Mattia Fazzini, Ali R Butt, and Ali Anwar. 2025. Lads: Leveraging llms for ai-driven devops. *arXiv preprint arXiv:2502.20825*.

Dasaprakash Krishnamurthy and Vinod Neelanath. 2025. Establishing a robust llmops framework for intelligent automation: Strategies and best practices. In *2025 Emerging Technologies for Intelligent Systems (ETIS)*, pages 1–5. IEEE.

Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, and Paulo Meirelles. 2019. A survey of devops concepts and challenges. *ACM computing surveys (CSUR)*, 52(6):1–35.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024. More agents is all you need. *arXiv preprint arXiv:2402.05120*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. Routing to the expert: Efficient reward-guided ensemble of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies (Volume 1: Long Papers), pages 1964–1974.

Kaushal Kumar Maurya, KV Aditya Srivatsa, and Ekaterina Kochmar. 2025. Selectllm: Query-aware efficient selection algorithm for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20847–20863.

Anil Kumar Mondru, Ramesh Bollavathini Shreyas, and Thanay Sisir Anabathula. 2024. A roadmap to success: Strategies and challenges in adopting aiops for it operations. *International Journal of Interpreting Enigma Engineers (IJIEE)*, 1(2).

Ndala Yves Mulongo. 2024. Key performance indicators of artificial intelligence for it operations (aiops). In *2024 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–8. IEEE.

Quang H Nguyen, Thinh Dao, Duy C Hoang, Juliette Decugis, Saurav Manchanda, Nitesh V Chawla, and Khoa D Doan. 2024. Metallm: A high-performant and cost-efficient dynamic framework for wrapping llms. *arXiv preprint arXiv:2407.10834*.

Paolo Notaro, Jorge Cardoso, and Michael Gerndt. 2021. A survey of aiops methods for failure management. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(6):1–45.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. 2024. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*.

Raghu Prabhakar, Ram Sivaramakrishnan, Darshan Gandhi, Yun Du, Mingran Wang, Xiangyu Song, Kejie Zhang, Tianren Gao, Angela Wang, Xiaoyan Li, and 1 others. 2024. Sambanova sn40l: Scaling the ai memory wall with dataflow and composition of experts. In *2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 1353–1366. IEEE.

Marija Šakota, Maxime Peyrard, and Robert West. 2024. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 606–615.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. 2023. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*.

Chenglei Si, Weijia Shi, Chen Zhao, Luke Zettlemoyer, and Jordan Boyd-Graber. 2023. Getting more out of mixture of language model reasoning experts. *arXiv preprint arXiv:2305.14628*.

Dimitris Stripelis, Zijian Hu, Jipeng Zhang, Zhaozhuo Xu, Alay Shah, Han Jin, Yuhang Yao, Salman Avestimehr, and Chaoyang He. 2024a. Polyrouter: A multi-llm querying system. *arXiv e-prints*, pages arXiv–2408.

Dimitris Stripelis, Zhaozhuo Xu, Zijian Hu, Alay Dilipbhai Shah, Han Jin, Yuhang Yao, Jipeng Zhang, Tong Zhang, Salman Avestimehr, and Chaoyang He. 2024b. Tensoropera router: A multi-model router for efficient llm inference. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 452–462.

Sainbayar Sukhbaatar, Olga Golovneva, Vasu Sharma, Hu Xu, Xi Victoria Lin, Baptiste Rozière, Jacob Kahn, Daniel Li, Wen-tau Yih, Jason Weston, and 1 others. 2024. Branch-train-mix: Mixing expert llms into a mixture-of-experts llm. *arXiv preprint arXiv:2403.07816*.

Jiashun Suo, Xiaojian Liao, Limin Xiao, Li Ruan, Jinquan Wang, Xiao Su, and Zhisheng Huo. 2025. Coserve: Efficient collaboration-of-experts (coe) model inference with limited memory. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pages 178–191.

Clovis Varangot-Reille, Christophe Bouvard, Antoine Gourru, Mathieu Ciancone, Marion Schaeffer, and François Jacquenet. 2025. Doing more with less–implementing routing strategies in large language model-based systems: An extended survey. *arXiv preprint arXiv:2502.00409*.

Xinyuan Wang, Yanchi Liu, Wei Cheng, Xujiang Zhao, Zhengzhang Chen, Wenchao Yu, Yanjie Fu, and Haifeng Chen. 2025. Mixllm: Dynamic routing in mixed large language models. *arXiv preprint arXiv:2502.18482*.

Yuanshuai Wang, Xingjian Zhang, Jinkun Zhao, Siwei Wen, Peilin Feng, Shuhao Liao, Lei Huang, and Wenjun Wu. 2024. Bench-coe: a framework for collaboration of experts from benchmark. *arXiv preprint arXiv:2412.04167*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. 2025. Masrouter: Learning to route llms for multi-agent systems. *arXiv preprint arXiv:2502.11133*.

Tuo Zhang, Asal Mehradfar, Dimitrios Dimitriadis, and Salman Avestimehr. 2025. Leveraging uncertainty estimation for efficient llm routing. *arXiv preprint arXiv:2502.11021*.

Yikang Zhang, Zhuo Chen, and Zhao Zhong. 2021. Collaboration of experts: Achieving 80% top-1 accuracy on imagenet with 100m flops. *arXiv preprint arXiv:2107.03815*.

Zesen Zhao, Shuowei Jin, and Z Morley Mao. 2024. Eagle: Efficient training-free router for multi-llm inference. *arXiv preprint arXiv:2409.15518*.

A Prompt Templates

In this section, we present the specific prompt templates employed in our experiments to facilitate reproducibility. Prompt 1 serves as the baseline instruction for the general-purpose task classifier. Adopting a zero-shot setting, it directs the LLM to map the input query directly to a predefined list of AIOps tasks without external context. Prompt 2 is utilized by the individual AIOps expert models during the inference phase. To enhance reasoning accuracy on complex DevOps problems, this prompt explicitly incorporates Chain-of-Thought (CoT) instructions ("Think step by step"), guiding the experts to decompose the problem before concluding with the final answer. Finally, Prompt 3 represents the advanced configuration of the task classifier enhanced by Retrieval-Augmented Generation (RAG). Unlike Prompt 1, it integrates retrieved domain-specific examples (denoted as $context$) into the instruction, thereby mitigating hallucinations and improving classification robustness in high-level or abstract task scenarios.

Prompt 1 - Classifier with General-purpose LLM

You are a classifier that can categorize questions into specific tasks. Your job is to analyze the following given question and determine which task from the provided list it most likely belongs to.
The tasks are as follows: $\{task\ list\}$.
The question is:
"$\{question\}$
A.$\{option_A\}$
B.$\{option_B\}$
C.$\{option_C\}$
D.$\{option_D\}$".
Provide your answer in the format: "**Task: $[selected\ task]$ **".

Prompt 2 - AIOps Experts with Chain of thought

Please answer the following DEVOPS question.
The question is: $\{question\}$
The options are as follows:
A. $\{option_A\}$
B. $\{option_B\}$
C. $\{option_C\}$
D. $\{option_D\}$
Think step by step and then finish your answer with "the answer is (X)" where X is the correct letter choice.

Prompt 3 - Classifier with RAG

You are a classifier that can categorize questions into specific tasks. Your job is to analyze the following given question and determine which task from the provided list it most likely belongs to.
The tasks are as follows: $\{task\ list\}$.
The question is:
"$\{question\}$
A.$\{option_A\}$
B.$\{option_B\}$
C.$\{option_C\}$
D.$\{option_D\}$".
You can refer to the following examples of questions and their corresponding tasks to decide the current question's task: $\{context\}$
Provide your answer in the format: "**Task: $[selected\ task]$ **".

B Datasets Information

We evaluate our framework using the DevOps-Eval benchmark. As outlined in Tab. 4, the dataset is stratified into two linguistic subsets, each corresponding to a distinct level of task abstraction. The English subset (Task Set A) comprises 1,420 samples focusing on *low-level AIOps tasks*, such as Log Parsing and Time Series Anomaly Detection, which require specific algorithmic reasoning. Conversely, the Chinese subset (Task Set B) encompasses 4,557 samples covering the comprehensive *high-level DevOps lifecycle*, ranging from Planning to Operation. This division allows us to assess the framework's adaptability across both concrete, data-centric tasks and abstract, process-oriented

Category	Task	Sample
English[a]	Log Parser	350
	Root Cause Analysis	250
	Time Series Anomaly Detection	300
	Time Series Classification	200
	Time Series Forecasting	320
Chinese[b]	Build	218
	Code	1321
	Deploy	255
	Monitor	216
	Operate	2041
	Plan	66
	Release	212
	Test	228

[a]Can be treated as "dataset with low-level tasks".
[b]Can be treated as "dataset with high-level tasks".

Table 4: Dataset information of DevOps-Eval.

scenarios.

C Implementation Details

To ensure the reproducibility of our experiments and provide transparency regarding the resource requirements, we present the detailed Task-Expert mapping and model configurations in Tab. 5.

Expert Model Selection We curated four distinct Expert Sets to evaluate the framework's adaptability across different model scales and deployment constraints.

- **Local Deployment (Sets 1 & 3):** For resource-constrained scenarios, we selected lightweight open-source models (e.g., *InternLM-7B*, *Qwen2-7B-Instruct*) deployed locally. Expert Set 1 focuses on low-level English AIOps tasks (Part I), while Expert Set 3 covers high-level Chinese DevOps tasks (Part II).
- **API Access (Sets 2 & 4):** To assess performance with stronger foundational capabilities, we utilized larger-scale or closed-source models (e.g., *DeepSeek-V3*, *Gemma-2-27B*) accessed via external APIs (OpenRouter or o3.fan). This setup allows us to benchmark the upper limits of our CoE-Ops framework without local hardware limitations.

Classifier Configuration As specified at the bottom of Tab. 5, we employed two distinct backbones for the task classifier to verify scalability: **Classifier 1** (DeepSeek-R1-Distill-Qwen-7B, local) and **Classifier 2** (DeepSeek-V3, API). Both classifiers were evaluated under two conditions: standard zero-shot classification and our proposed RAG-enhanced setting, to demonstrate the impact of contextual retrieval on routing accuracy.

D Radar Charts

To facilitate a holistic understanding of model competencies, this section presents the capability radar charts derived from our experimental evaluations. These visualizations map the performance distribution of CoE-Ops against individual expert models and baseline frameworks (e.g., MoE) across the specific task dimensions defined in Tab. 4. By projecting accuracy metrics onto a multi-axial plane, these charts vividly demonstrate the balanced capability landscape of CoE-Ops, contrasting it with the uneven or domain-skewed performance profiles often exhibited by standalone experts. This visual evidence further corroborates the effectiveness of our collaborative routing mechanism in bridging capability gaps and ensuring robust performance across the comprehensive DevOps lifecycle.

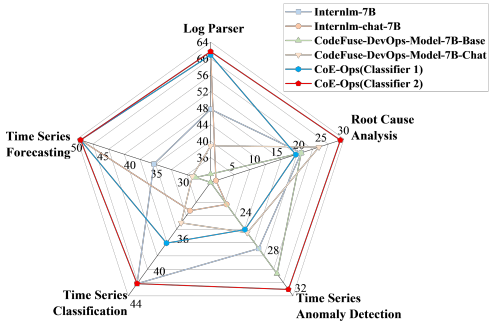


Figure 5: Capability Radar Chart of CoE-Ops with Expert Set 1 on DevOps-EVAL English (TASK SET A)

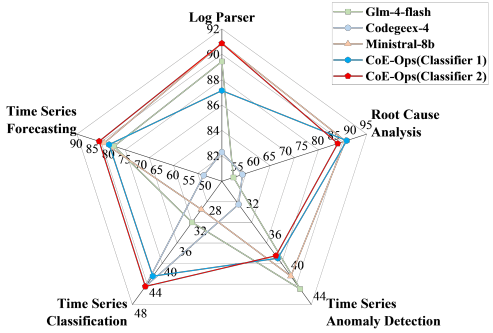


Figure 6: Capability Radar Chart of CoE-Ops with Expert Set 2 on DevOps-EVAL English (TASK SET A)

Part I: Task Set A (English)		
Task Name	**Expert Set 1**	**Expert Set 2**
Log Parser	Internlm-chat-7B[1]	Ministral-8b[2]
Root Cause Analysis	CodeFuse-DevOps-Model-7B-Chat[1]	Ministral-8b
Time Series Anomaly Detection	CodeFuse-DevOps-Model-7B-Base[1]	Glm-4-flash[3]
Time Series Classification	Internlm-7B[1]	Codegeex-4[3]
Time Series Forecasting	Internlm-chat-7B	Ministral-8b
Part II: Task Set B (Chinese)		
Task Name	**Expert Set 3**	**Expert Set 4**
Build	Internlm-chat-7B	Gemma-2-27b-it[2]
Code	Qwen2-7B-Instruction[1]	Doubao-1.5-lite[3]
Deploy	Internlm-chat-7B	Doubao-1.5-lite
Monitor	Mathstral-7B-v0.1[1]	Gemma-2-27b-it
Operate	Qwen2-7B-Instruction	Gemma-2-27b-it
Plan	Qwen2-7B-Instruction	Glm-4-flash[3]
Release	Mathstral-7B-v0.1	Gemma-2-27b-it
Test	Qwen2-7B-Instruction	Doubao-1.5-lite
Classifier Settings	**Model Configuration**	
Classifier 1	DeepSeek-R1-Distill-Qwen-7B[1] (with/without RAG)	
Classifier 2	DeepSeek-V3[2] (with/without RAG)	

[1] Deployed locally.
[2] Deployed via API (OpenRouter).
[3] Deployed via API (o3.fan).

Table 5: Task-Expert Mapping and Classifier Settings for DevOps-Eval Task Sets.

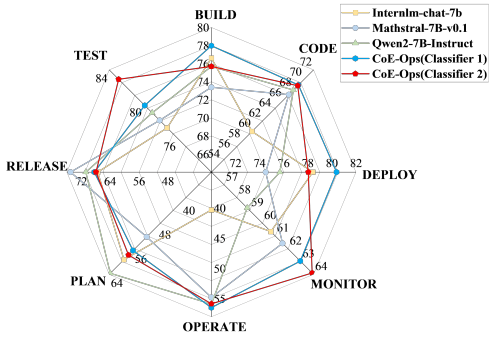


Figure 7: Capability Radar Chart of CoE-Ops with Expert Set 3 on DevOps-EVAL Chinese (TASK SET B)

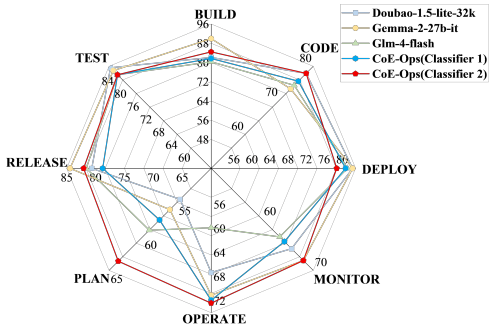


Figure 8: Capability Radar Chart of CoE-Ops with Expert Set 4 on DevOps-EVAL Chinese (TASK SET B)

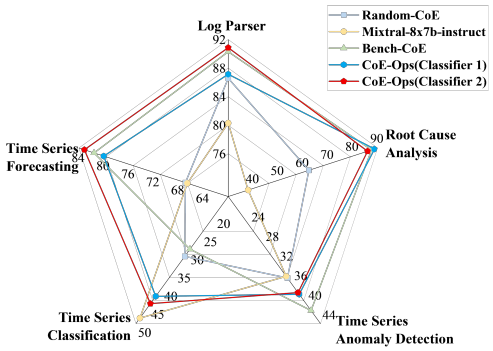


Figure 9: Capability Radar Chart of Comparative Experiments on DevOps-EVAL English (Task Set A)

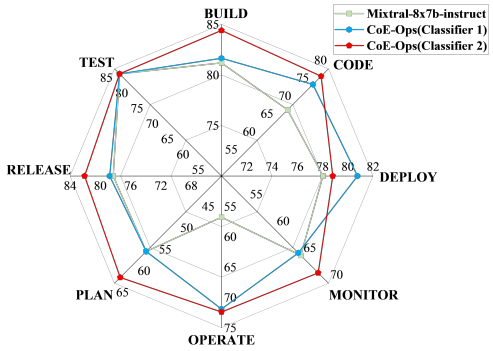


Figure 10: Capability Radar Chart of Comparative Experiments on DevOps-EVAL Chinese (Task Set B)