Artificial Intelligence and Computing Innovations

Intelligence Science Publishing

https://www.ispublishing.com

ISSN: 3070-4944

Vo.5, No.4, 2025

Learning-Driven Collaborative Scheduling for Multi-Tenant Distributed Inference Services

Zhibin Huang[1], Jason Luo[2], Changfan Chen[3*]

[1]Boston University, Boston, USA

[2]University of Kansas, Lawrence, USA

[3]Johns Hopkins University, Baltimore, USA

**Corresponding author: cchen50415@gmail.com*

Abstract: To address the scheduling complexity of multi-tenant distributed inference services under shared resources, model heterogeneity, and dynamic request conditions, this paper proposes a learning driven collaborative scheduling mechanism that models inference scheduling from a holistic system perspective. The approach abstracts system states and formulates scheduling decisions as a learnable mapping process. This enables the scheduling policy to dynamically adapt to workload variations and resource conditions during continuous operation. To capture intrinsic relationships in multi-tenant concurrent scenarios, collaborative utility modeling is incorporated into the scheduling objective. This guides scheduling behavior toward coordinated constraints among resource utilization efficiency, service stability, and tenant fairness. The proposed framework avoids strong dependence on specific execution details and provides a unified system-level treatment of request assignment and resource scheduling. It mitigates performance degradation caused by resource contention and workload fluctuation in multi-tenant parallel inference. System-level evaluation in multi-tenant distributed inference scenarios demonstrates that the method achieves strong overall performance in service response, system throughput, operational stability, and tenant fairness. These results confirm the effectiveness and practicality of learning driven collaborative scheduling in complex inference service environments.

Keywords: Multi-tenant inference services; Collaborative scheduling mechanism; Learning driven decision making; Distributed system optimization

1. Introduction

As deep learning models are widely deployed in intelligent applications, inference services have gradually evolved from isolated single model execution to distributed services that support multiple applications and concurrent users. In practical systems, different applications share the same inference

infrastructure in the form of tenants. Their request volumes, model complexities, and temporal patterns vary significantly. While such a multi-tenant environment improves overall resource utilization, it also greatly increases scheduling complexity. Inference services, therefore, become not only a problem of model computation, but also a comprehensive system issue involving resource management, quality of service assurance, and operational stability [1]. Under these conditions, coordinating inference demands from different tenants on shared computing resources becomes a fundamental challenge for large-scale intelligent services.

From a system perspective, the core difficulty of multi-tenant distributed inference lies in the persistent coexistence of resource contention and heterogeneous demands. Models owned by different tenants differ in architectural scale, computational intensity, and memory access behavior. Their request arrival processes are also highly stochastic and often bursty. Such heterogeneity makes it difficult for unified and static scheduling strategies to balance global efficiency and individual service quality. These strategies may perform well under specific workloads but often lead to performance fluctuations or degradation when conditions change. Moreover, scheduling decisions in multi-tenant systems are strongly coupled. The scheduling behavior of one tenant can affect the latency, throughput, or resource usage of others. This interdependence further complicates system-level optimization.

Most existing inference scheduling approaches rely on predefined rules or heuristic policies. They control execution by manually setting model priorities, batch sizes, or resource quotas. These methods are effective in stable environments with limited model diversity. However, their designs are usually based on simplified assumptions and cannot accurately represent the dynamic nature of real systems. When request patterns, resource availability, or tenant composition change, fixed policies often lack adaptability. This results in inefficient resource usage or imbalanced service quality. More importantly, such approaches typically separate model execution from system scheduling. They ignore potential collaboration among inference tasks, which limits further improvement of overall system performance.

With the growing adoption of learning based methods in system optimization, data-driven scheduling has emerged as a promising direction. Learning driven schedulers can continuously observe system states, historical behavior, and runtime feedback. They can gradually form implicit models of complex environments and make more targeted decisions under dynamic conditions [2]. In multi-tenant distributed inference services, learning mechanisms can reduce the limitations of manual rule design. They also provide new opportunities to discover cooperative behavior across models and tenants. By formulating scheduling as a learnable decision process, the system can adapt its strategy over time to uncertain workloads and resource conditions.

Against this background, studying learning driven collaborative scheduling for multi-tenant distributed inference services is of both theoretical and practical importance. On the one hand, it supports a shift from model-centric local optimization to global coordination based on overall system performance [3, 4]. This is essential for building efficient, stable, and scalable intelligent service infrastructures. On the other hand, exploring how learning methods can be applied to complex scheduling scenarios helps clarify their design principles and limitations. Such insights offer general guidance for the intelligent evolution of backend inference systems and are relevant to cloud inference, edge intelligence, and large-scale online services.

2. Related Work

Recent studies on deep learning inference services have mainly focused on system efficiency and quality of service assurance. One line of work targets the inference execution pipeline. It reduces computation and memory overhead per inference through model compression, operator fusion, low-precision computation, and hardware-aware optimization. These approaches can effectively shorten

response time or improve throughput in scenarios with a single model or a small number of models. However, their optimization objectives are usually limited to internal model execution paths [5]. They pay limited attention to system behavior under multi-model parallel execution and multi-tenant resource sharing. When inference services are deployed as shared platforms for multiple applications, model-level optimization alone is insufficient to address system-level issues such as request contention, workload fluctuation, and resource allocation conflicts.

Another line of research approaches the problem from a system scheduling perspective. It investigates deployment and resource management strategies for inference services in distributed environments. These methods typically involve request queuing, dynamic batching, resource partitioning, and priority control. They aim to improve system throughput or reduce service latency under constrained resources. Existing studies show that well-designed scheduling strategies can partially alleviate performance degradation caused by resource contention. However, most of these methods rely on manually defined rules or heuristic policies [6]. Their decision logic is often based on static assumptions and cannot fully capture the complex and continuously changing operating conditions of multi-tenant systems. When request patterns or tenant compositions change, fixed strategies often lack adaptability. This leads to unstable performance or imbalanced service quality among tenants.

With the increasing adoption of data-driven approaches in system optimization, some studies have begun to introduce learning mechanisms for scheduling decisions. These methods model system states, historical workloads, and scheduling outcomes. They use learning models to automatically adjust scheduling behavior in response to dynamic environments. Compared with rule-based approaches, learning driven scheduling shows advantages in handling high-dimensional state spaces and complex decision relationships. It can capture implicit patterns without explicit system modeling. However, existing work mostly focuses on single-objective or single-task settings. Inference requests are often treated as homogeneous workloads. Differences and interactions among models and tenants are insufficiently considered. The collaborative potential under multi-model parallel execution has not been fully explored.

Overall, existing studies have made progress in inference execution optimization, system scheduling design, and the integration of learning methods. Nevertheless, clear limitations remain in the complex setting of multi-tenant distributed inference services. On one hand, model-level optimization and system-level scheduling lack a unified perspective [7]. This prevents the formation of collaborative mechanisms oriented toward overall system quality. On the other hand, current learning driven approaches do not sufficiently characterize long-term resource competition and cooperative behavior in multi-tenant environments. How to coordinate inference processes across tenants and models under shared resources through learning based methods remains an important open research problem for achieving stable, efficient, and fair system operation.

3. Proposed Framework

3.1 Overall Framework Description

A multi-tenant distributed inference service can be viewed as a dynamic system composed of several service nodes and multiple concurrent tenants. During continuous operation, the system constantly receives inference requests from different tenants and needs to schedule and execute these requests under shared resource conditions. The overall model architecture is shown in Figure 1.

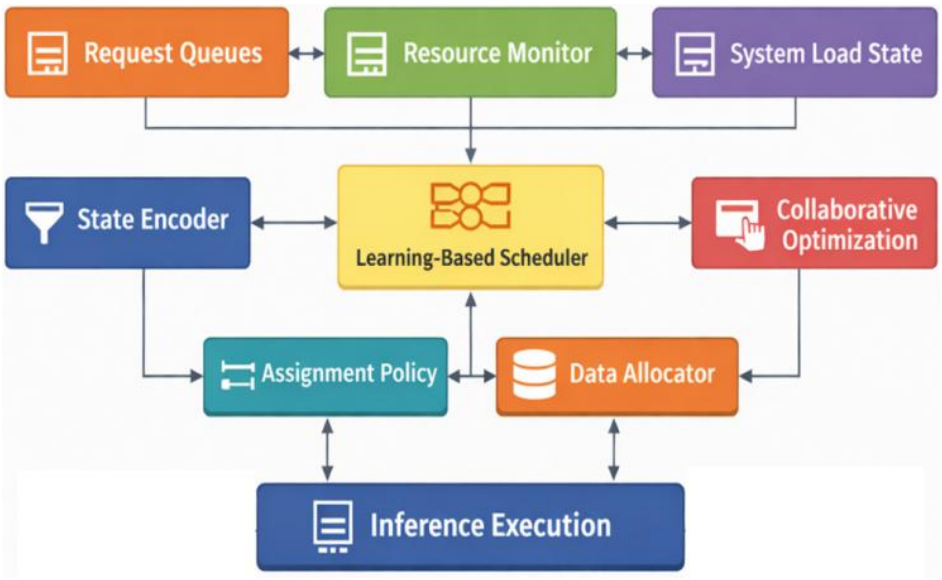

Figure 1. Overview of the proposed method

To uniformly describe the system's operation, a global runtime description vector at time t is introduced:

$$x_t = [d_t, c_t, l_t] \tag{1}$$

Where d_t represents the demand intensity distribution among tenants, c_t represents the current allocable computing capacity status, and l_t represents the load level within the system. At each time step, the scheduler generates a scheduling map x_t based on y_t , which characterizes the allocation relationship and execution ratio of requests to nodes. This modeling approach avoids explicitly depicting specific execution details, instead abstracting the scheduling decision space at the system level.

3.2 Learning-Driven Scheduling Mapping Function

To address the challenges of frequent and accurate state changes in multi-tenant environments, this paper employs a learnable mapping function to describe the scheduling process. The scheduling mapping is defined as follows:

$$y_t = F_w(x_t) \tag{2}$$

Here, F_w is a decision function with parameter w, used to map the system state to an executable scheduling scheme. To measure the long-term effectiveness of scheduling behavior, a time-averaged performance function is introduced:

$$J(w) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} h(x_t, y_t) \tag{3}$$

Where $h(\cdot)$ represents the system's operational quality evaluation function at a single moment. This form does not rely on explicit phase reward assumptions, but rather characterizes the performance of the scheduling strategy in the long-term service process from the perspective of overall operation, making the learning objective closer to the operational characteristics of actual inference services.

3.3 Construction of a Collaborative Utility for Multi-Tenants

In multi-tenant scenarios, the scheduling behaviors of different tenants are not independent; their resource consumption and execution rhythm are implicitly related. To characterize this relationship, a tenant-level load response function is introduced:

$$\mu_i(t) = \varphi_i(y_t) - \rho_i(c_t) \tag{4}$$

Where $\varphi_i(\cdot)$ represents the positive impact of the scheduling result on the service capacity of tenant i, and $\rho_i(\cdot)$ represents the constraint cost caused by resource scarcity. The overall system coordination goal is defined as follows:

$$h(x_t, y_t) = \sum_{i=1}^{M} w_i \mu_i(t) \tag{5}$$

Where w_i is the tenant weight coefficient. This design introduces cross-tenant aggregation at the objective function level, giving the scheduling process a natural cooperative optimization characteristic, thereby avoiding resource contention and performance imbalance problems that may occur from a single-tenant perspective.

3.4 Online Adaptive Update and Constraint Mechanism

To ensure that the scheduling strategy can be continuously adjusted according to changes in the system's operating state, this paper adopts an online update method to iteratively correct parameter w, and the update form is expressed as follows:

$$w^{(k+1)} = w^{(k)} + \lambda \Delta w^{(k)} \tag{6}$$

Where λ is the update step size, and $\Delta w^{(k)}$ represents the adjustment direction calculated based on the current running feedback. To prevent drastic changes in scheduling behavior between adjacent stages, a consistency constraint is introduced:

$$R = ||F_{w^{(k+1)}}(x_t) - F_{w^{(k)}}(x_t)||_2 \tag{7}$$

Furthermore, constraints and controls are applied during the update process. This mechanism enables the scheduling strategy to maintain its adaptive capabilities while preserving the continuity and stability of system operation, making it suitable for long-term multi-tenant distributed inference service scenarios.

4. Experimental Analysis

4.1 Dataset

Figures This study uses inference service request trace data from the Azure Public Dataset as the experimental data source. The dataset is collected from large-scale online inference services in real cloud environments. It records request arrival behavior and system operating states under shared computing resources among multiple tenants. The data has strong practical relevance.

The dataset covers inference request traces across multiple time spans. Individual requests differ significantly in arrival frequency, computational scale, and execution duration. This reflects typical workload heterogeneity in multi-tenant environments. By organizing and restructuring request sequences, it is possible to construct multi-tenant request queues, resource occupancy states, and system load evolution processes. These elements provide fundamental inputs for state modeling in scheduling strategies. The presence of bursty requests and long-term workload variations makes the dataset suitable for analyzing the adaptability of learning driven scheduling mechanisms in dynamic environments.

Based on this dataset, scheduling-related statistical features can be further extracted. These include request arrival intensity, changes in concurrency levels, and trends in resource usage. Such features support the learning and updating of scheduling policies under long-term operation. Since the data originates from real cloud inference services, its workload distributions and request patterns are highly realistic. This helps assess the applicability of the proposed scheduling mechanism in multi-tenant distributed inference services and provides a reliable data foundation for studying collaborative scheduling and system stability.

4.2 Experimental Results

This article first presents the results of the comparative experiments, as shown in Table 1. Specifically, Average Response Time and Tail Latency (P99) are reported in milliseconds (ms). Throughput is measured as requests per second (req/s) over the evaluation window. The Tenant Fairness Index is a unitless score in [0,1], where values closer to 1 indicate a more balanced allocation of service quality across tenants.

Table 1. Comparative experimental results

Method	Average Response Time	Tail Latency(P99)	Throughput	Tenant Fairness Index
Symphony [8]	182	410	915	0.82
TARA [9]	165	380	960	0.85
MtDB [10]	158	355	995	0.87
DSGTA [11]	149	332	1040	0.89
ServiceNet [12]	143	318	1075	0.91
Ours	129	276	1158	0.95

From a global perspective, all methods exhibit different levels of performance bottlenecks in multi-tenant distributed inference scenarios, whereas the learning driven collaborative scheduling mechanism maintains more stable service quality under complex workloads. Through continuous modeling of system states and online policy updates, the proposed approach achieves smoother execution behavior under normal request conditions. This leads to a further reduction in average response time and creates a clear contrast with static or semi-static scheduling strategies. The observation indicates that in environments with concurrent competition among tenants, predefined rules alone are insufficient to track system dynamics. Learning based scheduling can more effectively exploit state information to adjust execution pacing.

Under high load or resource-constrained conditions, tail latency provides a more direct indication of scheduling robustness. The results show that learning driven scheduling achieves better convergence when handling extreme requests. This behavior is closely related to the decision smoothing mechanism and the collaborative optimization objective embedded in the approach. By preserving behavioral continuity between scheduling updates, the system avoids sharp oscillations when facing bursty requests or changes in tenant behavior. As a result, the common amplification of long tail latency in multi-tenant settings is effectively mitigated.

From the perspective of resource utilization, throughput reflects how well a scheduling strategy mobilizes overall system processing capacity. Compared with methods that rely on static priorities or fixed allocation rules, learning driven scheduling more fully unlocks latent parallelism. It leads to a more balanced workload distribution across nodes. The results indicate that once cross-tenant collaborative relationships are identified, the model can automatically adjust resource usage patterns. This reduces resource fragmentation, shortens execution paths, and ultimately increases effective output over sustained operation periods.

Fairness is a critical consideration in multi-tenant environments. The collaborative scheduling mechanism incorporates tenant-level utility terms into the optimization objective. This enables finer-grained control over service differentiation among tenants. The results demonstrate that the system can preserve service stability for individual tenants while optimizing overall efficiency. It prevents scenarios where specific tenants suffer long-term degradation. This balance between global performance and tenant experience suggests that collaborative scheduling not only targets performance improvement but also supports sustainability and availability in real-world multi-tenant inference deployments.

The policy smoothing coefficient controls the magnitude of scheduling changes between adjacent decision stages. Its configuration affects decision continuity and the pacing of resource allocation under dynamic operating conditions. In multi-tenant distributed inference scenarios, uncertainty in request arrivals, differences in model workloads, and competition for shared resources jointly determine its role as a key factor balancing efficiency and stability. To examine how scheduling behavior varies under different levels of smoothing constraints, the following visualization compares performance across discrete parameter settings and characterizes the overall response of policy outputs as the smoothing coefficient changes, and the experimental results are shown in Figure 2.

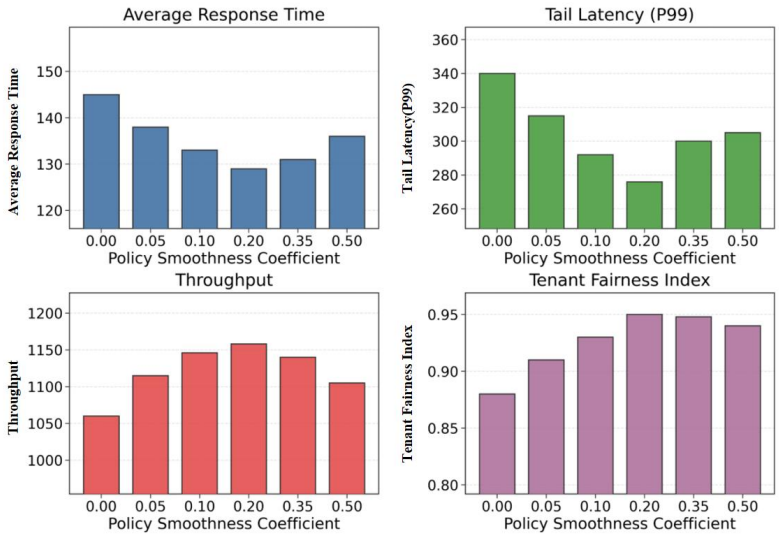

Figure 2. Key performance behavior of the learning driven collaborative scheduling mechanism in multi-tenant distributed inference scenarios under different policy smoothing coefficient settings

From the overall trend, the policy smoothing coefficient plays a significant role in regulating the continuity of scheduling decisions. As the smoothing constraint is gradually strengthened, scheduling behavior becomes more coherent across adjacent decision stages. This reduces additional waiting time and scheduling overhead caused by frequent adjustments and allows the system to maintain a more stable service rhythm under normal operating conditions. The introduction of continuity prevents the scheduler from reacting excessively to instantaneous state fluctuations and better matches the long-term operating characteristics of multi-tenant inference services.

When the system encounters high load or request variability, changes in tail latency further reveal the impact of the smoothing coefficient on robustness. An appropriate level of smoothing suppresses sharp oscillations in scheduling decisions under local state disturbances. This leads to a more orderly resource allocation process and alleviates the accumulation of extreme requests in queues. When the smoothing strength is too low or too high, the scheduler either becomes overly aggressive or responds too slowly. Both cases can amplify tail latency. This indicates that the smoothing coefficient must be carefully balanced between stability and responsiveness.

From the perspective of overall system processing capacity, throughput reflects how effectively the scheduling strategy activates resource parallelism. As the smoothing coefficient changes, a clear trade-off emerges between resource utilization efficiency and scheduling flexibility. A moderate smoothing constraint helps avoid frequent reallocations while preserving sufficient decision freedom. This enables the system to continuously release effective computing capacity and sustain high overall throughput. Such behavior aligns with the design goal of collaborative scheduling that emphasizes global efficiency.

Fairness, which is particularly important in multi-tenant environments, is also influenced by variations in the smoothing coefficient. With a well-chosen smoothing mechanism, differences in service quality among tenants are effectively constrained. Scheduling decisions no longer remain biased toward specific tenants due to short-term workload changes. This observation shows that policy smoothing not only improves system stability but also implicitly reinforces cross-tenant coordination. It allows the scheduling mechanism to enhance efficiency while maintaining fairness and sustainability in multi-tenant inference services.

The scheduling decision frequency determines the temporal granularity at which the system updates its scheduling policy during operation. Variations in this setting directly affect how scheduling behavior responds to fluctuations in system state. In multi-tenant distributed inference services, decision updates that are too frequent or too sparse can both disrupt the rhythm of resource allocation and undermine operational stability. To characterize the behavior of the scheduling mechanism under different decision frequency configurations, the following visualization analyzes discrete frequency settings and illustrates the overall performance response as the scheduling update cadence changes, and the experimental results are shown in Figure 3.

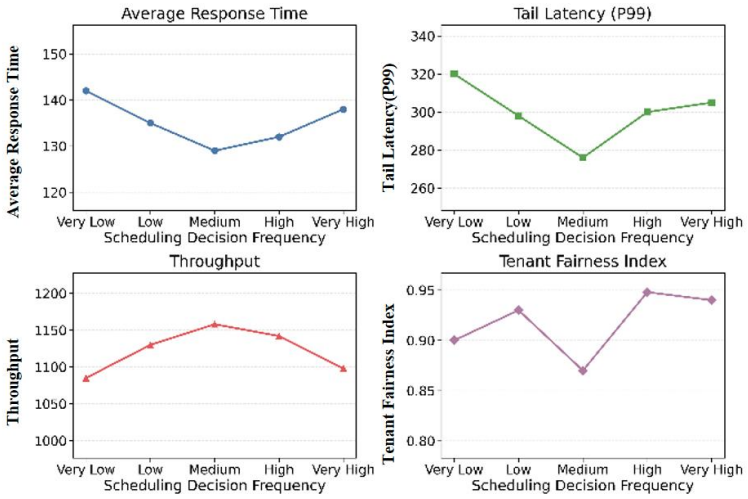

Figure 3. Performance stability variations of the learning driven collaborative scheduling mechanism in multi-tenant distributed inference systems under different scheduling decision frequency settings

In Figure 3, "Very low" corresponds to updating the scheduling decision every 10 seconds, "low" corresponds to updating every 5 seconds, "medium" corresponds to updating every 2 seconds, "high" corresponds to updating every 1 second, and "very high" corresponds to updating every 0.5 seconds. These settings all indicate that a scheduling policy update is triggered at a fixed period during operation, thereby controlling the response strength and resource allocation stability to system state fluctuations at different time granularities.

Overall behavior indicates that scheduling decision frequency directly shapes how finely the system perceives state variations and how quickly it adjusts its actions. When decision updates are too sparse, the scheduling policy reacts with noticeable delay. It cannot promptly reflect changes in the queue backlog or resource occupancy. As a result, the service process exhibits strong inertia. Under this setting, the system tends to preserve existing allocation patterns and cannot fully exploit the adaptive capability of learning driven scheduling in dynamic environments.

As the decision frequency increases, the system updates scheduling actions at a finer temporal scale. Resource allocation and request handling become more aligned with the current operating state. At this

stage, the scheduling policy achieves a more appropriate balance between continuity and flexibility. It helps suppress performance fluctuations caused by short-term workload variations and allows the system to remain relatively stable under multi-tenant concurrency. This behavior is consistent with the design objective of collaborative scheduling that emphasizes global state modeling and online adjustment.

When the decision frequency is further increased, scheduling updates become excessively frequent, and the system responds more sensitively to state changes. Although such a configuration enables rapid resource reallocation, it may also introduce additional scheduling disturbances. Execution paths can fluctuate more strongly between adjacent stages. Frequent policy updates weaken decision continuity and can negatively affect system stability. This outcome highlights the potential cost of overly high decision frequency in multi-tenant environments.

From the perspective of multi-tenant fairness, variations in decision frequency also influence service isolation and coordination. A moderate update frequency helps maintain a balanced service rhythm across tenants. It prevents resource allocation from repeatedly shifting due to transient state changes. Overall, this sensitivity analysis shows that scheduling decision frequency must be aligned with workload dynamics and the update capacity of learning driven schedulers. Only under such alignment can a reasonable trade-off be achieved among efficiency, stability, and multi-tenant coordination.

The resource supply ratio describes the relative relationship between available schedulable computing capacity and request workload during system operation. Changes in this ratio directly affect the decision space of the scheduling policy and the flexibility of resource allocation. In multi-tenant distributed inference environments, different levels of resource supply correspond to different degrees of contention and execution constraints. As a result, the scheduling mechanism operates under substantially different system conditions. To analyze the behavior of the learning driven collaborative scheduler under varying resource availability, the following visualization examines performance under discrete resource supply configurations and illustrates the overall response trend as the resource ratio changes, and the experimental results are shown in Figure 4.

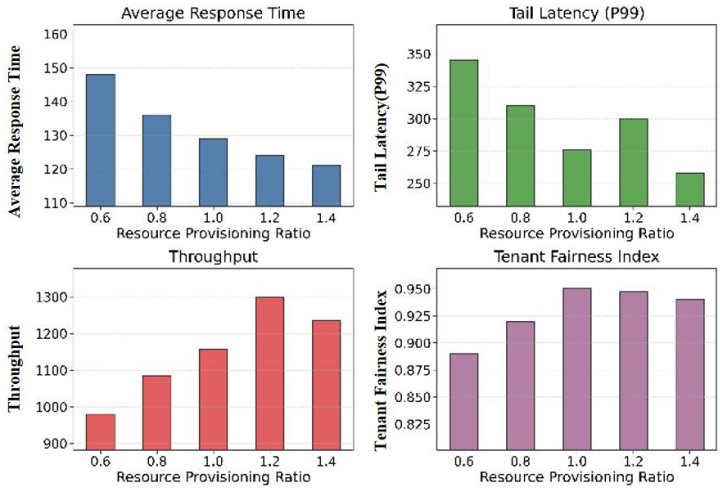

Figure 4. Overall performance variation trends of the learning driven collaborative scheduling mechanism in multi-tenant distributed inference systems under different resource supply ratio conditions

From the observed patterns, the resource supply ratio directly defines the schedulable space of the system in a shared environment. As resources become more abundant, the scheduling policy can optimize request execution paths over a wider range. This alleviates queuing and contention caused by resource scarcity. With higher supply levels, the scheduler gains greater control over system dynamics. Request

processing becomes smoother. This behavior aligns with the design goal of learning driven scheduling that dynamically adjusts resource allocation through global state awareness.

At the level of response time and tail latency, changes in the resource supply ratio have a pronounced impact on system stability. When supply is constrained, even collaborative scheduling remains limited by resource bottlenecks. Requests accumulate within the system. As resources increase, the scheduler can better match demand with available capacity. Execution becomes more orderly. This reduces the amplification effect of extreme workloads on latency. The trend highlights how collaborative scheduling amplifies stability gains as resource conditions improve.

From the perspective of processing capacity, throughput varies significantly with the resource supply ratio. This reflects the sensitivity of the scheduling strategy to the release of parallelism. In ranges where resources are sufficiently available, the scheduler can effectively coordinate execution across tenants. Overall system output increases markedly. When supply continues to grow, throughput gains gradually diminish. Performance becomes governed less by resource constraints and more by scheduling efficiency and collaboration quality. This behavior is consistent with saturation effects commonly observed in distributed inference systems.

Regarding multi-tenant fairness, adjustments in the resource supply ratio also influence service balance across tenants. Moderate resource redundancy helps the scheduler preserve service consistency while maintaining overall efficiency. It prevents biased allocation driven by intense competition. When resources are either too scarce or excessively abundant, fairness can degrade to some extent. This indicates that collaborative scheduling requires proper alignment between resource conditions and policy constraints. Overall, the sensitivity analysis shows that learning driven collaborative scheduling can jointly support efficiency, stability, and multi-tenant fairness within an appropriate range of resource supply.

The tenant population size reflects the level of concurrency complexity in a multi-tenant distributed inference system. Changes in this scale directly affect the intensity of request contention and the available space for scheduling decisions. As the number of tenants varies, the scheduling policy must cope with both workload aggregation effects and changes in resource allocation granularity. To examine the behavior of learning driven collaborative scheduling under different tenant scales, the following visualization illustrates how system performance responds to discrete tenant size configurations, and the experimental results are shown in Figure 5.

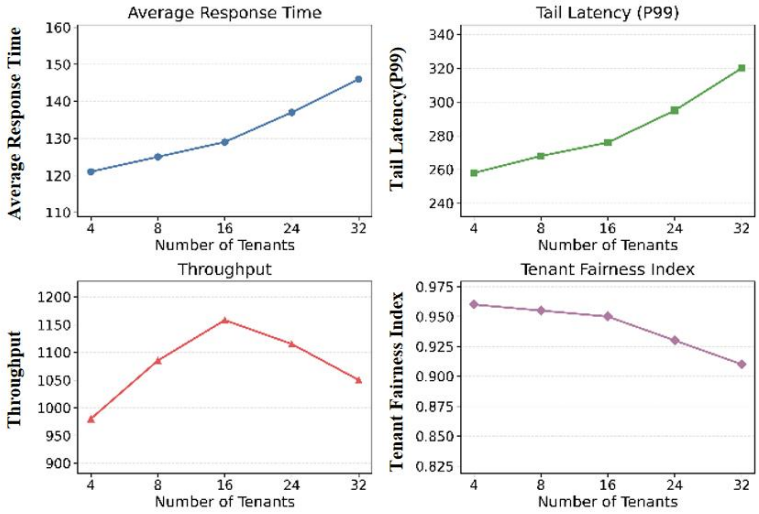

Figure 5. Performance variation trends of the learning driven collaborative scheduling mechanism in multi-tenant distributed inference systems under different tenant population sizes

As the tenant population increases, the system faces higher concurrency complexity and stronger resource contention. The scheduling policy must operate within a higher-dimensional state space. The overall trend shows that growth in tenant scale makes the scheduling environment more dynamic. The system must handle requests from more sources and more complex workload aggregation effects. This places greater demands on global state awareness and decision stability of the scheduling strategy.

In terms of response time and tail latency, an increase in tenant count amplifies queue buildup and resource contention. Scheduling pressure along the temporal dimension becomes more evident. As concurrent entities grow, the system relies more heavily on precise scheduling decisions. Only scheduling mechanisms that can effectively model cross-tenant interactions are able to maintain relatively stable service behavior under sustained load accumulation. This pattern indicates that tenant scale is an important external factor influencing system stability.

From the perspective of system processing capacity, throughput exhibits stage-dependent behavior as tenant scale changes. Under moderate tenant sizes, the scheduling strategy can effectively integrate requests from different tenants and achieve efficient resource reuse. When the number of tenants continues to grow, scheduling complexity and coordination overhead increase simultaneously. This constrains overall processing efficiency. The observation suggests that learning driven collaborative scheduling must continuously balance parallel efficiency and scheduling overhead as tenant scale expands.

At the level of fairness, changes in tenant population directly affect service consistency. As tenant scale increases, the scheduling policy must coordinate resource demands at finer granularity to prevent long-term bias toward a subset of tenants. The observed trends indicate that collaborative scheduling can maintain good service balance within a certain scale range. Under high concurrency, however, fairness becomes more difficult to preserve. This highlights the critical role of tenant scale in shaping the robustness of scheduling strategies in multi-tenant inference systems.

The proportion of request task types characterizes how tasks with different computational complexity and execution properties are composed within the overall workload. Changes in this proportion significantly affect how the scheduling policy organizes and allocates resources. In multi-tenant distributed inference scenarios, an imbalance in task type distribution amplifies model heterogeneity and execution path divergence. This places higher demands on scheduling stability and coordination capability. To analyze the behavior of learning driven collaborative scheduling under different task type proportion settings, the following visualization presents the response trends of system performance as task structure varies across discrete ratio configurations, and the experimental results are shown in Figure 6.

When the proportion of request task types changes, the computational complexity structure carried by the system is adjusted accordingly. The scheduling policy must coordinate among more heterogeneous execution paths. As the share of computation-intensive tasks increases, scheduling decisions must consider not only request volume but also resource occupancy duration and execution characteristics of different tasks. This shifts the scheduling problem from simple load distribution to a holistic trade-off across heterogeneous execution efficiency.

At the level of response time and tail latency, imbalance in task type proportions amplifies the sensitivity of scheduling to execution path differences. When high complexity tasks dominate the workload, the accumulation of long-running requests becomes more pronounced. The scheduler must preserve overall execution rhythm while preventing short tasks from being persistently blocked. This behavior indicates that learning driven collaborative scheduling must rely on global state modeling to suppress the impact of extreme requests on system stability under task heterogeneity.

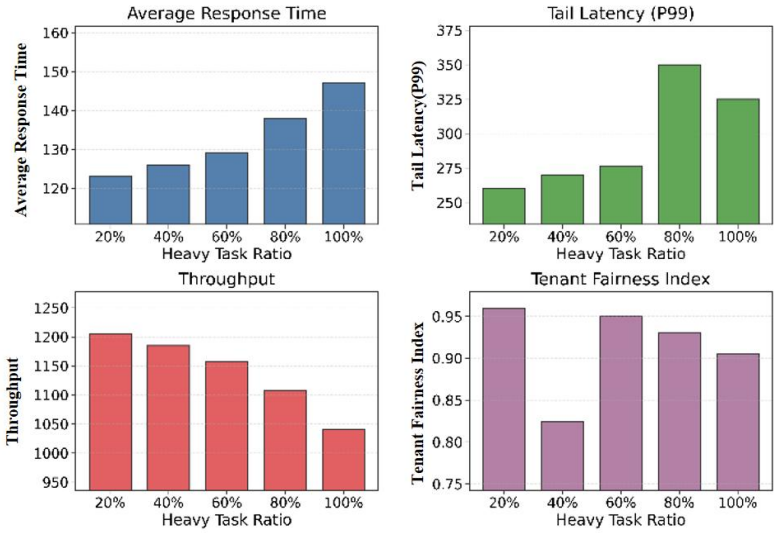

Figure 6. Performance variations of the learning driven collaborative scheduling mechanism in multi-tenant distributed inference systems under different request task type proportion configurations

From the perspective of overall system processing capacity, throughput variations reflect the adaptability of the scheduling strategy in resource utilization. As the task structure shifts toward more complex types, the number of requests that can be completed per unit time is naturally constrained. The scheduler must adopt more refined execution arrangements to mitigate prolonged resource occupation by long tasks. This trend shows that collaborative scheduling focuses not only on whether resources are fully utilized, but also on whether allocation patterns support sustained processing efficiency.

In terms of multi-tenant fairness, adjustments in task type proportions reshape competitive relationships among tenants. When requests from certain tenants are dominated by high complexity tasks, the scheduler must avoid excessive bias toward lightweight tasks while maintaining overall efficiency. Observed trends indicate that learning driven collaborative scheduling can partially relieve fairness pressure induced by task structure changes. Under highly skewed task proportions, however, the system must adopt a more cautious balance between efficiency and fairness. This highlights the significant influence of task heterogeneity on the design of multi-tenant scheduling strategies.

5. Conclusion

This paper focuses on the scheduling complexity of multi-tenant distributed inference services under dynamic workloads and shared resources. It proposes a learning driven collaborative scheduling mechanism that formulates scheduling decisions from a holistic system perspective. Through continuous modeling of system states and coordinated characterization of cross-tenant utility, the approach guides scheduling behavior toward an inherent balance among efficiency, stability, and fairness. It addresses the limited adaptability of traditional rule-driven scheduling in complex inference scenarios. The proposed framework enables more effective coordination of concurrent tenant requests and heterogeneous resource allocation. It provides a system-level solution for building inference services that support high concurrency and sustained operation.

From the perspective of application and evolution, learning driven collaborative scheduling offers general support for large-scale deployment of cloud inference platforms, edge intelligence services, and industry-oriented intelligent systems. As inference models continue to grow in scale and service forms become increasingly diverse, distributed inference systems will face more complex operating conditions and stricter stability requirements. Future extensions can enrich system state representations and

collaborative constraint designs. Integration with emerging computing architectures and inference paradigms can further enhance the adaptability and generalization of scheduling policies. These developments will lay a solid foundation for efficient operation and reliable deployment of next-generation intelligent inference services.

References

[1] X. Zhang, X. Wang and X. Wang, "A Reinforcement Learning-Driven Task Scheduling Algorithm for Multi-Tenant Distributed Systems," Proceedings of the 2025 5th International Conference on Intelligent Communications and Computing (ICICC), pp. 197-201, 2025.

[2] C. Batista, F. Morais, E. Cavalcante et al., "Managing Asynchronous Workloads in a Multi-Tenant Microservice Enterprise Environment," Software: Practice and Experience, vol. 54, no. 2, pp. 334-359, 2024.

[3] A. Mishra, "Evaluating the Architectural Patterns for Multi-Tenant Deployments," IJLRP-International Journal of Leading Research Publication, vol. 4, no. 12, 2023.

[4] S. Kim, S. Seah and et al., "MoCA: Memory-Centric, Adaptive Execution for Multi-Tenant Deep Neural Networks," Proceedings of the 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2023.

[5] M. Simić, J. Dedeić, M. Stojkov et al., "A Hierarchical Namespace Approach for Multi-Tenancy in Distributed Clouds," IEEE Access, vol. 12, pp. 32597-32617, 2024.

[6] O. Rottenstreich and J. Yallouz, "Edge-Disjoint Tree Allocation for Multi-Tenant Cloud Security in Datacenter Topologies," IEEE/ACM Transactions on Networking, vol. 32, no. 4, pp. 2858-2874, 2024.

[7] P. Liu and J. Guitart, "Dynamic In-Node Group-Aware Scheduling for Multi-Tenant Machine Learning Services on Kubernetes," Proceedings of the 2025 IEEE 18th International Conference on Cloud Computing (CLOUD), pp. 65-74, 2025.

[8] M. Bin-Yahya, A. Shani, H. Shafieirad et al., "Symphony: Collective Coordination in Multi-Tenant GPU Clusters," Proceedings of the 2025 IEEE 33rd International Conference on Network Protocols (ICNP), pp. 1-13, 2025.

[9] F. Saleh, S. Karmoshi, A. O. Fapojuwo et al., "TARA: Tenant-Aware Resource Allocation in Multi-Tenant Data Centers," IEEE Transactions on Network and Service Management, 2024.

[10] S. Hossain, W. Tang, C. Chenli et al., "MtDB: A Decentralized Multi-Tenant Database for Secure Data Sharing," Cryptology ePrint Archive, 2025.

[11] S. El Kafhali and O. Ghandour, "DSGTA: A Dynamic and Stochastic Game-Theoretic Allocation Model for Scalable and Efficient Resource Management in Multi-Tenant Cloud Environments," Future Internet, vol. 17, no. 12, p. 583, 2025.

[12] A. Gama Garcia, J. M. Alcaraz Calero, H. Mora Mora et al., "ServiceNet: Resource-Efficient Architecture for Topology Discovery in Large-Scale Multi-Tenant Clouds," Cluster Computing, vol. 27, no. 7, pp. 8965-8982, 2024.