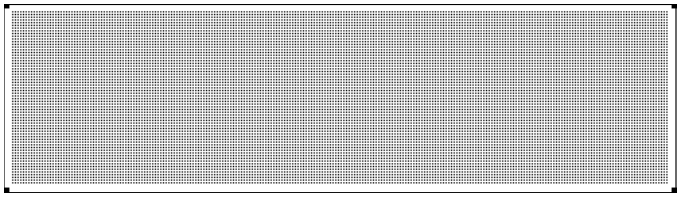
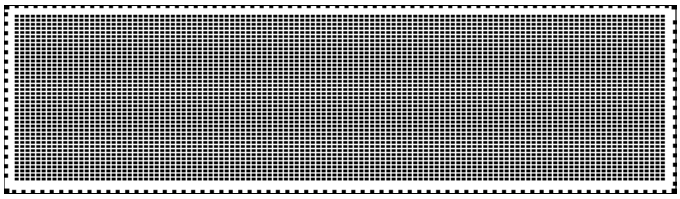
APPENDIX

A. Visualization of all maps

We visualize all the large maps for evaluation in fig. 6 and all the down-scaled small maps for training in fig. 7, which keep the obstacle patterns in the corresponding large maps.

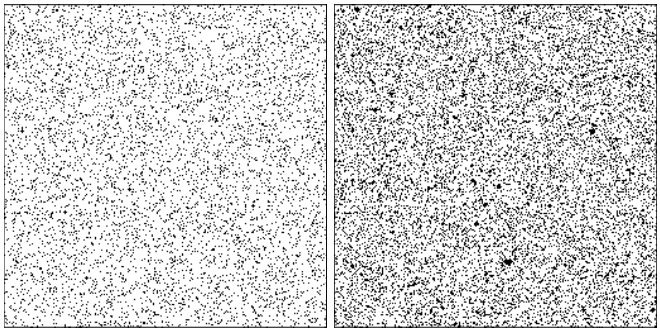


(a) Sortation



(b) Warehouse

(c) Paris (d) Berlin

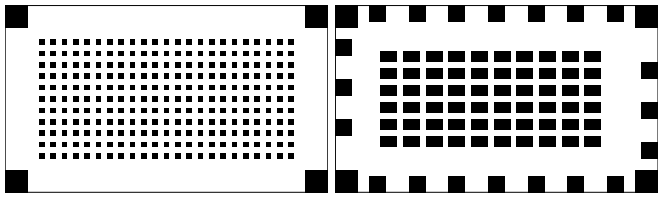


(e) Random1 (f) Random2

Fig. 6: Large maps with $10,000$ agents for evaluation.

B. Evaluation with Different Numbers of Agents

In this section, we compare Learnable PIBT and PIBT with different global guidance and different numbers of agents. The conclusions are similar to the ones in section V. With the same global guidance, Learnable PIBT consistently outperforms PIBT, proving the effect of learning. Also, different global guidance excels in different scenarios. An interesting observation is that in map Random2, Backward Dijstraj (BD) performs better with $< 10,000$ agents, and Dynamic Guidance (DG) performs better with $> 10,000$ agents. The reason may be that with more agents, there is potentially more congestion, and DG addresses congestion better than BD.



(a) Sortation (b) Warehouse

(c) Paris (d) Berlin

(e) Random1 (f) Random2

Fig. 7: Small maps with 600 agents for training.

C. Evaluation on Learn-to-Follow Benchmark

This section compares different decentralized methods with the Learn-to-Follow Benchmark [12] to validate the superiority of our SILLM (Learnable PIBT). Specifically, we compare Learnable PIBT trained with imitation learning and with reinforcement learning, Follower [12], SCRIMP [10], and PIBT [5]. For a simple comparison, we only use Backward Dijstraj as global guidance. All the training settings are the same as the ones in the Follower paper [12]. Notably, Learnable PIBT and Follower are trained on 40 Mazes maps and tested on 10 test maps and other maps. Our Learnable PIBT trained with imitation learning performs the best consistently across 4 different kinds of maps. Notably, Follower actually only outperforms PIBT in Mazes maps but may fail to outperform PIBT in other maps, which means the

generalization ability of Follower still needs improvement. In contrast, our Learnable PIBT is much more generalizable.

D. Real-World Mini Example

Since during the planning process, our algorithm assumes the position of all agents to be perfectly known at all times, we use ground truth positions for our virtual robots and use external localization (here, the Optitrack Motion Capture System) to obtain accurate position information for our real robots. However, the planned path may not be executed accurately due to disturbances and control inaccuracies. To eliminate these errors, we implement an Action Dependency Graph (ADG) [32], [33].

The video demo is available in the supplementary material. From our experiment with 10 real agents, we observe that agents can reach their goals quickly without collisions, and errors are eliminated by the ADG, demonstrating the potential of using our method in the real world. In our experiment with 100 virtual agents, we compare PIBT with Learnable PIBT. We can observe that Learnable PIBT outperforms PIBT with 50% more throughput.

E. Computation Resources

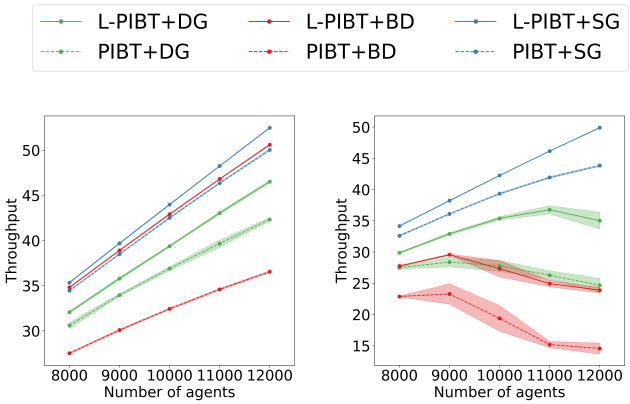
Our models are trained on servers with 72 vCPU AMD EPYC 9754 128-Core Processor, 4 RTX 4090D (24GB), and 240 GB memory. Training on each map takes less than 12 hours.

F. Reinforcement Learning Implementation

In the ablation study (Section V-B), we show that Imitation Learning is better than simple MAPPO-based Reinforcement Learning. Specifically, we use the following reward function.

$$r(v, v') = h(v) - h(v') - 1 \tag{1}$$

where h is the heuristic function defined in Section IV-B, v and v' are the current and next locations of an agent. Take Backward Dijkstra heuristics as an example. If v' is 1-step closer to the goal, the reward will be 0. Otherwise, the reward will be a negative penalty. If no other agents act as obstacles, the agent should follow its shortest path given this reward function after learning. Reward design is crucial to the performance of RL and worth further study.

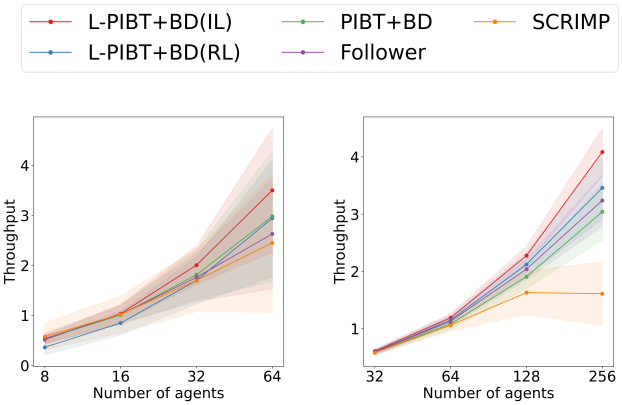


(a) Sortation (b) Warehouse

(c) Paris (d) Berlin

(e) Random1 (f) Random2

Fig. 8: Evaluation with different numbers of agents.



(a) Random 20*20 (b) Mazes 65*65

(c) den520d 64*64 (d) Paris_1 64*64

Fig. 9: Evaluation on Learn-to-Follow Benchmark.

TABLE IV: The comparison of PIBT and L-PIBT with different guidance. The left part is the result of downscaled small instances. The right part is the result of the original large instances. We evaluate each instance with 8 runs of different starts and goals. Each column records the mean throughput with standard deviation in the parentheses. The Time (in seconds) and Score refers to the average single-step planning time and the average score defined in Section V-A.

Algorithm	Small maps with 600 agents								Large maps with 10,000 agents							
	Sort	Ware	Pari	Berl	Ran1	Ran2	Time	Score	Sort	Ware	Pari	Berl	Ran1	Ran2	Time	Score
PIBT [5]	7.79 (0.36)	4.62 (0.10)	5.59 (0.15)	5.12 (0.35)	10.84 (0.08)	6.80 (0.48)	0.004	0.60	32.44 (0.10)	19.39 (2.04)	15.43 (0.34)	15.09 (0.26)	46.51 (0.08)	29.08 (1.26)	0.014	0.61
L-PIBT	14.76 (0.38)	8.69 (0.33)	7.76 (0.17)	7.16 (0.62)	**12.73** (0.11)	**10.76** (0.12)	0.007	0.89	42.91 (0.07)	27.34 (1.31)	18.84 (0.20)	19.16 (0.33)	**53.74** (0.10)	**44.02** (0.90)	0.024	0.80
PIBT+SG	13.66 (0.22)	9.91 (0.24)	6.18 (0.19)	4.50 (0.74)	11.66 (0.12)	7.46 (0.37)	0.004	0.74	42.51 (0.10)	39.34 (0.11)	18.11 (0.34)	17.62 (0.26)	46.89 (0.14)	25.57 (0.51)	0.014	0.75
L-PIBT+SG	**16.52** (0.09)	**14.28** (0.12)	**8.85** (0.11)	**7.19** (0.14)	12.34 (0.11)	10.09 (0.10)	0.007	0.98	**43.98** (0.07)	**42.24** (0.05)	22.54 (0.17)	21.59 (0.23)	52.49 (0.10)	32.44 (0.31)	0.023	0.85
PIBT+DG (TrafficFlow [6])	11.78 (0.20)	9.10 (0.42)	7.44 (0.24)	5.70 (0.43)	10.35 (0.37)	8.48 (0.10)	0.016	0.76	36.89 (0.22)	27.78 (0.88)	27.51 (0.69)	27.03 (0.54)	45.62 (0.84)	33.64 (2.32)	0.570	0.81
L-PIBT+DG	14.41 (0.16)	11.67 (0.54)	8.34 (0.07)	6.77 (0.28)	11.14 (0.13)	9.18 (0.14)	0.029	0.88	39.38 (0.09)	35.39 (0.26)	**31.00** (0.89)	**30.52** (0.98)	50.48 (0.14)	44.37 (0.11)	0.721	0.94

G. Detailed Comparison for Different Guidance

We report detailed throughput and average running time for comparing PIBT and L-PIBT with different guidance in Table IV. Notably, in contrast to the conclusion in the large map, Static Guidance works the best among the three heuristics in the small map for training. Overall, since all the heuristics are manually designed, the best heuristic is instance-dependent and should be evaluated empirically. Also, these heuristics only define the framework, but there could be a lot of hyperparameters that affect the performance. Automatic hyperparameter tuning can be helpful but doesn't necessarily remove the structural bias in the framework.