Graph Assisted Offline-Online Deep Reinforcement Learning (GOODRL) for Dynamic Workflow Scheduling

Authors: Yifan Yang, Gang Chen, Hui Ma, Cong Zhang, Zhiguang Cao, Mengjie Zhang

ICLR 2025

2025/4/24

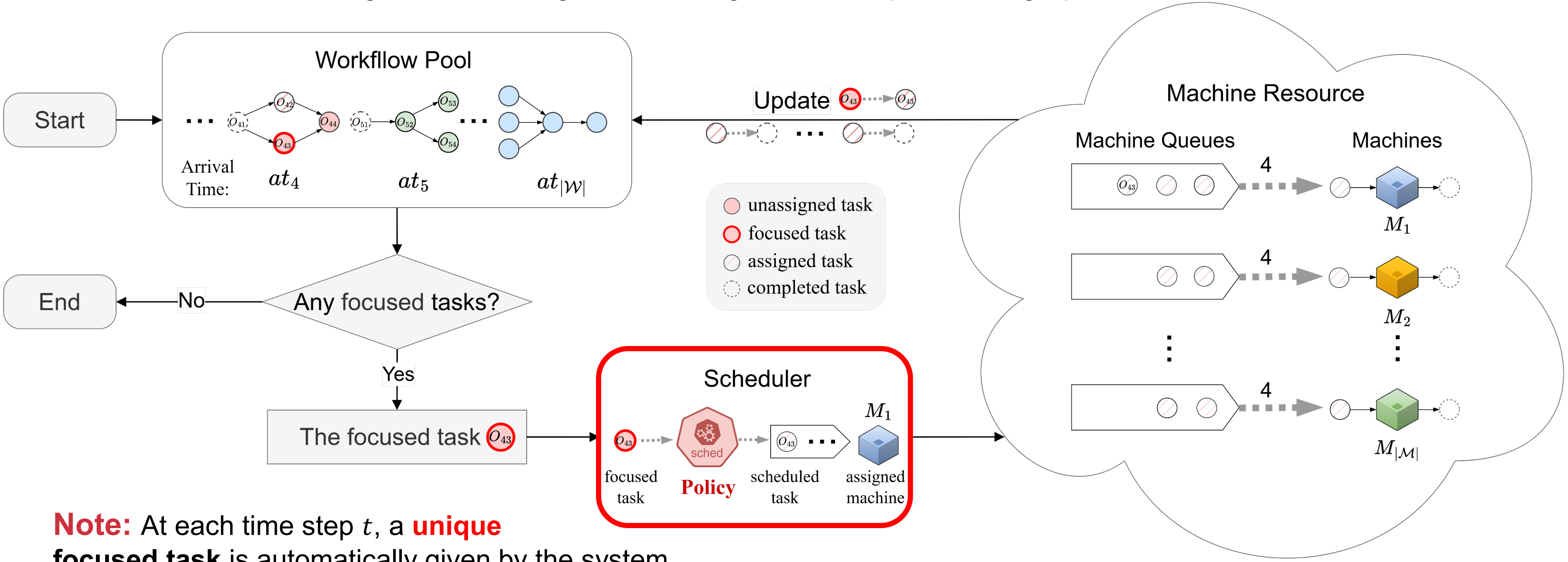
Paper

Code

Introduction

What is Dynamic Workflow Scheduling (DWS)?

- **Goals**: Assign dynamically arriving workflow tasks to machines to minimize **mean flowtime**
- **Workflows** are **DAGs**: Nodes = Tasks, Edges = Dependencies
- **Machines**: Heterogeneous configurations, e.g., different processing speeds



Note: At each time step t, a **unique focused task** is automatically given by the system

Introduction

Why is DWS Challenging?

1. Flexible Task Assignment Across Heterogeneous Machines

- Real-world cloud environments are **heterogeneous** with machines of varying configurations
- It is crucial to **intelligently** allocate tasks to the **most suitable** machines
- Ignoring heterogeneity leads to inefficient resource use and longer workflow flowtimes

2. Unpredictable Workflow Arrivals and Patterns

- Workflows arrive in real time and constantly change in amount and patterns
- Need to consider the **complex relationship** between newly arrived, ongoing, and completed workflows

3. Rapidly changing environments

- System workload and resource status are constantly changing
- Necessitates **real-time** decisions-making
- Necessitates **adaptive** scheduling strategies to cope with environmental changes

Introduction

Limitations of Existing Approaches

Priority Dispatching Rules (PDRs)

- Hand-craft heuristic
- Fast, intuitive, and easy to implement
- Require extensive expertise and time-consuming tuning
- Unable for online adaption to newly collected data

Genetic Programming-based Hyper-Heuristic (GPHH)

- Automatically evolves **tree-based PDRs** through iterative evaluation-and-evolution
- State-of-the-art for DWS
- Unsuitable for online adaption to newly collected data

Deep Reinforcement Learning (DRL)

- Successfully learns **neural network-based PDRs** via RL
- Suitable for online adaption through fine-tuning
- Existing vector/matrix-based state representations fail to capture complex task–machine interactions in DWS

Related Work in Learning-to-Optimize (L2O)

- Unable to capture complex and dynamic relationships between workflows and machines.
- Neglecting the critic's role in Actor-Critic-based RL stability for large-scale problems
- Unable to continuously learn in the face of future environmental changes

	Graph Representations	Neural Network Architectures	Training Methods	Problem Scales
[1]	Static disjunctive graphs	Shared feature extractor	Unmodified Proximal Policy Optimization (PPO)	≤2,000 tasks
[2]	Static disjunctive graphs	Only one feature extractor	Unmodified REINFORCE	≤2,000 tasks
[3]	Static disjunctive graphs	Only one feature extractor	Self-supervised learning	≤2,000 tasks
Ours	Novel **dynamic** graphs	**Separate** feature extractor	Novel **offline-online** PPO	≤**600,000** tasks

[1] Zhang, C., Song, W., Cao, Z., Zhang, J., Tan, P. S., & Chi, X. (2020). Learning to dispatch for job shop scheduling via deep reinforcement learning. In *NeurIPS*.

[2] Zhang, C., Cao, Z., Song, W., Wu, Y., & Zhang, J. (2024). Deep reinforcement learning guided improvement heuristic for job shop scheduling. In *ICLR*.

[3] Corsini, A., Porrello, A., Calderara, S., & Dell'Amico, M. (2024). Self-labeling the job shop scheduling problem. In *NeurIPS*.

Our Approach – GOODRL

Overall Goal: Introduce **G**raph Assisted **O**ffline-**O**nline **D**eep **R**einforcement **L**earning (GOODRL) to learn an adaptive and intelligent **scheduling agent** for DWS.

Challenges

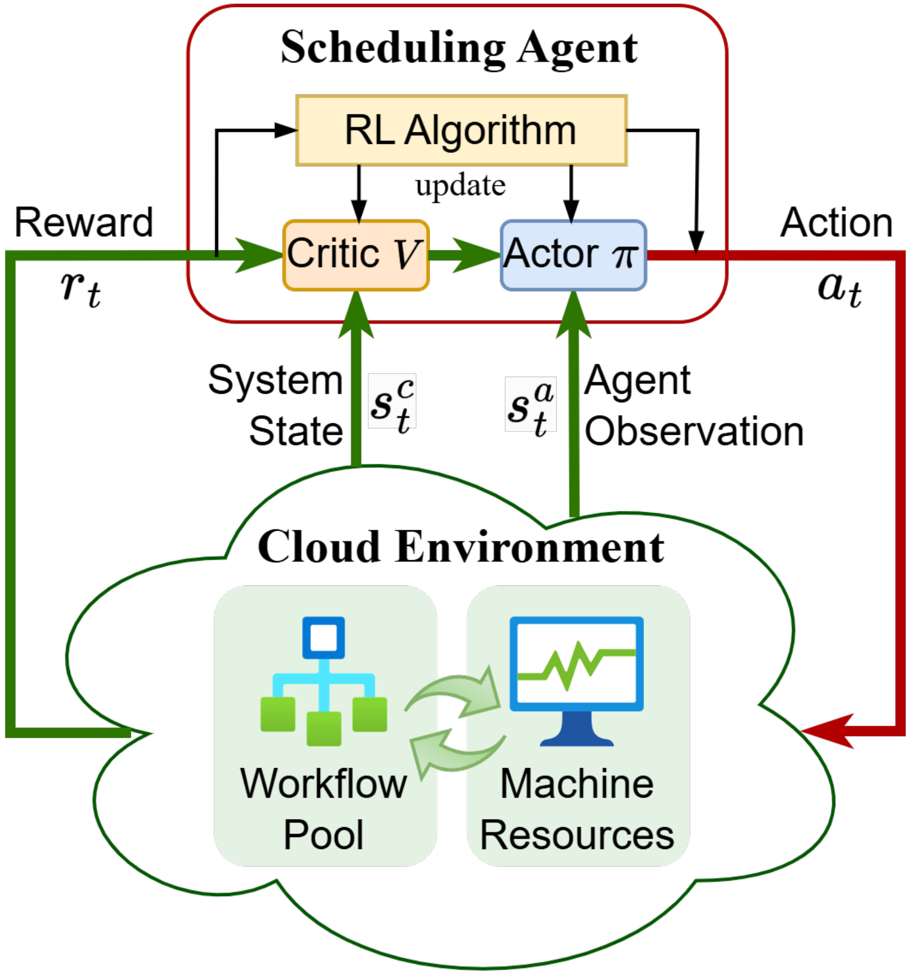
1. Flexible Task Assignment Across Heterogeneous Machines
2. Unpredictable Workflow Arrivals and Patterns
3. Rapidly changing environments

Key Innovations

- **Task-Specific Graph & Graph Attention Actor Network**
  - Precisely differentiate all eligible machines.
  - Explicitly captures the future impact of each machine on the current task at both topological and feature levels.
- **System-Oriented Graph & Graph Attention Critic Network**
  - Accurately capture real-time changes in the system state.
  - Seamlessly integrate newly arriving workflows with existing ones.
- **Offline-Online Training Method**
  - Offline imitation learning followed by standard PPO.
  - Online PPO with gradient control and decoupled high-frequency critic techniques.

Methodology

Overview of GOODRL

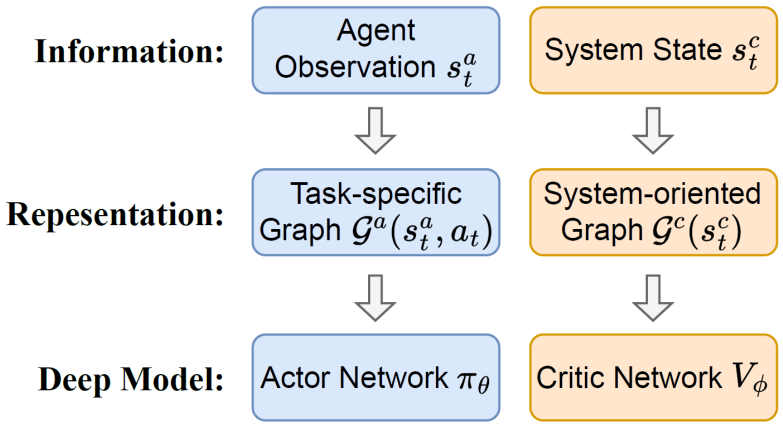


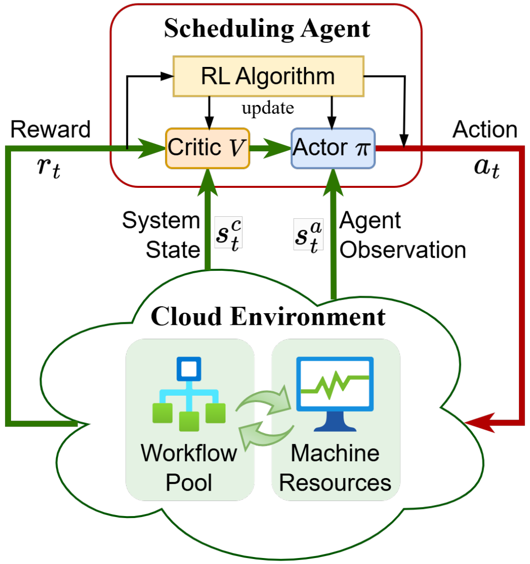
The goal of GOODRL is to learn an adaptive and intelligent **scheduling agent** for DWS.

- Step1: Formulate DWS as an RL problem
  (Innovation of graph representations)
- Step2: Graph Attention Actor & Critic Networks
  (Innovation of neural network architectures)
- Step3: Two-stage Offline-Online Learning
  (Innovation of training methods)

Methodology

Step1: Formulate DWS as an RL Problem



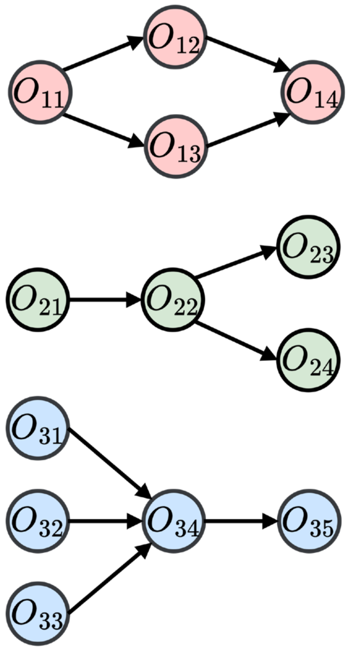


- **System State**: Snapshot of **the entire DWS system** at any time, including all tasks, machines, workflows, and their dependencies.
- **Agent Observation**: **Partial view** of the system from the agent's perspective, tailored for decision-making.
- **Action**: Assign the focused task to an eligible machine's waiting queue.
- **Transition**: Transit from state s_t to state s_{t+1} after an action is executed, updating workflow and machine information.
- **Rewards**: Defined as the negative normalized sum of workflow flowtimes completed between consecutive decision steps. The objective is $min\ \frac{1}{|\mathcal{W}|}\sum_{i=1}^{|\mathcal{W}|} F_i$.

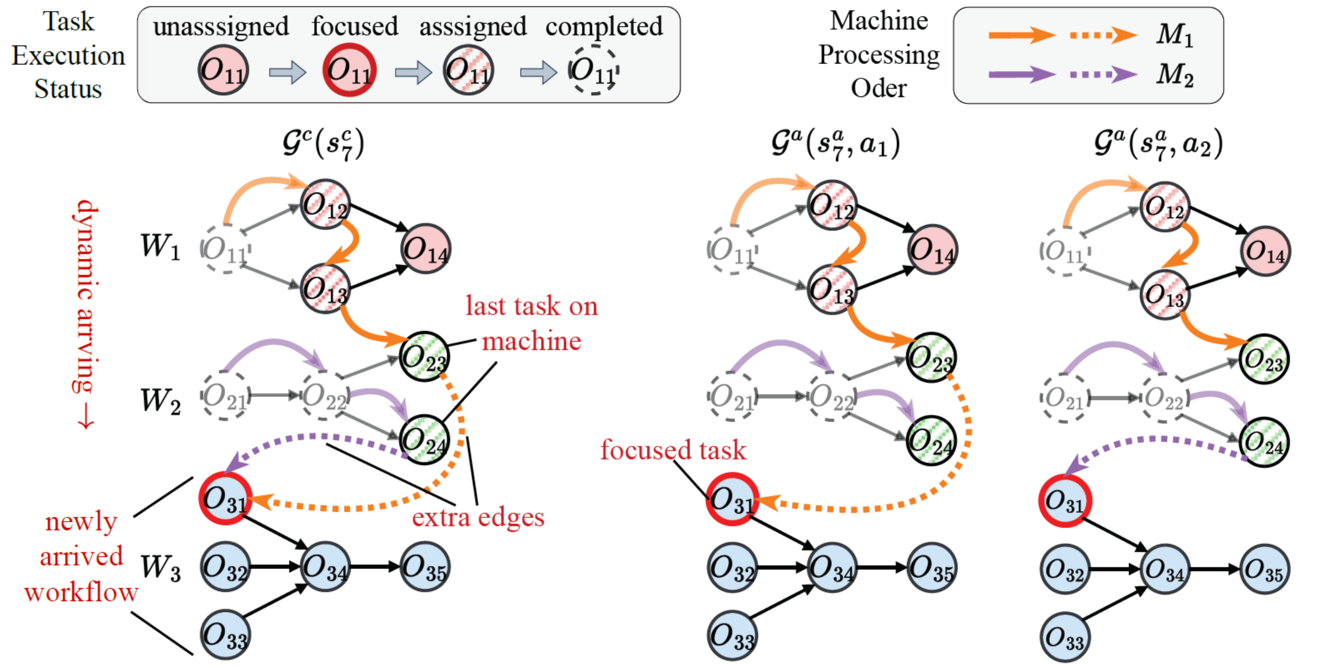
Methodology

Step1: Dynamic Graph Representations

Example: The **tasks** $\mathcal{O} = \{O_{ij}\}$ of **workflows** W_1, W_2, W_3 are assigned to **machines** M_1 and M_2.



At state s_7, should the focused task O_{31} to be assigned to machine M_1 or M_2?

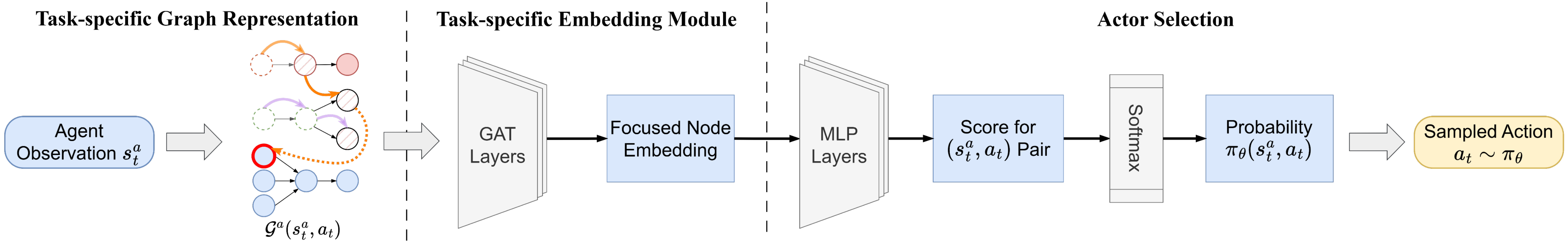


(a) System-oriented Graph Representation (b) Task-specific Graph Representation

- For critic network V_ϕ, represents the entire system state
- For actor network π_θ, focuses on task-machine interactions

Methodology

Step2: Actor Network Architecture



- **Pairwise Processing**: Evaluate each (s, a) pair separately, considering the immediate and future impact of assigning any machine to the focused task.
- **Focused Embedding**: Directly focus on the embedding of the focused task, rather than using mean pooling to combine embeddings of all nodes.

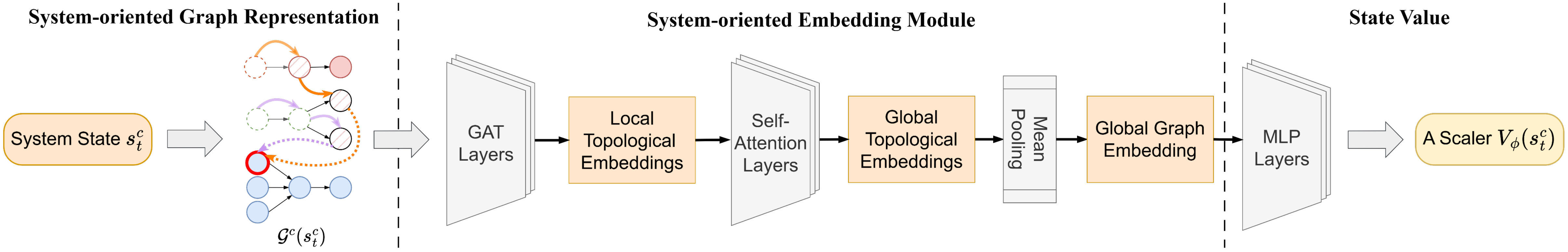
$$\mathcal{L}_{CE} = \frac{1}{|\mathcal{D}|} \sum_{s_t^a, a_t \in \mathcal{D}} \mathrm{CrossEntropy}(\pi_\theta(s_t^a, \cdot), a_t)$$

Ablation Study

Actor Architecture	100-th	200-th	300-th	400-th	500-th	600-th	700-th	800-th	900-th
Ours-TSEM	2.7486	**2.7106**	**2.6881**	**2.6647**	**2.6498**	**2.6038**	**2.5726**	**2.5297**	**2.5091**
TSEM w/o. pair	3.1707	3.1597	3.1538	3.1468	3.1435	3.1394	3.1365	3.1333	3.1302
TSEM w. mean	**2.7099**	2.7209	2.7152	2.6659	2.7109	2.6172	2.5989	2.5334	2.5243

Methodology

Step2: Critic Network Architecture



- **Comprehensive Context Awareness**: Process the information of each edge in bi-direction and use additional edges between the focused task and all eligible machines.
- **Long-range Interaction Modeling**: Use a self-attention mechanism to capture long-range dependencies across all task nodes, including those belonging to newly arrived workflows.

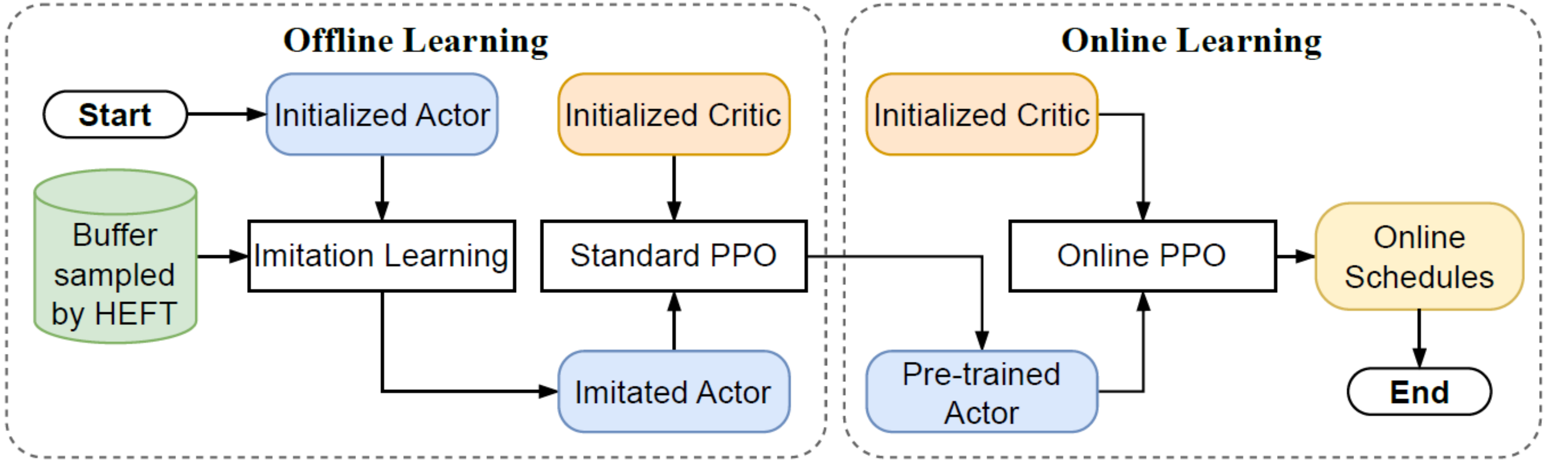
$$\mathcal{L}_{MSE} = \frac{1}{|\mathcal{D}|} \sum_{s_t^c \in \mathcal{D}} (V_\phi(s_t^c) - R_t)^2$$

Ablation Study

Critic Architecture	100-th	200-th	300-th	400-th	500-th	600-th	700-th	800-th
Ours-SOEM	**16.3971**	14.0938	**10.4907**	**9.5811**	**7.8581**	7.5675	**7.1238**	**6.0035**
SOEM w/o. edge	17.3012	**13.4737**	11.6626	9.8066	8.8853	**7.5266**	7.5607	7.593
SOEM w/o. self	20.6114	16.1826	14.6813	12.6997	12.0733	10.7019	10.1497	8.5121

Methodology

Step3: Two-stage Offline-Online Learning



- **Offline Phase**:
  - Pre-train actor network via imitation learning to mimic the behavior of experts (e.g., HEFT).
  - Use Proximal Policy Optimization (PPO) algorithm for joint actor-critic training.
- **Online Phase**:
  - Enhanced PPO with *gradient control* and *decoupled high-frequent critic updates*.

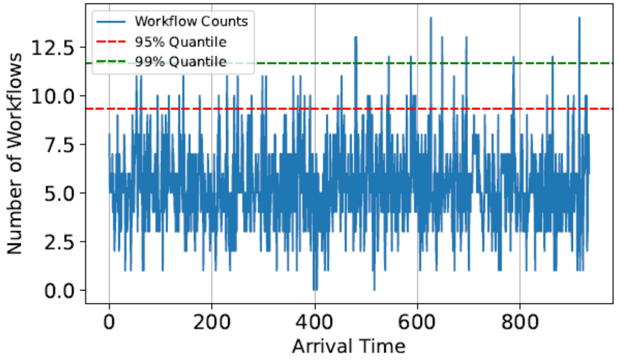
Ablation Study

Training Method	150-th	175-th	200-th	225-th	250-th
Ours-Online	**1.62%**	**1.50%**	**1.57%**	**1.52%**	**1.52%**
Online w/o. grad.	-1.18%	-1.08%	-1.24%	-1.36%	-1.64%
Online w/o. freq.	-184.80%	-261.27%	-283.93%	-336.86%	-382.54%

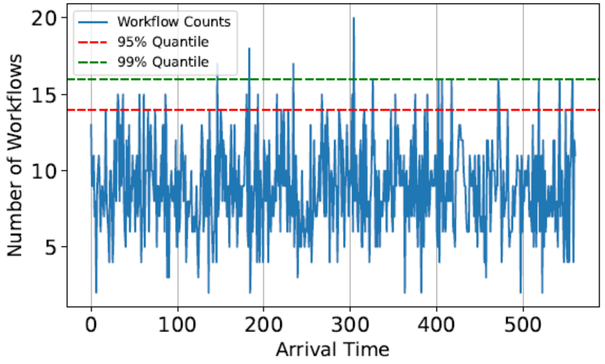
Experimental Setup

Environment Settings

- Workflow patterns: Montage, CyberShake, SIPHT, Inspiral
- Machines: 5 types $\times$ 5 each, 6 types $\times$ 4 each
- Arrival patterns: Poisson, $\lambda = \{5.4, 9\}$ workflows/hour



(a) $\lambda = 5.4$



(b) $\lambda = 9.0$

Baselines

- Traditional heuristics: **EST, PEFT, HEFT**
- Evolutionary computation approach: **GPHH** (30 independent runs)
- DRL-based approach: **ERL-DWS** (5 independent runs)

Model Configurations

- Actor network: 2 GAT layers and 4 MLP layers, with each of layer has 128 hidden-dimensions
- Critic network: 2 GAT layers, 1 self-attention layers, and 4 MLP layers, with hidden-dimension =128

Experiments

Offline Scenario Performance

Scenarios	EST Obj.	EST Gap	PEFT Obj.	PEFT Gap	HEFT Obj.	HEFT Gap	GPHH Obj.	GPHH Gap	ERL-DWS Obj.	ERL-DWS Gap	Ours-Offline Obj.	Ours-Offline Gap
$\langle 5\times 5, 5.4, 1k\rangle$	1243.15	204.51%	551.30	35.04%	509.95	24.91%	**408.24**	**0.00%**	1889.47	362.83%	413.29	1.24%
$\langle 5\times 5, 9, 1k\rangle$	1152.40	177.94%	510.55	23.14%	478.44	15.39%	430.28	3.78%	2180.41	425.89%	**414.61**	**0.00%**
$\langle 6\times 4, 5.4, 1k\rangle$	1083.02	290.07%	438.40	57.90%	391.61	41.05%	322.52	16.16%	713.87	157.11%	**277.65**	**0.00%**
$\langle 6\times 4, 9, 1k\rangle$	990.20	248.92%	391.17	37.84%	357.95	26.13%	300.20	5.78%	1523.83	436.95%	**283.79**	**0.00%**
$\langle 5\times 5, 5.4, 3k\rangle$	1235.14	202.87%	551.33	35.19%	508.10	24.59%	**407.81**	**0.00%**	2670.81	554.91%	408.41	0.15%
$\langle 5\times 5, 9, 3k\rangle$	1153.02	179.00%	510.22	23.46%	477.07	15.44%	427.04	3.33%	3582.70	766.91%	**413.27**	**0.00%**
$\langle 6\times 4, 5.4, 3k\rangle$	1081.28	289.98%	438.62	58.19%	390.64	40.89%	386.77	39.49%	1108.95	299.96%	**277.27**	**0.00%**
$\langle 6\times 4, 9, 3k\rangle$	992.46	250.72%	389.94	37.80%	356.08	25.83%	358.40	26.65%	2748.28	871.19%	**282.98**	**0.00%**
$\langle 5\times 5, 5.4, 5k\rangle$	1231.70	202.34%	550.53	35.13%	507.91	24.67%	408.38	0.24%	2944.35	622.73%	**407.39**	**0.00%**
$\langle 5\times 5, 9, 5k\rangle$	1146.62	177.17%	509.61	23.19%	477.12	15.33%	427.88	3.43%	4299.75	939.38%	**413.68**	**0.00%**
$\langle 6\times 4, 5.4, 5k\rangle$	1076.75	288.11%	437.53	57.71%	389.24	40.30%	386.95	39.47%	1281.00	361.73%	**277.44**	**0.00%**
$\langle 6\times 4, 9, 5k\rangle$	992.92	250.55%	388.68	37.22%	356.47	25.85%	297.40	5.00%	3480.87	1128.92%	**283.25**	**0.00%**
	5.08		4		2.92		1.92		5.92		**1.17**	

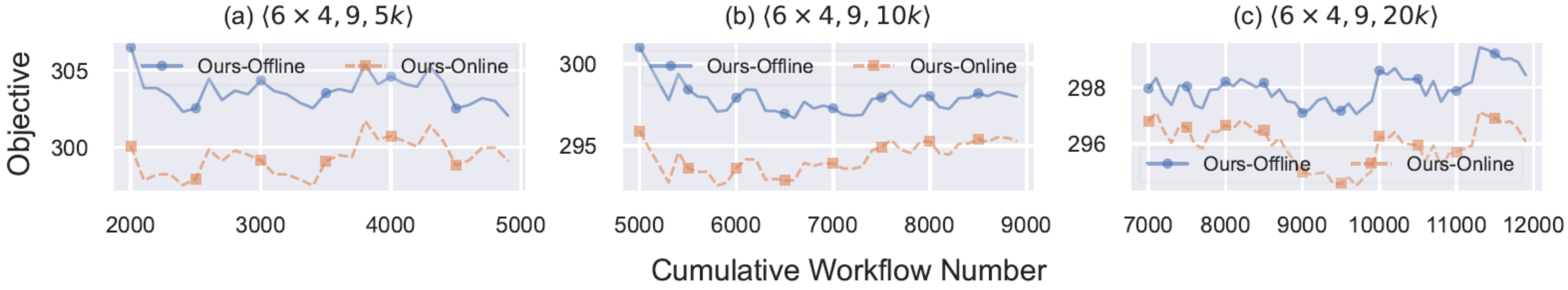
Observations

- **GOODRL** achieves the **lowest** *mean flowtime* in most offline scenarios
- Outperforms heuristics by **up to 290.07%**
- More **robust** performance than GPHH and ERL-DWS

Experiments

Online Scenario Performance

Scenarios	EST Obj.	EST Gap	PEFT Obj.	PEFT Gap	HEFT Obj.	HEFT Gap	GPHH Obj.	GPHH Gap	ERL-DWS Obj.	ERL-DWS Gap	Ours-Offline Obj.	Ours-Offline Gap	Ours-Online Obj.	Ours-Online Gap
$\langle 6\times 4, 5.4, 5k\rangle$	1076.01	277.05%	439.28	53.93%	391.63	37.23%	303.70	6.42%	1349.12	372.74%	286.43	0.37%	**285.38**	**0.00%**
$\langle 6\times 4, 5.4, 10k\rangle$	1077.09	279.13%	439.64	54.75%	390.26	37.37%	305.31	7.47%	1778.26	525.94%	**284.09**	**0.00%**	285.12	0.36%
$\langle 6\times 4, 5.4, 20k\rangle$	1072.90	276.97%	439.88	54.55%	391.18	37.44%	309.12	8.61%	2257.78	693.29%	286.08	0.52%	**284.61**	**0.00%**
$\langle 6\times 4, 9, 5k\rangle$	994.00	233.40%	387.84	30.09%	355.51	19.24%	303.57	1.82%	1246.91	318.24%	301.00	0.96%	**298.14**	**0.00%**
$\langle 6\times 4, 9, 10k\rangle$	993.97	238.09%	387.64	31.85%	355.21	20.82%	307.27	4.52%	1838.20	525.24%	297.19	1.09%	**294.00**	**0.00%**
$\langle 6\times 4, 9, 20k\rangle$	997.53	231.28%	388.79	29.12%	356.39	18.36%	312.56	5.08%	2783.78	835.93%	301.11	1.24%	**297.44**	**0.00%**
	6		5		4		3		7		1.83		**1.17**	



Observations

- GOODRL-Online **further improves** scheduling performance upon GOODRL-Offline
- Demonstrates effective online adaptation even in **large-scale scenarios** (e.g., 20k workflows)

Experiments

Scalability & Transferability

Scalability to significant changes

Scenarios	Workflow Pattern	Arrival Rate	Machine Number	EST	PEFT	HEFT	GP	ERL-DWS	Ours
1	√	–	–	1954.59	961.26	881.55	962.35	14103.84	**862.59**
2	√	√	–	2114.21	1005.76	904.06	832.37	6403.65	**791.86**
3	–	√	3×15	1793.76	927.33	872.71	1015.96	3208.32	**761.24**
4	–	√	4×10	1512.44	684.15	643.34	517.05	2696.69	**509.17**
5	–	√	5×7	1317.28	561.51	513.70	396.07	2534.30	**385.44**
6	–	√	6×5	1190.84	450.93	404.47	286.00	2420.63	**282.07**

GOODRL can effectively handle **significant changes** in workflow patterns, arrival rates, and machine configurations without retraining

Transferability to FJSS

FJSS Size	MOR	SPT	FIFO	MWKR	DRL-G	DRL-S	Ours
10×5	116.69	129.06	119.62	115.29	111.67	**105.61**	112.57
20×5	217.17	229.89	216.13	216.98	211.22	207.50	**202.38**
30×10	320.18	347.40	328.50	319.89	313.04	312.20	**304.63**
40×10	425.19	443.30	427.22	425.70	416.18	415.15	**395.70**

GOODRL can also performs **competitively** on **other scheduling problems** such as FJSS [1]

[1] Song, W., Chen, X., Li, Q., & Cao, Z. (2022). Flexible job-shop scheduling via graph neural network and deep reinforcement learning. *IEEE Transactions on Industrial Informatics*.

Experiments

Extensibility & Inference Time

Extensibility to multi-objective problems

Scenarios	Objectives	Single-Obj.	Multi-Obj.	Diff.
$\langle 5 \times 5, 5.4, 30 \rangle$	*flowtime*	401.77	420.29	+4.61%
	cost	139.82	82.28	-41.15%
$\langle 5 \times 5, 5.9, 30 \rangle$	*flowtime*	408.49	413.02	+1.11%
	cost	116.32	97.51	-16.17%
$\langle 6 \times 4, 5.4, 30 \rangle$	*flowtime*	277.57	286.73	+3.30%
	cost	192.24	143.47	-25.37%
$\langle 6 \times 4, 9, 30 \rangle$	*flowtime*	285.93	306.90	+7.33%
	cost	135.58	91.18	-32.75%

GOODRL can **support other practical objectives**, such as *cost* and *flowtime*, by modifying the reward function

Average inference time to make a decision

Scenarios	GPHH	ERL-DWS	Ours
$\langle 5 \times 5, 5.4, 1k \rangle$	0.7 ms	2.6 ms	6.1 ms
$\langle 5 \times 5, 9, 1k \rangle$	1.0 ms	2.7 ms	7.6 ms
$\langle 6 \times 4, 5.4, 1k \rangle$	0.6 ms	2.7 ms	6.0 ms
$\langle 6 \times 4, 9, 1k \rangle$	0.7 ms	2.5 ms	6.8 ms

GOODRL's inference time is less than the communication latency and data transfer time in cloud, hence **short enough** to meet real-world requirements

Conclusion

Contributions

- **Task-Specific Graph Representation & Graph Attention Actor Network**:
  Dynamically evaluate both immediate and future impacts among tasks, workflows, and machines.
- **System-oriented Graph Representation & Graph Attention Critic Network**:
  Model complex interactions across multiple workflows and machines for accurate value estimation.
- **Offline Imitation Learning & Enhanced Online PPO**:
  Efficient pre-training with imitation learning, followed by robust fine-tuning via gradient control and decoupled high-frequency critic updates.
- **Superior performance** compared to state-of-the-art baselines in minimizing mean flowtime.

Future Work

- Extend to more complex cloud environments (e.g., Unlimited machine configurations)
- Develop multi-objective learning techniques (e.g., Pareto-optimal learning)
- Incorporate constraint handling mechanisms (e.g. Learning an additional constraint control policy)