Automated Guided Vehicle Routing in the Container terminals: A MARL Approach

Sirui Ding, Zhaowei Liang, Chenglin Li, Fanghan Liu

Introduction

- **Automated Guided Vehicle** (AGV) is one of the commonly used equipment in warehouses and terminals. Typically, there are multiple AGVs within a facility, making it an important issue to coordinate all AGVs to work together effectively. In our work, we have adopted the **Multi-Agent Reinforcement Learning** (MARL) approach to find the optimal solution for AGV operation. Specifically, we created a warehouse environment where inbound and outbound orders may randomly appear, and established corresponding rewards and penalties for different AGV behaviors such as picking up, delivering, colliding, and going out of bounds. Building on this, we applied the QMIX algorithm, allowing AGVs to autonomously decide whether to pick up and the path of movement based on observed state information and reward signals. Our model achieved excellent results in smaller-scale environments (for up to 5 AGVs) and reaching efficiency comparable to mathematical programming and assignment, but exhibited instability and some weird behaviors in larger settings.

Motivation

- AGV is used to carries the goods between storage points in warehouses and terminal, which is the key elements to achieve smart terminal or smart warehousing. Therefore, a productive cooperative path control and action selection is necessary for each AGV in a terminal. However, current AGV technologies are still mainly based on automatic control and mathematical programming, where human prior knowledge about paths, efficiency, plays a significant role. To further improve efficiency and approach the global optimal solution, we hope to adopt reinforcement learning-based methods for AGV action selection and path planning. Specifically, drawing on knowledge and intuition from game theory, we wish to examine how, when each AGV possesses intelligence, they would cooperate based on rewards and states to efficiently accomplish the target tasks.

Problem Formulation

- **Environment**: We model the terminal as a grid world, where each unit of length can be seen as the length of one "step" of AGV. AGVs are free to move around the facility, and upon reaching storage points, if there is cargo waiting to be transported, they will accept the order, obtaining both the cargo and the coordinate of the cargo's destination. Similarly, when AGVs reach the destination point, they successfully complete the transport and re-enter an unladen state.
- **Action Space**: There are five possible actions for AGVs, including moving up, down, left, right, and remaining stationary (in case when AGVs might go out of the field or collide with others).
- **Reward Function**: There are 2 types of rewards and 2 types of punishments for AGV behaviors, as shown in the table below:

Table: Rewards for Different Actions

Behaviors	Accepting Order	Finishing Order	Crossing Border	Collision
Rewards	$r_1 > 0$	$r_2 > r_1$	$p_1 < 0$	$p_2 < 0$

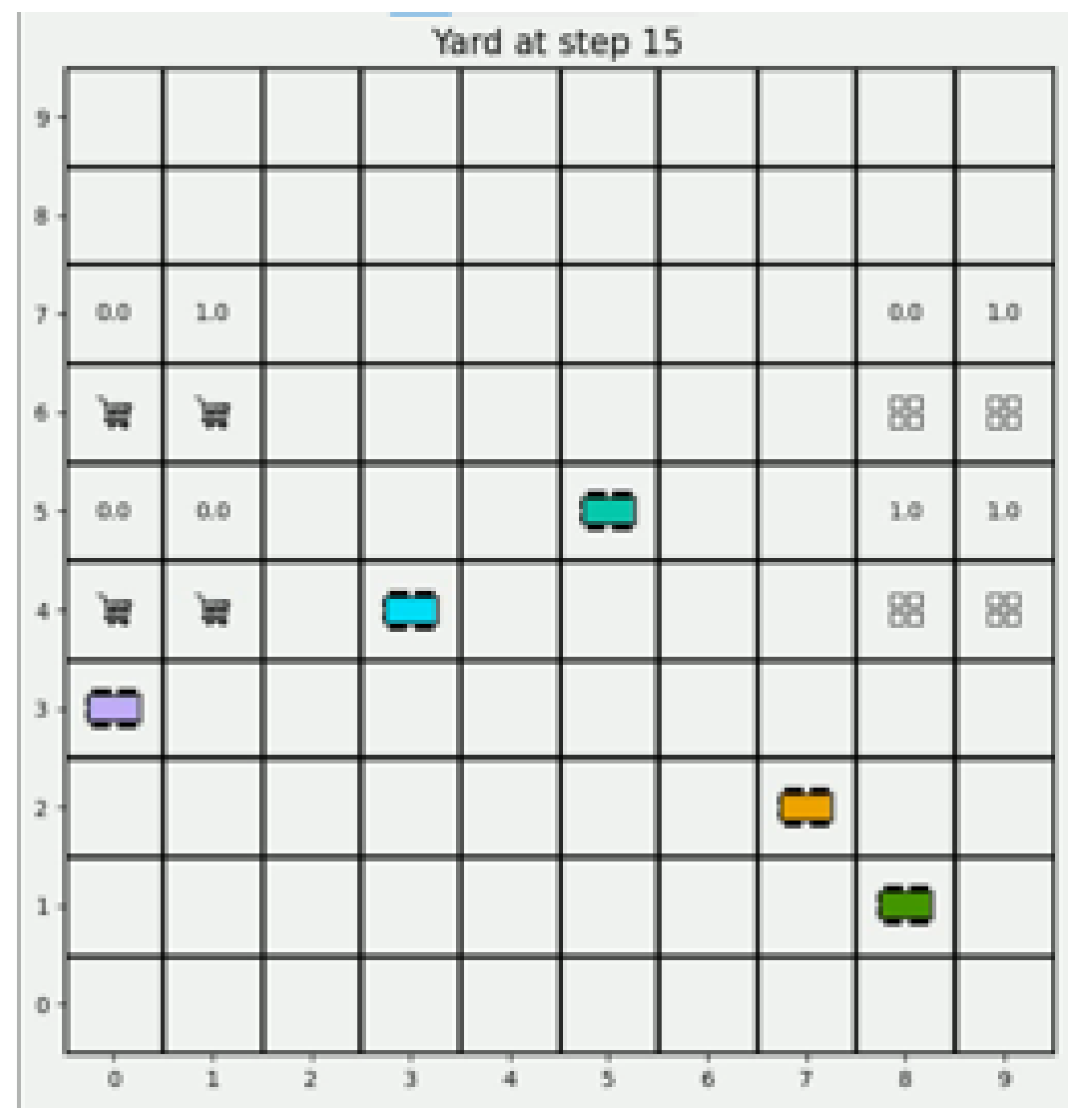

Figure: Grid World Environment for AGVs

Method Details

- **Algorithm**: We use QMIX algorithm to complete task assignment and path planning for AGVs simultaneously. QMIX is a value-based multi-agent method composed of RNN networks for agents to evaluate their own Q value and a MLP structure to evaluate the total value of the current step based on the state and individual Q values. Each agent will choose actions respectively based on their individual Q values.

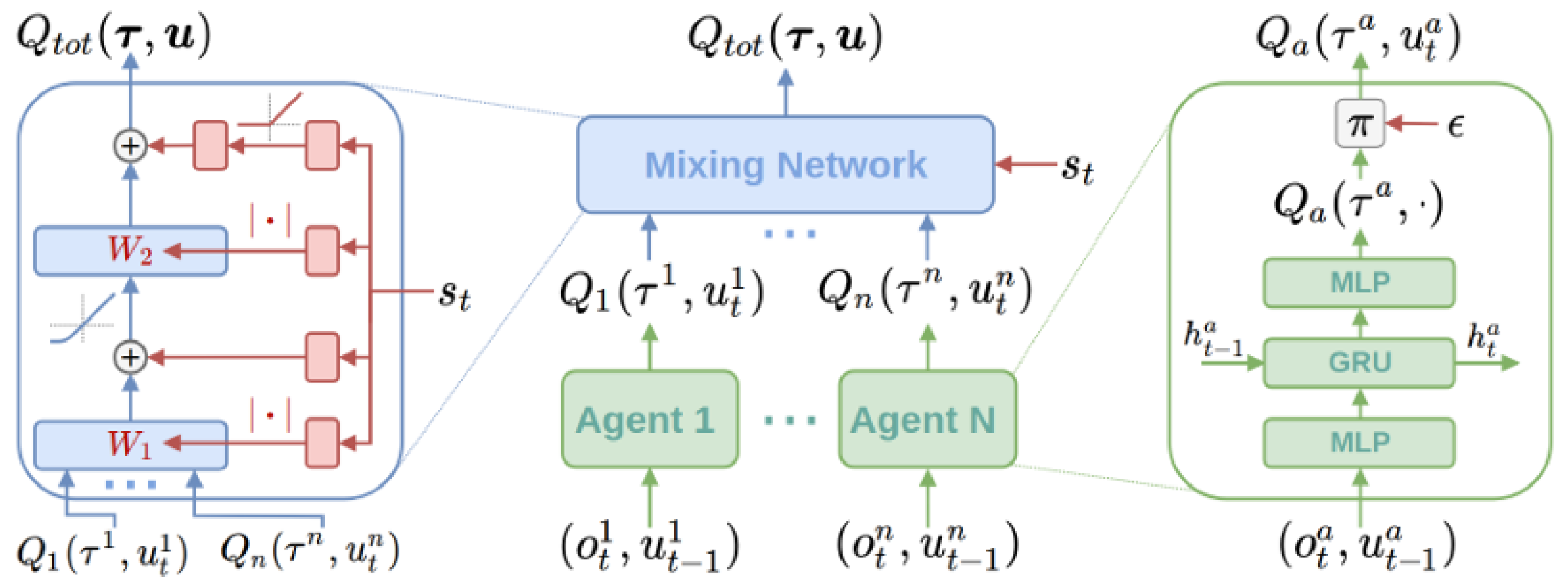

Figure: The Structure of QMIX

- **States**: In the direct method, grid maps of the environment is used as states, which can provide comprehensive information of tasks and positions. In comparison, in order to reduce the dimension of input, the region of observation for each agent only contains its surrounding map (grids it can reach within 3 steps, see figure below) and the direction of its target point. To further improve generalization abilities of our algorithm, in additional experiments, we extracted abstract features from grid maps and limited the agents' observation to the reward information of states in its adjacent map.

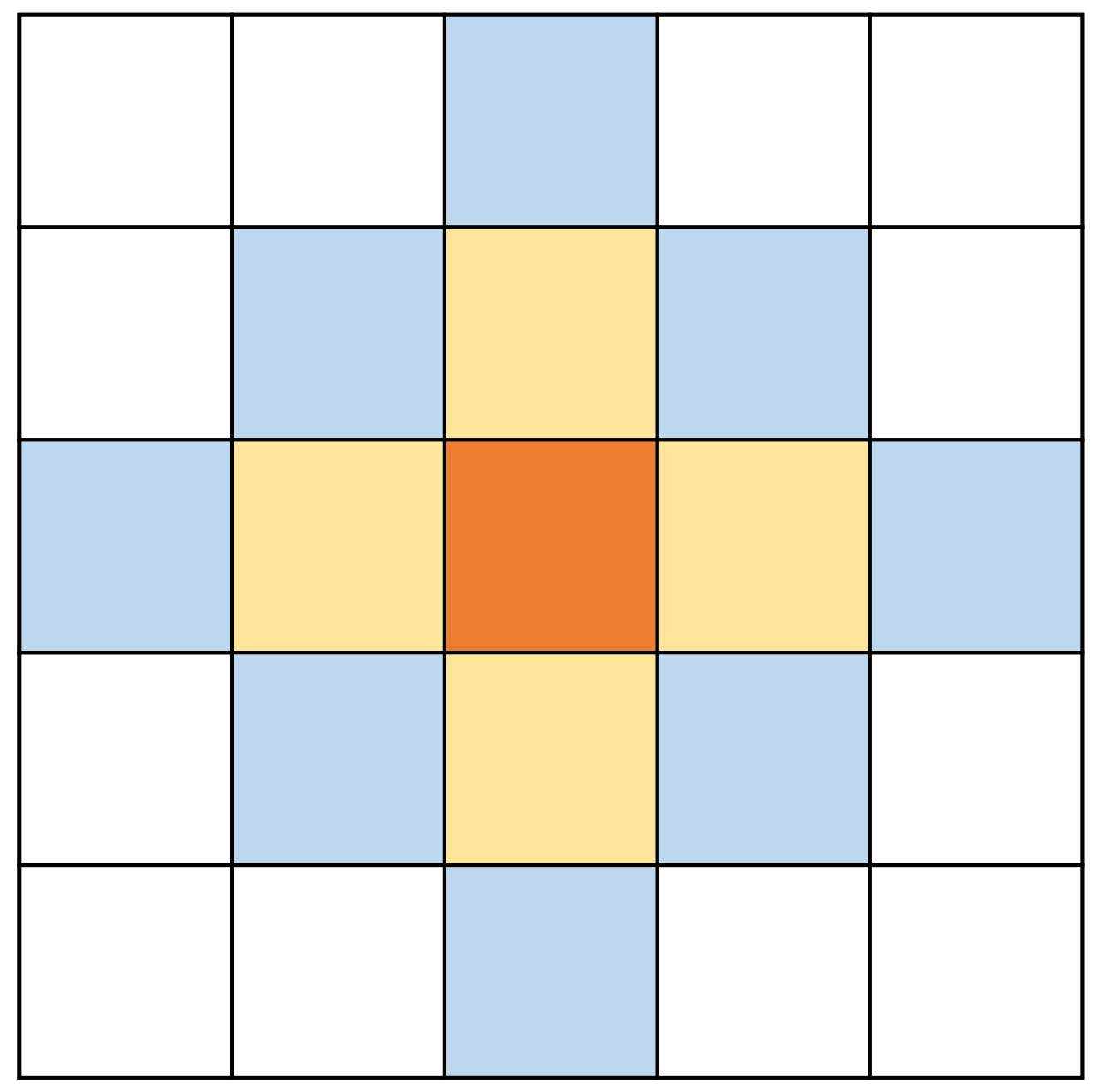

Figure: The Surrounding Map

Experiments and Results

- We conducted experiments to demonstrate feasibility of proposed methods. The following figures display results of experiments in a small-scale environment, with a 10 × 10 map, 2 packaging blocks and 2 AGVs. Empirical results suggest that in this case, the direct method does not suffer from the curse of dimensionality and performs better than the method with abstraction.

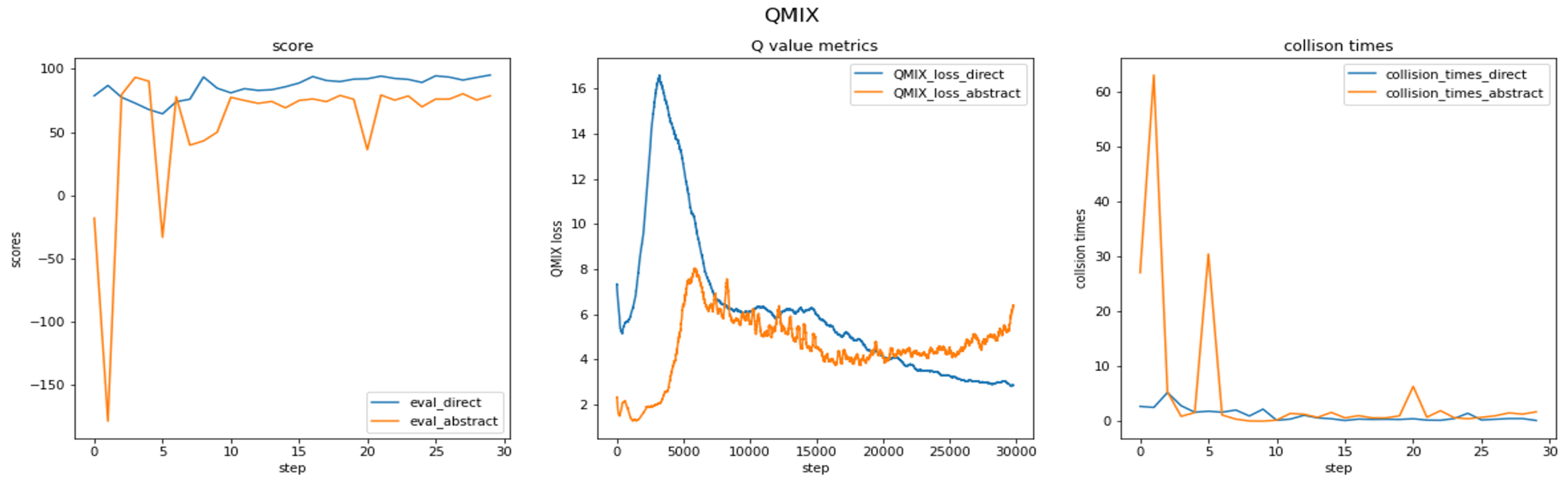

Figure: Training Curves in the Small Model

- The following figures display results of experiments in a medium-scale environment with 5 AGVs and 18 packaging blocks. It reveals that abstraction has positive effect in this more complicated setting, reducing training loss in QMIX training and number of collisions simultaneously.

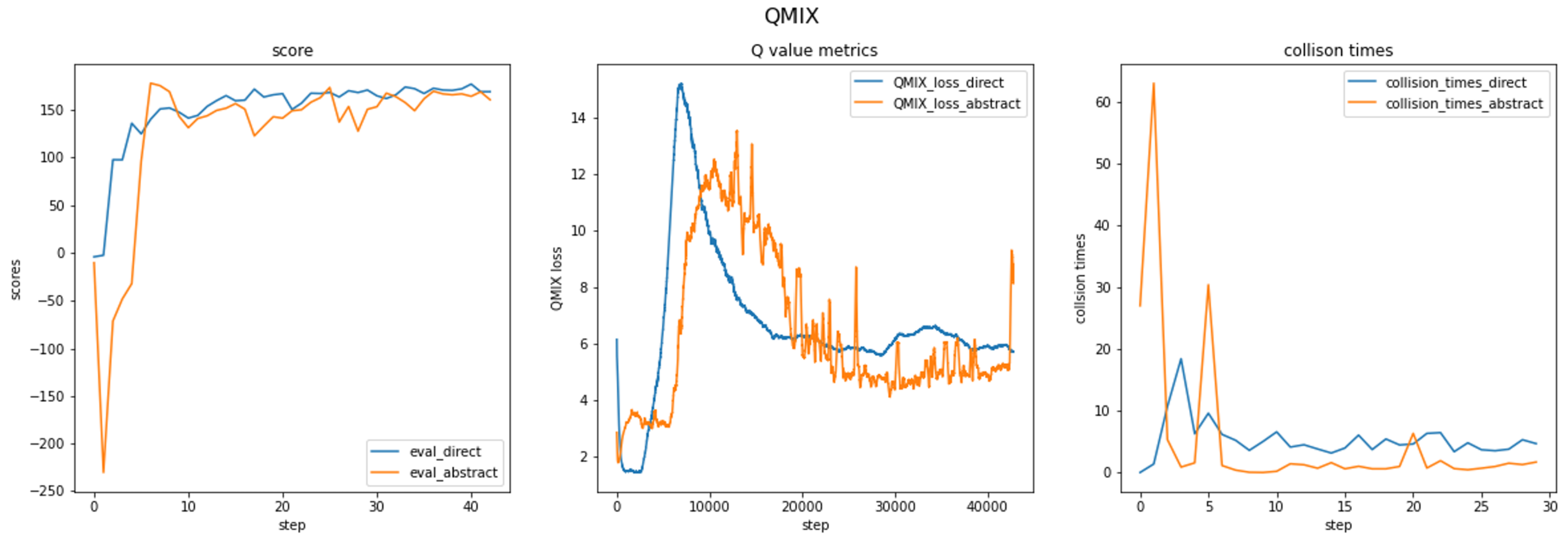

Figure: Training Curves in the Medium Model

Conclusions and Limitations

- Our results demonstrate that the proposed method can achieve high efficiency and successful task completion in small and medium-scale environments with up to 5 AGVs. However, we identified some limitations when scaling up to larger environments, where the model's performance becomes unstable and the time to complete tasks increases substantially.
- Future work can try to overcome existing limitations by exploring methods to facilitate agent communication and improving the scalability of the model. Enhancing the observable state dimensions and network complexity may bring potential improvements, but it will be critical to take careful considerations regarding computational efficiency and resource management. Additionally, investigating alternative algorithms or hybrid approaches could provide more robust solutions for larger-scale applications.