# Adaptive Orchestration of Heterogeneous LLMs with Dynamic Escalation for Multi-Agent Collaboration

Bipin Bohara[1](bbdzv@mst.edu), Hao Wang[2] (hwang9@stevens.edu), Praveenkumar Kondikoppa[3](pkondikoppa@google.com), Seung-Jong Park[1](seung-jong.park@mst.edu)

[1]Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA
[2]Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA
[3]Google LLC, Mountain View, CA, USA

## Motivation

Resource-constrained multi-agent systems need to coordinate heterogeneous LLMs efficiently without relying on large models for every task [2].

- Large models are costly and impractical for all requests.
- Small models are underutilized despite capable of handling many low-risk tasks.
- Static model assignment cannot adapt to increasing task difficulty or agent divergence.
- Adaptive heterogeneous orchestration across agents and rounds remains underexplored.

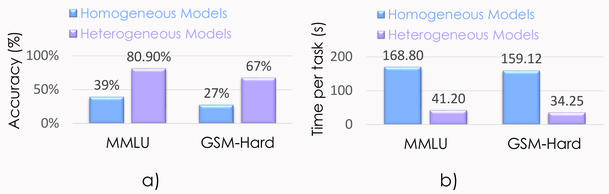



Figure 1: Comparison of homogeneous vs heterogeneous model-agents for a small sample of MMLU and GSM-Hard benchmarks, a) Accuracy, and b) Average time (in seconds) per task using both approach

Figure takeaway:

- **Figure 1:** Heterogeneous models improve accuracy while also reducing execution time by 75.59% on MMLU and 78.47% on GSM-Hard
- **Figure 2:** Concept: combining small and large models enables better task allocation, lower resource usage, and reduced dependence on large LLMs.

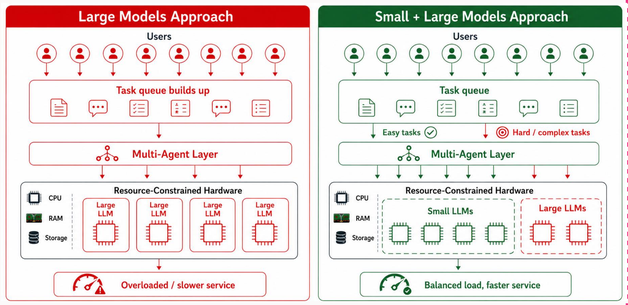



Figure 2: Concept of Large and Small + Large model architectures under resource-constrained hardware.

## Objectives

This work develops a resource-efficient orchestration framework for heterogeneous LLM agents with the following objectives:

- Design a two-orchestrator framework:
  - **O-1** selects expert agents and the initial model tier;
  - **O-2** monitors execution and escalates capability when needed.
- Coordinate small and large models efficiently under resource constraints.
- Maintain accuracy while reducing cost and execution time.
- Evaluate performance across benchmarks using accuracy, cost, execution time, and escalation frequency.

## Design

Multi-Agent Collaboration Framework[1]

- **Agent Type:** Task-specific experts assigned via lightweight SLM
- **Collaboration:** Multi-round exchange and refinement
- **Discussion Paradigm:** Memory, Relay, Report, or Debate
- **Decision Protocol:** Consensus, Voting, or Judge

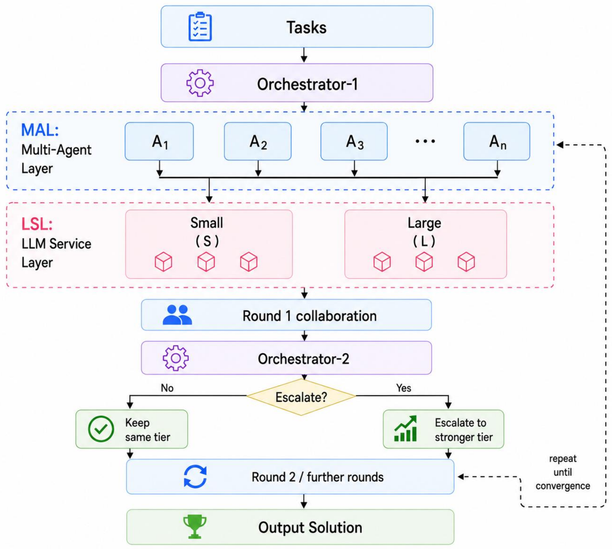



Figure 3: System Architecture

Adaptive Two-Orchestrator Framework

**O-1:** Initial Assignment

- Evaluates task-level risk.
- Assigns the initial model tier.
- Low risk → small models; high risk → stronger models.

**O-2:** Runtime Adaptation

- Monitors collaboration signals.
- Detects divergence or insufficient capability.
- Maintains or escalates the model tier.

Overall: O-1 allocates resources initially; O-2 adapts them during execution.

## Implementation Details

Table 1: Hardware configuration of the Kubernetes cluster used for experimental evaluation.

| | |
|---|---|
| **Configuration** | Kubernetes Cluster deployed on 1 Head node, 9 compute nodes with specifications as below |
| **Compute Node** | CPU: 2 × Intel Xeon Gold 6548Y+, 32 cores/64 threads each<br>Mem: 16 × 32 GB DDR5-5600 RDIMMs; 512 GB each<br>GPUs: 4x NVIDIA L40S PCIe GPUs |
| **Head Node** | CPU: 2 × Intel Xeon Gold 6428N, 32 cores/64 threads each<br>Mem: 16 × 64 GB DDR5-5600 RDIMMs; 1,024 GB total |

Table 2: Model deployment configurations for the small and large language models, all models are served using vLLM [3]

| Models | Deployment Configurations |
|---|---|
| gpt-oss-20b<br>NVIDIA-Nemotron-3-Nano-30B-A3B-FP8<br>Qwen3.5-35B-A3B-FP8 | CPU: 1x Intel Xeon Gold 6548Y+, 32 cores/64 threads<br>Mem: 256 GB<br>GPU: 2x NVIDIA L40S GPUs |
| gpt-oss-120b<br>NVIDIA-Nemotron-3-Super-120B-A12B-FP8<br>Qwen3.5-122B-A10B-FP8 | CPU: 1 entire compute node<br>Mem: 1 entire compute node<br>GPU: 4x NVIDIA L40S GPUs |
| Ministral-3-3B-Reasoning-2512 | CPU: 1x Intel Xeon Gold 6548Y+, 16 cores/42 threads<br>Mem: 128 GB<br>GPU: 1x NVIDIA L40S GPU |

## Results

We evaluate the proposed orchestration framework by comparing **Large-only** and **Small + Large** configurations across five datasets in terms of:

1. **Task Allocation:** Tasks handled by small models, large models and escalated during execution
2. **Accuracy:** Task performance across both configurations
3. **Execution time:** End-to-end task completion time
4. **Computational Cost:** Resource usage in terms of $

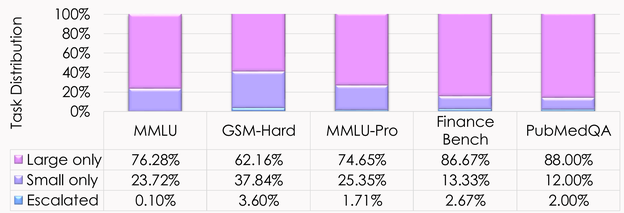



Figure 4: Distribution of task allocations across small and large LLMs, distinguishing tasks directly assigned by O-1 from those escalated by O-2

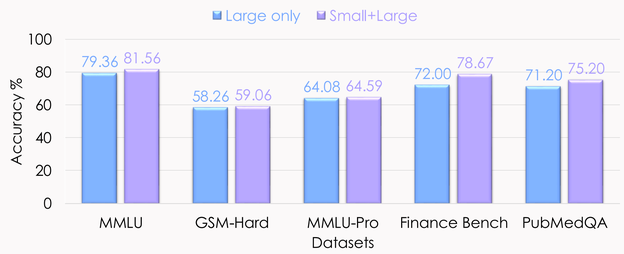



Figure 5: Performance comparison between the Large-only and Small + Large LLM configurations across benchmarks

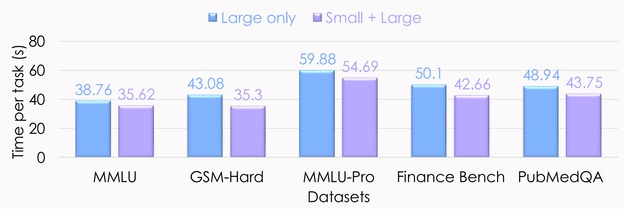



Figure 6: Operation time (in seconds) comparison for the Large-only and Small + Large LLM configurations; operation time also includes SLMs execution time

Table 3: Comparison and Reduction of cost (in $) between the Large-only and Small + Large LLM configurations

| Datasets | Large only | Small + Large | Cost reduction % |
|---|---|---|---|
| MMLU | 0.3585 | 0.2902 | 19.06% |
| GSM-Hard | 0.3985 | 0.2644 | 33.65% |
| MMLU-Pro | 0.5539 | 0.4413 | 20.32% |
| Finance Bench | 0.4634 | 0.3682 | 20.55% |
| PubMedQA | 0.4526 | 0.3802 | 15.99% |

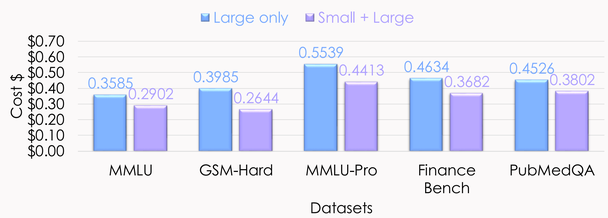



Figure 7: Cost ($) comparison between the Large-only and Small + Large LLM configurations.

## Conclusion

This study demonstrates that adaptive orchestration can improve efficiency while maintaining strong performance.

- **Accuracy** improved by **0.51%-6.67%** over Large-only baseline
- **Cost** reduced by **15.99%-33.65%**
- **Execution time** reduced by **8.1%-18.06%**
- **Escalation**: Only **0.1%-3.6%** of tasks required escalation

Heterogeneous LLMs can be coordinated efficiently and adaptively without sacrificing performance.

## References

1. Jonas Becker, Lars Benedikt Kaesberg, Niklas Bauer, Jan Philip Wahle, Terry Ruas, and Bela Gipp. 2025. MALLM: Multi-Agent Large Language Models Framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 418–439, Suzhou, China. Association for Computational Linguistics.
2. Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In Proceedings of the 41st International Conference on Machine Learning (ICML'24), Vol. 235. JMLR.org, Article 467, 11733–11763.
3. Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In Proceedings of the 29th Symposium on Operating Systems Principles (SOSP '23). Association for Computing Machinery, New York, NY, USA, 611–626. https://doi.org/10.1145/3600006.3613165

## Acknowledgements

This material is based upon work partially supported by the National Science Foundation under Grant No. #2403248 and #2403399.