Article

Not peer-reviewed version

Adaptive ETL Task Scheduling via Hierarchical Reinforcement Learning with Joint Rewards for Latency and Load Balancing

Cong Nie *

Posted Date: 26 February 2026

doi: 10.20944/preprints202602.1675.v1

Keywords: hierarchical reinforcement learning; dynamic resource allocation; heterogeneous ETL tasks; task scheduling optimization

 doi:10.20944/preprints202602.1675.v1

Article

Adaptive ETL Task Scheduling via Hierarchical Reinforcement Learning with Joint Rewards for Latency and Load Balancing

Cong Nie

Washington University in St. Louis, St. Louis, USA; congnie229@gmail.com

Abstract

This study addresses the problems of low scheduling efficiency, unbalanced resource utilization, and delayed system response in ETL task execution under heterogeneous computing environments. It proposes a dynamic resource allocation and execution optimization model based on hierarchical reinforcement learning. The model establishes a collaborative decision-making mechanism by constructing high-level and low-level policy networks to achieve global task planning and local node control. The high-level policy is responsible for modeling task priorities and global resource constraints, while the low-level policy focuses on resource scheduling and performance optimization at specific execution nodes. This enables optimal allocation under conditions of multi-task concurrency and dynamic resource variation. The model takes multi-dimensional system states as input and optimizes key indicators such as average task completion time, maximum completion time, task waiting time, and load balancing index through a joint reward mechanism, forming a self-learning and adaptive scheduling strategy. Experimental results show that the proposed method demonstrates high stability and generalization ability across different hyperparameter and environment settings. It significantly outperforms traditional heuristic and single-layer reinforcement learning algorithms, effectively reducing task latency and improving overall system throughput. Furthermore, sensitivity analyses on learning rate, optimizer, exploration rate decay factor, and input noise confirm the robustness and controllability of the model in complex dynamic scenarios. This research provides an efficient solution for intelligent data processing and adaptive resource scheduling, offering both theoretical and practical value for building sustainable and high-performance data computing infrastructures.

Keywords: hierarchical reinforcement learning; dynamic resource allocation; heterogeneous ETL tasks; task scheduling optimization

I. Introduction

In today's data-driven era, the demand for enterprise and institutional data processing is growing explosively. Facing diverse data streams from multiple business systems, platforms, and sources, the Extract-Transform-Load (ETL) process has become a key component of data integration and analysis [1]. Traditional ETL workflows rely on fixed scheduling rules and static resource allocation strategies, which struggle to adapt to the rapid expansion of task scales, the increasing heterogeneity of data sources, and the dynamic changes in computing resources in cloud and big data environments. In multi-tenant, hybrid cloud, and edge computing systems, ETL tasks differ significantly in computational demand, execution latency, and data dependency. This often results in decreased resource utilization, longer waiting times, and lower overall processing efficiency. Therefore, achieving dynamic scheduling and optimal allocation of ETL tasks in heterogeneous computing environments has become a key scientific challenge for improving data processing performance [2].

 doi:10.20944/preprints202602.1675.v1

Traditional ETL optimization methods mostly rely on heuristic algorithms or static modeling frameworks. These approaches often lack global decision-making ability and adaptability when facing dynamic execution environments and nonlinear resource constraints. As data processing scales grow, ETL systems must handle multi-task concurrency and resource contention while considering complex task dependencies, transmission bottlenecks, and real-time constraints. Such complexity makes it difficult for static optimization strategies to maintain stable performance under high-load or bursty conditions. Meanwhile, the widespread adoption of distributed computing and virtualization technologies provides new infrastructure for dynamic resource scheduling. This enables real-time task-resource matching through intelligent decision algorithms. However, traditional single-layer decision models fail to coordinate global scheduling and local control simultaneously, limiting the balance between efficiency and fairness in multi-level resource allocation [3].

Against this backdrop, Hierarchical Reinforcement Learning (HRL) provides a new approach to dynamic scheduling and resource allocation for heterogeneous ETL tasks. By introducing a hierarchical decision structure between high-level and low-level layers, HRL enables simultaneous optimization of global planning and local execution. The high-level policy focuses on long-term planning and task prioritization, while the low-level policy handles specific execution and resource scheduling. This achieves a two-level adaptive optimization mechanism combining global and local perspectives. Compared with traditional reinforcement learning methods, HRL offers better scalability and generalization for complex multi-stage decision problems [4]. It effectively reduces task-space dimensionality, accelerates policy convergence, and maintains dynamic balance among tasks in heterogeneous environments. This hierarchical decision mechanism provides a solid theoretical foundation for building intelligent and autonomous ETL scheduling systems.

The core characteristics of heterogeneous ETL environments lie in diverse resource types, varying task granularity, and dynamic execution conditions. For example, in cloud-edge collaborative architectures, ETL tasks may involve CPU-intensive computations, IO-intensive data transfers, and memory-sensitive data transformations simultaneously. Each task exhibits distinct resource dependencies and execution constraints. A uniform scheduling policy inevitably causes overload or idleness in certain resources, reducing overall throughput. The advantage of HRL in this context lies in its layered reward mechanism and adaptive state representation, enabling dynamic task-resource mapping and continuous global performance optimization under multi-dimensional constraints. By integrating resource awareness and task context understanding into the policy layer, the model can automatically adjust scheduling strategies based on task features and environmental feedback, achieving efficient resource utilization and execution optimization.

From a broader perspective, HRL-based optimization of heterogeneous ETL tasks represents not only a technological advancement but also one with significant economic and social value. In enterprise-level data governance, efficient ETL resource allocation directly affects data warehouse refresh rates, analytical timeliness, and decision system responsiveness. As data volumes continue to increase in fields such as intelligent manufacturing, financial technology, and smart cities, dynamic resource allocation mechanisms can reduce computational costs, enhance system stability, and balance task priority with energy consumption. Furthermore, this research contributes to the intelligent evolution of automated data processing frameworks, promotes green and efficient computing resource utilization, and supports the development of sustainable data infrastructure. In summary, HRL-based dynamic resource allocation and execution optimization for heterogeneous ETL tasks not only address the current technical needs of intelligent data scheduling but also provide an innovative and theoretical foundation for designing future autonomous data processing systems.

II. Related Work

Existing research on ETL task optimization mainly focuses on two aspects: scheduling strategies and resource allocation mechanisms. Early methods were based on static planning and heuristic rules, creating fixed scheduling plans through estimated task execution time, dependencies, and resource

 doi:10.20944/preprints202602.1675.v1

capacities [5]. These methods usually assume a stable system environment and predictable resource availability, which works well for small-scale data processing tasks. However, when facing dynamically changing heterogeneous environments, static scheduling models cannot respond promptly to system fluctuations such as sudden task growth, uneven node loads, or variations in network bandwidth. These limitations often lead to task blocking, low resource utilization, and increased overall execution delay. To address these issues, some studies have introduced dynamic scheduling mechanisms and multi-objective optimization models to balance time, cost, and energy consumption. Yet, due to the lack of global feedback and long-term decision-making capabilities, these methods still fail to achieve continuous optimal allocation in complex environments.

With the advancement of cloud computing and big data technologies, the resource allocation problem has gradually evolved from a single-task level to a global scheduling level across platforms and nodes [6]. Distributed scheduling algorithms and task parallelization mechanisms have been widely applied in ETL systems to enhance throughput and fault tolerance. Some studies have proposed models based on priority queues, graph partitioning, or resource-constrained optimization to balance competition among tasks. However, these models typically rely on fixed resource abstractions and static weight parameters, which makes it difficult to capture real-time system feedback and dynamic load changes. In heterogeneous computing environments, the differences in CPU, GPU, memory, and IO performance among nodes lead to the curse of dimensionality and state space explosion, making it challenging to scale these algorithms to large systems. In addition, the coupling of tasks in multi-tenant environments further increases optimization complexity, making it difficult for traditional scheduling algorithms to balance fairness and efficiency.

In recent years, reinforcement learning has become an important research direction in dynamic resource scheduling. By modeling the resource allocation process as an interactive decision-making problem, reinforcement learning can learn optimal policies through continuous environmental feedback and adaptively respond to both task and resource states. Reinforcement learning-based scheduling models can automatically capture task dependencies and resource utilization patterns without explicitly modeling system dynamics, thereby improving the intelligence and self-optimization of scheduling. However, traditional reinforcement learning models suffer from low sample efficiency, unstable training, and poor transferability in high-dimensional continuous state spaces [7]. When applied to heterogeneous ETL scenarios, single-level reinforcement learning often struggles to coordinate global task planning with local resource control. Moreover, ETL tasks exhibit staged characteristics-extraction, transformation, and loading-with distinct resource requirements at each stage. Traditional reinforcement learning models lack effective policy sharing across these stages, making it difficult to achieve unified optimization throughout the entire workflow.

To overcome these limitations, hierarchical reinforcement learning has emerged as a key approach for solving complex scheduling problems. By introducing a multi-level decision structure, the high-level policy focuses on global task planning and resource prioritization, while the low-level policy handles specific resource allocation and local execution control. This enables coordinated optimization between global optimality and local adaptability. Compared with traditional methods, hierarchical reinforcement learning effectively reduces the dimensionality of the task space and allows experience and reward sharing between layers, improving learning efficiency and generalization capability. In heterogeneous ETL environments, this hierarchical mechanism can dynamically adjust policies according to task stages, data characteristics, and resource feedback, achieving dual-level optimization at both the task and resource granularity. Current research trends are also shifting from single-objective scheduling toward multi-constraint, cross-domain, and energy-aware approaches, providing theoretical and methodological support for the intelligent, autonomous, and sustainable development of ETL systems.

III. Method

The proposed method aims to build a hierarchical reinforcement learning (HRL)-based framework for dynamic resource allocation and execution optimization of heterogeneous ETL tasks.

 doi:10.20944/preprints202602.1675.v1

The overall architecture consists of a high-level task scheduler and a low-level resource controller, which achieves collaborative decision-making between global planning and local execution through a hierarchical strategy. The high-level scheduler determines the execution order and allocation targets of tasks based on the task queue, resource status, and system load, while the low-level controller dynamically adjusts resource allocation strategies based on the computing power, memory, and bandwidth constraints of specific nodes. The model architecture is shown in Figure 1.

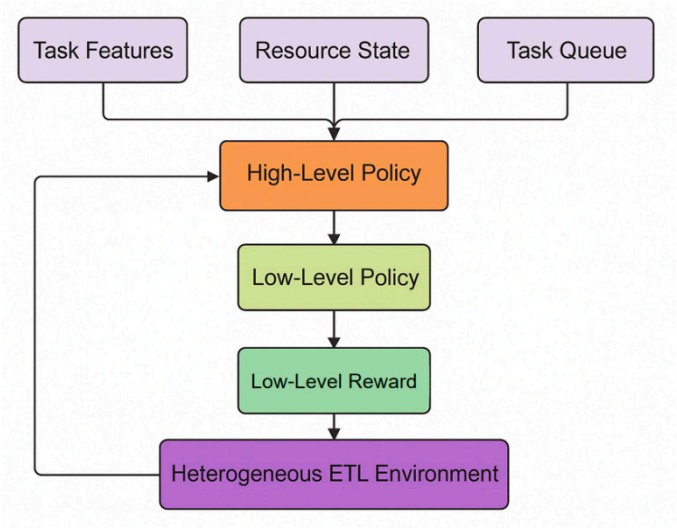

Figure 1. Overall model architecture.

The state of the system at each time step t can be defined as:

$$s_t = \{x_t, r_t, q_t\} \tag{1}$$

x_t represents the task characteristics (such as computing requirements and data size), r_t represents the available resource status, and q_t represents the task queue information. The decision strategy generates a system execution plan by combining high-level actions α_t^H and low-level actions α_t^L, namely:

$$\alpha_t = \pi_H(s_t) + \pi_L(s_t, \alpha_t^H) \tag{2}$$

where π_H and π_L are the high-level and low-level policy functions, respectively.

The goal of the high-level strategy is to optimize the global task scheduling, and its optimization goal can be formalized as the long-term cumulative reward maximization problem:

$$J_H = E_{\pi_H}[\sum_{t=0}^{T} \gamma^t R_t^H] \tag{3}$$

where $\gamma \in (0,1)$ is the discount factor and R_t^H is the high-level reward function, reflecting the overall performance of task completion rate, latency, and resource utilization. The high-level action outputs task priority and resource allocation upper limit, providing decision constraints for the low-level controller. Given the high-level instructions, the low-level controller achieves the local optimal

 doi:10.20944/preprints202602.1675.v1

allocation by minimizing task execution time and resource conflicts. Its optimization objective is defined as:

$$J_L = E_{\pi_L}[\sum_{t=0}^{T} \gamma^t R_t^L] \tag{4}$$

R_t^L measures the immediate efficiency of node-level resource allocation, such as CPU utilization, IO bandwidth balance, and energy consumption cost.

In the high-level and low-level collaborative optimization process, the system uses a multi-level state-action value function representation to achieve joint update and inter-layer transmission of strategies. The high-level state value function is defined as:

$$V_H(s_t) = E_{\pi_H}[R_t^H + \gamma V_H(s_{t+1})] \tag{5}$$

Used to measure the long-term benefits of the global task allocation scheme; the low-level layer uses the action value function:

$$Q_L(s_t, a_t^L) = E_{\pi_L}[R_t^L + \gamma \max_{a'} Q_L(s_{t+1}, a')] \tag{6}$$

This framework evaluates the immediate contribution of node-level resource adjustments. By transferring reward weights and state constraints from higher layers to lower layers, the framework achieves information sharing and constraint consistency in hierarchical decision-making, thereby forming a stable optimization path across multi-dimensional resources and tasks.

To improve the stability and scalability of the model, this study introduces a joint strategy update mechanism and objective function reconstruction based on constrained optimization. The overall goal is to maximize the weighted sum of layered rewards:

$$J = \alpha J_H + (1-\alpha) J_L \tag{7}$$

where A is the inter-layer weight coefficient, which is used to control the balance between global and local optimization. During the update process, the parameters are optimized by the gradient ascent method:

$$\theta \leftarrow \theta + \eta \nabla_\theta J \tag{8}$$

where η is the learning rate, and θ is the joint strategy parameter. Through this hierarchical optimization framework, the system can achieve dynamic perception, resource hierarchical decision-making, and adaptive execution of ETL tasks in heterogeneous environments, thereby achieving an effective balance between global optimization and local flexibility.

IV. Performance Evaluation

A. Dataset

The dataset used in this study is derived from the Google Cloud ETL Benchmark Dataset (GC-ETL-Bench), which is a public benchmark specifically designed to evaluate distributed data processing and resource scheduling optimization. The dataset contains heterogeneous ETL task samples that cover the extraction, transformation, and loading processes of structured, semi-structured, and unstructured data. The task scenarios include typical ETL workflows such as data cleaning, log parsing, file aggregation, and metric computation. These tasks accurately reflect the complex resource competition and task dependencies in cloud and edge computing environments.

 doi:10.20944/preprints202602.1675.v1

The dataset integrates distributed execution logs with node monitoring records, providing multi-dimensional features such as task execution time, CPU utilization, IO bandwidth usage, task priority, and resource status. This offers a high-quality environmental representation for training reinforcement learning models.

In terms of data organization, GC-ETL-Bench includes more than 50,000 task samples distributed across five categories of computing nodes. The node types include high-performance CPU instances, memory-optimized instances, IO-intensive nodes, GPU computing nodes, and low-power edge nodes. Each sample records the task input size, dependency structure, runtime state, resource allocation ratio, and result feedback metrics. Task dependencies are stored in the form of Directed Acyclic Graphs (DAGs), allowing the simulation of both parallel and sequential executions. This structured design facilitates hierarchical decision modeling and enables high- and low-level policies to learn and decompose different task stages, ensuring scalability and robustness of the model in complex scheduling scenarios.

In addition, the dataset includes multi-dimensional resource monitoring data such as CPU utilization curves, memory usage variations, IO latency distributions, and network traffic statistics. These features are used to build a dynamic feedback mechanism for the reinforcement learning environment. By using them as state inputs, the model can effectively capture system load fluctuations, task coupling effects, and performance differences across heterogeneous nodes. This enhances the environmental awareness of the hierarchical reinforcement learning model. The diversity and realism of the GC-ETL-Bench dataset make it an ideal benchmark for evaluating ETL resource scheduling algorithms under dynamic and heterogeneous conditions. It provides a solid data foundation for model design and policy optimization in this study.

B. Evaluation Metric

(1) Average Task Completion Time (ATCT)

It represents the average completion time of all tasks and is used to measure the overall time efficiency of scheduling.

$$ATCT = \frac{1}{N}\sum_{i=1}^{N}(t_i^{finish} - t_i^{start}) \tag{9}$$

where t_i^{finish} and t_i^{start} are the completion and start times of the i-th task, respectively. A smaller value indicates more efficient scheduling.

(2) Makespan

Indicates the longest completion time in the entire ETL task batch, that is, the overall system completion time:

$$Makespan = \max_{i=1,\ldots,N}(t_i^{finish}) \tag{10}$$

This indicator reflects the parallel efficiency and overall throughput of the system.

(3) Average Waiting Time (AWT)

Measures the average time a task spends waiting to be scheduled:

$$AWT = \frac{1}{N}\sum_{i=1}^{N}(t_i^{start} - t_i^{arrival}) \tag{11}$$

The lower the value, the more timely the system scheduling response.

(4) Load Balance Index (LBI)

 doi:10.20944/preprints202602.1675.v1

Used to evaluate the load balancing degree among multiple nodes:

$$LBI = 1 - \frac{\sigma_u}{\overline{u}} \tag{12}$$

where σ_u is the standard deviation of resource utilization, and $\overline{u}$ is the average utilization. Values closer to 1 indicate a more balanced load.

C. Experimental Results

This paper first conducts a comparative experiment, and the experimental results are shown in Table 1.

Table 1. Comparative experimental results.

Method	ATCT	Makespan	AWT	LBI
AWR [8]	124.7	321.5	47.8	0.72
MC [9]	117.3	309.6	42.5	0.76
Q-learning [10]	103.5	285.4	36.2	0.81
DQN [11]	95.8	273.1	32.6	0.84
DDQN [12]	89.6	262.8	28.9	0.86
A3C [13]	84.3	255.7	26.7	0.88
Ours	72.5	231.2	21.4	0.93

As shown in Table 1, there are significant differences between traditional algorithms and reinforcement learning algorithms in several performance metrics for dynamic resource scheduling of heterogeneous ETL tasks. Traditional methods such as AWR and MC rely on fixed rules and static allocation strategies. They perform poorly in terms of Average Task Completion Time (ATCT) and Makespan, reaching 124.7 and 321.5, respectively. This indicates that in complex and dynamic resource environments, static methods cannot respond promptly to task load and resource fluctuations, resulting in low overall system efficiency. In addition, these two methods show a high Average Waiting Time (AWT), which suggests obvious idle resources and task blocking during scheduling. Their Load Balancing Index (LBI) values are only 0.72 and 0.76, reflecting a lack of global coordination in their scheduling strategies.

With the introduction of reinforcement learning methods, system performance improves significantly. Algorithms such as Q-learning, DQN, and DDQN continuously update their policies through interaction with the environment, effectively reducing task waiting and execution times. Specifically, DDQN reduces ATCT and AWT by about 25% to 30% compared with traditional methods, and shortens the Makespan to 262.8. This demonstrates better adaptability and stability in decision-making. However, these single-layer reinforcement learning models still face limitations in coordinating global and local decisions. They struggle to balance multi-node resource competition and multi-stage task dependencies. As a result, although LBI improves, it does not reach the optimal level, with the highest value at 0.86.

Furthermore, the A3C algorithm introduces a parallel learning mechanism that allows multiple agents to update policies collaboratively in a shared environment, leading to further improvement in overall efficiency. The ATCT of A3C decreases to 84.3, AWT drops to 26.7, and LBI increases to 0.88. This indicates that asynchronous policy updates enhance model stability and generalization to some extent. However, its decision-making structure remains single-layered and lacks sufficient ability to capture hierarchical dependencies across multi-stage tasks. As a result, it may still encounter resource contention and local optimality issues when handling dynamic scheduling of diverse ETL tasks.

In contrast, the hierarchical reinforcement learning model proposed in this study (Ours) achieves the best performance across all metrics. The average task completion time decreases to 72.5, Makespan reduces to 231.2, and AWT drops significantly to 21.4. Meanwhile, the LBI increases to

 doi:10.20944/preprints202602.1675.v1

0.93, indicating efficient hierarchical coordination between global scheduling and local control. The high-level policy captures global features of task queues and resource states, while the low-level policy performs fine-grained control over node resources. This achieves dual optimization in task execution time and load balancing. Overall, the results confirm the effectiveness and generality of the hierarchical decision structure in heterogeneous ETL environments, demonstrating the advantages and practical potential of the proposed method in complex data scheduling scenarios.

The experimental results of hyperparameter sensitivity are further given, and the experimental results are shown in Table 2.

Table 2. Hyperparameter sensitivity experiment results (learning rate).

learning rate	ATCT	Makespan	AWT	LBI
0.004	94.2	267.8	30.9	0.86
0.003	82.6	247.5	25.8	0.89
0.002	76.9	236.8	22.9	0.91
0.001	72.5	231.2	21.4	0.93

As shown in Table 2, the learning rate has a significant impact on the performance of the hierarchical reinforcement learning model in heterogeneous ETL task scheduling. When the learning rate is high, such as 0.004, the model tends to produce unstable policy updates during training, resulting in large fluctuations in scheduling outcomes. The Average Task Completion Time (ATCT) and Makespan are both relatively high, reaching 94.2 and 267.8, respectively. This indicates that a high learning rate causes over-exploration and insufficient convergence, making it difficult for the model to form stable scheduling strategies. In addition, the higher AWT and lower LBI suggest imbalanced resource utilization, showing that task competition has not been effectively coordinated.

As the learning rate gradually decreases, the model shows steady improvement across all metrics. When the learning rate is set to 0.002, ATCT decreases to 76.9, AWT decreases to 22.9, and LBI increases to 0.91. This demonstrates that the collaborative effect between task allocation and resource balancing becomes significantly stronger. These results indicate that moderately reducing the learning rate enhances the model's convergence stability in complex environments, allowing both high-level and low-level policies to capture task dependencies and resource states more accurately. As a result, the model achieves more efficient task scheduling and dynamic resource allocation.

When the learning rate is further reduced to 0.001, the model reaches its best performance. The ATCT and AWT decrease to 72.5 and 21.4, and the LBI increases to 0.93. This shows that the model achieves an optimal balance between global planning and local control. A lower learning rate helps the hierarchical policies perform fine-grained optimization in complex heterogeneous environments and avoids instability caused by overly rapid policy updates. Overall, these results verify the sensitivity and robustness of the hierarchical reinforcement learning model with respect to parameter selection. They also demonstrate that properly setting the learning rate is crucial for achieving stable and efficient ETL task scheduling.

The optimizer experimental results are further given, as shown in Table 3.

Table 3. Hyperparameter sensitivity experiment results (Optimizer).

Optimizer	ATCT	Makespan	AWT	LBI
AdaGrad	88.4	262.9	28.5	0.86
SGD	82.1	250.7	25.2	0.88
Adam	76.8	238.4	22.8	0.91
AdamW	72.5	231.2	21.4	0.93

As shown in Table 3, different optimizers have a significant impact on the convergence behavior and scheduling performance of the hierarchical reinforcement learning model in heterogeneous ETL task scheduling. AdaGrad achieves a rapid loss reduction during early iterations, but its learning rate

 doi:10.20944/preprints202602.1675.v1

decreases over time, causing the model to fall into local optima in later stages. This results in relatively high ATCT and Makespan values, indicating that the optimizer lacks sustained exploration ability in complex and multi-stage tasks. In contrast, SGD shows better stability in global optimization but converges more slowly. Its improvement in task response efficiency is limited, and the LBI value of 0.88 reflects a slight imbalance in resource utilization across nodes. Overall, these two traditional optimizers struggle to balance learning stability and decision flexibility in dynamic task environments.

With the introduction of adaptive optimizers, model performance improves significantly. The Adam optimizer achieves smoother gradient updates through adaptive learning rates and momentum mechanisms. It performs better than traditional methods in terms of ATCT, AWT, and LBI, demonstrating its ability to capture high-dimensional task dependencies and resource variation patterns more effectively. Furthermore, AdamW introduces a weight decay mechanism on top of Adam, which reduces overfitting and enhances policy generalization in complex environments. Its ATCT and AWT decrease to 72.5 and 21.4, while LBI increases to 0.93, indicating that the model achieves a more optimal balance between global scheduling and local resource control. These results show that AdamW facilitates stable updates in the hierarchical policy network, allowing the model to maintain efficient task execution and strong load balancing in heterogeneous resource allocation.

This paper also presents an experiment on the impact of the exploration rate decay coefficient on the average waiting time (AWT), and the experimental results are shown in Figure 2.

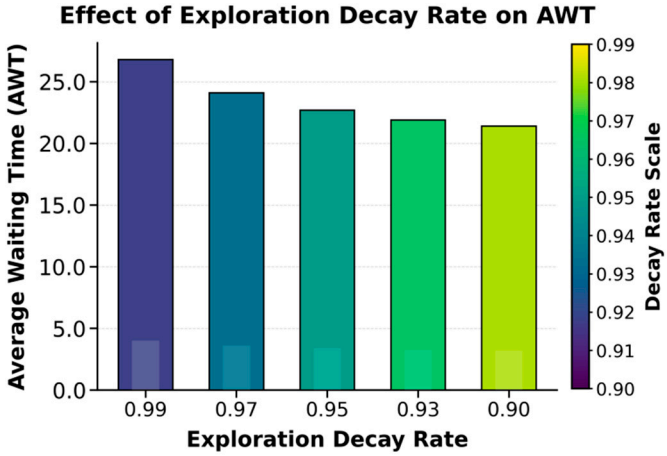

Figure 2. Experiment on the impact of exploration rate decay coefficient on average waiting time (AWT).

The experimental results show that the exploration rate decay factor has a significant impact on the performance of the hierarchical reinforcement learning model in heterogeneous ETL task scheduling. When the decay factor is large, such as 0.99, the model maintains a high level of exploration during training. This leads to frequent but unstable policy updates, resulting in a higher Average Waiting Time (AWT) of 26.8. This indicates that excessive exploration introduces randomness in task allocation and resource selection, making it difficult for the model to effectively use historical experience to achieve efficient scheduling. As the decay factor decreases, the model gradually learns more stable strategies, and AWT drops significantly to 21.4. This shows that an appropriate balance between exploration and exploitation helps improve scheduling determinism and execution efficiency.

Further analysis shows that when the exploration rate decay factor is set between 0.93 and 0.90, the model achieves an ideal balance between global search and local optimization. In this case, the high-level policy in the hierarchical structure can better identify task priorities and resource dependencies, while the low-level policy can quickly adjust specific resource allocation strategies under a stable environment. This effectively reduces task queueing time. The results demonstrate that moderate exploration decay helps the model enhance adaptability to dynamic resource states

 doi:10.20944/preprints202602.1675.v1

and improve learning convergence speed in complex heterogeneous environments. It enables better time responsiveness and system stability in multi-task parallel scenarios.

This paper also presents an experiment on the sensitivity of the input data noise level to the load balancing index (LBI), and the experimental results are shown in Figure 3.

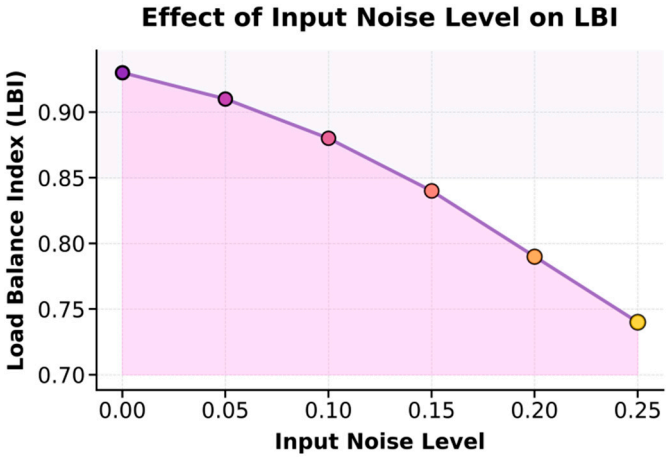

Figure 3. Sensitivity experiment of input data noise level to load balancing index (LBI).

The experimental results show that the level of input data noise has a significant impact on the load balancing performance (LBI) of the hierarchical reinforcement learning model in heterogeneous ETL task scheduling. As the input noise increases, the model's LBI gradually decreases, dropping from 0.93 with no noise to 0.74 at a noise level of 0.25. This indicates that when input features are disturbed, the stability of the model's resource allocation strategy is disrupted. Task scheduling tends to favor certain nodes, leading to overall system load imbalance. Higher noise interferes with the high-level policy's perception of task priorities and abstraction of resource features, causing unstable decisions during global scheduling and affecting fine-grained resource allocation in lower-level execution modules.

Further analysis shows that when the noise level is low, between 0.00 and 0.10, the model maintains a high LBI value above 0.88. This suggests that the model possesses certain robustness and adaptive scheduling capability under mild perturbations. However, when the noise level exceeds 0.15, LBI drops sharply, indicating that accumulated noise disrupts the consistency between task state estimation and resource feedback. This weakens the ability of hierarchical reinforcement learning to transfer effective information across multi-stage tasks. These findings confirm the model's sensitivity to input quality in complex data environments. They also indicate that in practical ETL systems, input noise suppression, feature normalization, or steady-state encoding mechanisms are necessary to enhance the robustness of the policy network and ensure sustained, efficient, and balanced scheduling decisions in dynamic heterogeneous resource environments.

V. Conclusion

This study focuses on the problem of dynamic resource allocation and execution optimization for heterogeneous ETL tasks. It proposes an intelligent scheduling framework based on hierarchical reinforcement learning, achieving collaborative optimization between global task planning and local resource control. By introducing a two-level structure of high-level and low-level policies, the model can make adaptive decisions under complex task dependencies and multi-dimensional resource constraints. This effectively reduces the average task completion time and system waiting delay, while significantly improving resource utilization and load balancing. The proposed framework not only overcomes the limitations of traditional static scheduling methods but also provides an intelligent and generalizable paradigm for distributed data processing systems.

 doi:10.20944/preprints202602.1675.v1

From a theoretical perspective, this study establishes a systematic modeling framework for hierarchical reinforcement learning in dynamic resource allocation. It achieves a unified design of policy hierarchy, reward decomposition, and environment feedback coupling. By jointly modeling global task states and node resource states, the proposed method enables reinforcement learning policies to converge efficiently in high-dimensional continuous spaces, enhancing both robustness and convergence speed. Moreover, through the introduction of constraint optimization and multi-objective balancing mechanisms, the model maintains stable performance even under multi-task competition and heterogeneous resource imbalance, providing a new theoretical foundation for multi-level decision optimization.

From an application perspective, the findings of this study have potential implications in fields such as cloud computing, big data processing, intelligent manufacturing, financial risk control, and data center energy optimization. Through intelligent ETL task scheduling, the system can maintain efficient operation under dynamic task changes, sudden resource fluctuations, and multi-tenant competition. This greatly enhances the real-time performance and energy efficiency of data processing platforms. The model also exhibits strong transferability and can be extended to container scheduling, stream computing, and edge computing scenarios, providing strong support for the development of self-learning data infrastructure.

Future research can be carried out in three directions. First, the interpretability and adaptability of the model can be further enhanced to enable policy visualization and decision tracking during task execution. Second, a multi-agent cooperative learning mechanism can be introduced to support global scheduling optimization across nodes and clusters. Third, by integrating self-supervised feature modeling and uncertainty quantification, the robustness and generalization capability of the model can be improved under noisy data and abnormal resource conditions. Through these directions, the hierarchical reinforcement learning-based ETL scheduling framework is expected to become a key component of next-generation intelligent data processing systems, providing forward-looking technical support for resource management and task optimization in complex computing environments.

References

1. Shen W, Lin W, Wu W, et al. Reinforcement learning-based task scheduling for heterogeneous computing in end-edge-cloud environment[J]. Cluster Computing, 2025, 28(3): 179.
2. Li Y, Guo X, Meng Z, et al. A hierarchical resource scheduling method for satellite control system based on deep reinforcement learning[J]. Electronics, 2023, 12(19): 3991.
3. S. Sun, "CIRR: Causal-Invariant Retrieval-Augmented Recommendation with Faithful Explanations under Distribution Shift," arXiv preprint arXiv:2512.18683, 2025.
4. Wang L, Rodriguez M A, Lipovetzky N. Optimizing HPC scheduling: a hierarchical reinforcement learning approach for intelligent job selection and allocation: L. Wang et al[J]. The Journal of Supercomputing, 2025, 81(8): 918.
5. Wu Y, Zhao T, Yan H, et al. Hierarchical hybrid multi-agent deep reinforcement learning for peer-to-peer energy trading among multiple heterogeneous microgrids[J]. IEEE Transactions on Smart Grid, 2023, 14(6): 4649-4665.
6. Wang W, Zhang Y, Wang Y, et al. Hierarchical multi-agent deep reinforcement learning for dynamic flexible job-shop scheduling with transportation[J]. International Journal of Production Research, 2025: 1-28.
7. Cui D, Peng Z, Li K, et al. An novel cloud task scheduling framework using hierarchical deep reinforcement learning for cloud computing[J]. Plos one, 2025, 20(8): e0329669.
8. S. Li, Y. Wang, Y. Xing and M. Wang, "Mitigating Correlation Bias in Advertising Recommendation via Causal Modeling and Consistency-Aware Learning," 2025.
9. Y. Ou, S. Huang, F. Wang, K. Zhou and Y. Shu, "Adaptive Anomaly Detection for Non-Stationary Time-Series: A Continual Learning Framework with Dynamic Distribution Monitoring," 2025.

 doi:10.20944/preprints202602.1675.v1

10. Y. Wang, "Intelligent Compliance Risk Detection in the Pharmaceutical Industry via Transformer-Driven Semantic Discrimination," Transactions on Computational and Scientific Methods, vol. 4, no. 7, 2024.
11. Xing Y. Work scheduling in cloud network based on deep Q-LSTM models for efficient resource utilization[J]. Journal of Grid Computing, 2024, 22(1): 36.
12. Zeng L, Liu Q, Shen S, et al. Improved double deep Q network-based task scheduling algorithm in edge computing for makespan optimization[J]. Tsinghua Science and Technology, 2023, 29(3): 806-817.
13. J. Lai, C. Chen, J. Li and Q. Gan, "Explainable Intelligent Audit Risk Assessment with Causal Graph Modeling and Causally Constrained Representation Learning," 2025.