Spatially Grouped Curriculum Learning for Multi-Agent Path Finding

Thomy Phan[1], **Sven Koenig**[2,3]
[1]University of Bayreuth, Germany
[2]University of California, Irvine, USA
[3]Örebro University, Sweden
thomy.phan@uni-bayreuth.de, sven.koenig@uci.edu

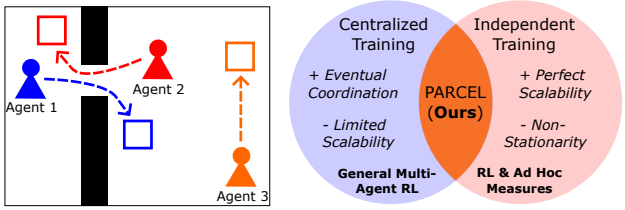
Abstract

Multi-agent path finding (MAPF) is the challenging problem of finding conflict-free paths with minimal costs for multiple agents. While traditional MAPF solvers are centralized using heuristic search, *reinforcement learning (RL)* is becoming increasingly popular due to its potential to learn decentralized and generalizing policies. RL-based MAPF must cope with *spatial coordination*, which is often addressed by combining independent training with ad hoc measures like replanning and communication. Such ad hoc measures often complicate the approach and require knowledge beyond the actual accessible information in RL, such as the full map occupation or broadcast communication channels, which limits generalizability, effectiveness, and sample efficiency. In this paper, we propose *Partitioned Attention-based Reverse Curricula for Enhanced Learning (PARCEL)*, considering a bounding region for each agent. PARCEL trains all agents with overlapping regions jointly via self-attention to avoid potential conflicts. By employing a reverse curriculum, where the bounding regions grow as the policies improve, all agents will eventually merge into a single coordinated group. We evaluate PARCEL in two simple coordination tasks and four MAPF benchmark maps. Compared with state-of-the-art RL-based MAPF methods, PARCEL demonstrates better effectiveness and sample efficiency without ad hoc measures.

Code — github.com/thomyphan/spatial-curricula-mapf

1 Introduction

A wide range of real-world applications like goods transportation in warehouses, smart manufacturing, and traffic management can be formulated as *Multi-Agent Path Finding (MAPF)* problem, where the goal is to find conflict-free paths for multiple agents with minimal costs (Li et al. 2021b; Zhang et al. 2023). Finding optimal solutions w.r.t. flowtime or makespan is NP-hard (Stern et al. 2019; Ratner and Warmuth 1986). Despite the problem complexity, there exists a variety of MAPF solvers that find optimal (Sharon et al. 2012), bounded suboptimal (Cohen and Koenig 2016), or quick feasible solutions (Li et al. 2021a; Okumura 2023). Most traditional MAPF solvers are centralized and require global information, broadcast communication, and human knowledge, thus being expensive and not generalizable to diverse and uncertain scenarios (Sartoretti et al. 2019).



(a) Example Scenario (b) Training Paradigms

Figure 1: **(a)** Example of three agents with assigned goals (hollow squares). While agents 1 and 2 should be trained jointly to coordinate on passing the black wall, agent 3 can be trained independently to reach its goal. **(b)** Common multi-agent RL paradigms, with strengths and weaknesses, and PARCEL as a partially independent RL alternative.

Reinforcement learning (RL) is becoming increasingly popular due to its potential to learn decentralized and generalizing policies with limited prior knowledge. These policies must be capable of *spatial coordination*, where agents synchronize their joint movements, especially in shared regions like corridors, bottleneck locations, etc., as illustrated in Fig. 1a. Due to the combinatorial nature of the joint movements, many RL-based MAPF methods use *independent training* of agents with ad hoc measures like replanning and communication for reactive coordination (Skrynnik et al. 2024; Wang et al. 2023). Such ad hoc measures often complicate the approach and require knowledge beyond the actual accessible information in RL, such as the full map occupation or broadcast communication channels, which limits generalizability, effectiveness, and sample efficiency (Phan et al. 2025a).

Centralized training for decentralized execution (CTDE), on the other hand, offers a more general and principled approach to learn coordinated policies by incorporating global information only during training, e.g., in a laboratory or simulation environment, using value factorization or centralized critics (Foerster et al. 2018; Rashid et al. 2020; Yu et al. 2022). While CTDE has demonstrated impressive results in video game scenarios, it does not scale well in highly constrained tasks, such as MAPF, though (Phan et al. 2024a).

To address the tension between centralized and independent training for RL-based MAPF, as shown in Fig. 1b, we propose *Partitioned Attention-based Reverse Curricula for Enhanced Learning (PARCEL)* to provide a middle ground. PARCEL considers bounding regions for all agents and only trains them jointly via self-attention if their regions overlap to avoid potential conflicts. By employing a reverse curriculum, where the bounding regions grow as the policies improve, all agents will eventually merge into a single coordinated group. Our contributions are as follows:

- We propose spatial grouping of agents via bounding regions defined by a reverse curriculum method.
- We formulate PARCEL for coordinated and partially independent curriculum learning by considering the spatial groups via self-attention and actor-critic learning.
- We evaluate PARCEL in two simple coordination tasks and four MAPF benchmark maps. Our results are compared with state-of-the-art RL-based MAPF methods, demonstrating better effectiveness and sample efficiency without additional ad hoc measures.

Our paper focuses on the *machine learning aspect* and therefore excludes heuristic search additions to assess the general learning (in-)capabilities of common RL-based MAPF methods regarding spatial coordination in a fair way.

2 Background

2.1 Multi-Agent Path Finding (MAPF)

We regard *maps* as undirected and unweighted *graphs* $G = \langle \mathcal{V}, \mathcal{E} \rangle$, where vertex set $\mathcal{V}$ contains all possible locations and edge set $\mathcal{E}$ contains all possible transitions between adjacent locations. An *instance* I consists of a map G and a set of *agents* $\mathcal{D} = \{1, ..., N\}$ with each agent $i \in \mathcal{D}$ having a *start location* $p_{start,i} \in \mathcal{V}$, and a *goal location* $p_{goal,i} \in \mathcal{V}$. At every time step t, each agent $i \in \mathcal{D}$ can move along the edges in $\mathcal{E}$ or wait at its current location $p_{i,t}$ (Stern et al. 2019). MAPF aims to find a collision-free plan for all agents. A *plan* $P = \{\phi_1, ..., \phi_N\}$ consists of individual paths $\phi_i = \langle p_{i,0}, ..., p_{i,l(\phi_i)} \rangle$ per agent i, where $\{p_{i,t}, p_{i,t+1}\} \in \mathcal{E}$, $p_{i,0} = p_{start,i}$, $p_{i,l(\phi_i)} = p_{goal,i}$, and $l(\phi_i)$ is the length or *travel time* of path ϕ_i. The *(shortest path) distance* between two vertices $p, p' \in \mathcal{V}$ is denoted by $d(p, p') = d(p', p) \geq 0$.

We consider *vertex conflicts* $\langle i, j, p, t \rangle$ that occur when two agents $i, j \in \mathcal{D}$ occupy the same location $p \in \mathcal{V}$ at time step t and *edge conflicts* $\langle i, j, p, p', t \rangle$ that occur when two agents $i, j \in \mathcal{D}$ traverse the same edge $\{p, p'\} \in \mathcal{E}$ in opposite directions at time step t (Stern et al. 2019). A plan P is a *solution*, i.e., *feasible*, when it does not have any vertex/edge conflicts. We want to find a solution by minimizing the *flowtime* $\sum_{\phi \in P} l(\phi)$ or the *makespan* $max_{\phi \in P} l(\phi)$.

2.2 Multi-Agent RL (MARL)

For RL-based MAPF, we formulate the MAPF problem as a *stochastic game* $SG = \langle \mathcal{D}, \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \Omega \rangle$, where $\mathcal{D} = \{1, ..., N\}$ is the set of agents, $\mathcal{S}$ is a set of states $s_t = \langle p_{t,i}, ..., p_{t,N} \rangle$, $\mathcal{A} = \mathcal{A}_1 \times ... \times \mathcal{A}_N$ is the set of joint actions $a_t = \langle a_{t,1}, ..., a_{t,N} \rangle$ with $|\mathcal{A}_i| = max_{p \in \mathcal{V}} \{degree(p) + 1\}$, $\mathcal{T}(s_{t+1}|s_t, a_t)$ is the transition probability, $\mathcal{R}(s_t, a_t) = \langle r_{t,1}, ..., r_{t,N} \rangle \in \mathbb{R}^N$ is the joint reward with $r_{t,i}$ being the reward of agent $i \in \mathcal{D}$, $\mathcal{O}$ is a set of local observations $o_{t,i}$ for each agent i, and $\Omega(s_{t+1}) = o_{t+1} = \langle o_{t+1,1}, ..., o_{t+1,N} \rangle \in \mathcal{O}^N$ is the subsequent joint observation (Emery-Montemerlo et al. 2004). Each location in s_t is unique such that $p_{t,i} \neq p_{t,j}$ for each agent pair $i, j \in \mathcal{D}$ with $i \neq j$. The state transitions are deterministic, where a valid move action will change the location $p_{t,i}$ of each agent i to $p_{t+1,i}$ with $\{p_{t,i}, p_{t+1,i}\} \in \mathcal{E}$. Any attempt to move over a non-existent edge or cause a collision, i.e., a vertex or edge conflict, is treated as a wait action. The individual reward $r_{t,i}$ is $+1$ if agent i reaches $p_{goal,i}$, zero if it stays at its goal location $p_{goal,i}$, and -1 otherwise (Phan et al. 2024a). Each agent i can observe the state s_t through $o_{t,i}$, i.e., a local neighborhood around its location $p_{t,i}$, modeling limited sensors for decentralized decisions (Oliehoek and Amato 2016).

Each agent i maintains an action-observation *history* $\tau_{t,i} = \langle o_{0,i}, a_{0,i}, ..., a_{t-1,i}, o_{t,i} \rangle$. The joint policy is denoted as $\pi = \langle \pi_1, ..., \pi_N \rangle$ with *local policies* π_i, where $\pi_i(a_{t,i}|\tau_{t,i})$ is the action selection probability of agent i. Each local policy π_i can be evaluated with a *value function* $Q_i^\pi(s_t, a_t) = \mathbb{E}_\pi[\sum_{\alpha=0}^{T-1} r_{t+\alpha,i} | s_t, a_t]$ for all s_t and a_t with horizon $T > 0$. An *optimal joint policy* π^* is defined by:

$$\pi^* = \langle \pi_1^*, ..., \pi_N^* \rangle = \mathit{argmax}_\pi \mathbb{E}_{\pi,I}[\sum_{i \in \mathcal{D}} Q_i^\pi(s_0, a_0)] \quad (1)$$

which minimizes the expected flowtime for any instance I.

To learn optimal policies π_i^* in large state spaces, approximators $\hat{\pi}_{i,\theta}$ parameterized with θ, are trained with gradient ascent on an estimate of $J = \mathbb{E}_{\hat{\pi},I}[Q_i^\pi(s_0, a_0)]$. *Policy gradient methods* use gradients defined by (Sutton et al. 2000):

$$g = A_i^{\hat{\pi}}(s_t, a_t) \nabla_\theta log \hat{\pi}_{i,\theta}(a_{t,i}|h_{t,i}) \quad (2)$$

where $A_i^{\hat{\pi}}(s_t, a_t) = Q_i^{\hat{\pi}}(s_t, a_t) - b_i(s_t)$ is the *advantage* of agent i and $b_i(s_t)$ is its state-dependent baseline. *Actor-critic* approaches like A2C or *Proximal Policy Optimization (PPO)* (Schulman et al. 2017), where $\hat{\pi}_{i,\theta}$ serves as the *actor*, often approximate $\hat{A}_i \approx A_i^{\hat{\pi}_i}$ by replacing $Q_i^{\hat{\pi}}(s_t, a_t)$ with $\sum_{\alpha=0}^{T-1} r_{t+\alpha,i}$ and b_i with $\mathbb{E}_{\hat{\pi}_i,I}[Q_i^{\hat{\pi}}]$. $Q_i^{\hat{\pi}}$ can be approximated with a *critic* $\hat{Q}_{i,\omega}$, and parameters ω using value-based RL (Watkins and Dayan 1992; Mnih et al. 2015). We omit the parameters θ, ω and write $\hat{\pi}_i$, $\hat{Q}_i$ in the following.

Naive *independent RL* suffers from *non-stationarity*, which can lead to ineffective and uncoordinated policies (Laurent et al. 2011). To mitigate this issue, modern MARL uses CTDE, where training takes place in a laboratory or a simulator with access to global information to learn coordinated policies that can be executed independently under partial observability afterward (Foerster et al. 2018; Rashid et al. 2020). However, CTDE scales poorly in MAPF tasks (Fig. 1b) (Phan et al. 2024a; Skrynnik et al. 2024).

3 Related Work

3.1 Grouping in MARL

Agent grouping has been explored in modern MARL, where a centralized meta-policy is commonly used for the group

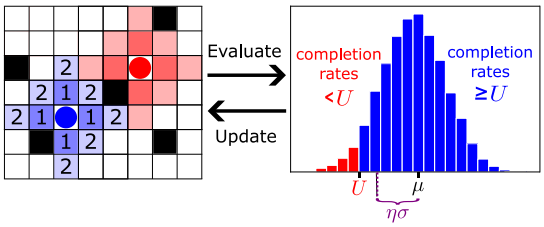


Figure 2: Curriculum scheme of CACTUS. The agents (colored circles) are trained and evaluated w.r.t. an allocation radius R_{alloc}, defining the shaded bounding regions $\mathcal{G}_i \subseteq \mathcal{V}$ around the agents. When the average completion rate μ exceeds a curriculum threshold U with a certain confidence such that $\mu - \eta\sigma \geq U$, then R_{alloc} is incremented by 1. The numbers in the blue cells $p \in \mathcal{G}_{blue}$ denote the shortest path distance $d(p, p_{start,blue})$ to the blue agent.

assignment (Li, Wang, and Xu 2025; Zang et al. 2023). Instead of training the groups separately, they are merely used for more tractable processing via CTDE, e.g., value factorization using *VAST* or *REFIL* (Iqbal and Sha 2019; Iqbal et al. 2021; Phan et al. 2021), to mitigate non-stationarity.

3.2 Machine Learning-Guided MAPF

Machine learning has been used in traditional centralized MAPF solvers to guide heuristic search (Alkazzi and Okumura 2024; Huang, Dilkina, and Koenig 2021; Huang et al. 2022; Kaduri, Boyarski, and Stern 2020; Phan et al. 2024b, 2025b). We focus on policy learning for *decentralized MAPF* without using any centralized MAPF solver.

3.3 Reinforcement Learning-Based MAPF

RL-based MAPF has become popular due to its potential to learn decentralized and generalizing policies with limited prior knowledge (Alkazzi and Okumura 2024; Sutton 2019). Due to the combinatorial nature of the joint movements, many RL-based MAPF methods use independent training of agents – despite its non-stationarity issues and lacking guarantees regarding spatial coordination (Laurent et al. 2011). To accommodate these limitations, ad hoc measures like replanning and communication are used for reactive coordination, e.g., *CostTracer* and *SCRIMP* (Skrynnik et al. 2024; Wang et al. 2023). Further additions are imitation learning, reward shaping, and overfitting to conventions, e.g., *PRIMAL* (Damani et al. 2021; Sartoretti et al. 2019). Such ad hoc measures often complicate the approach and require knowledge beyond the accessible information in stochastic games, i.e., $o_{t,i}$, limiting generalizability, effectiveness, and sample efficiency (Phan, Phan, and Koenig 2025).

3.4 Reverse Curricula for MAPF

Curriculum learning aims to master complex tasks through stepwise solving of easier (sub-)tasks (Bengio et al. 2009; Florensa et al. 2017). *CACTUS* is a *reverse curriculum approach*, which randomly places goals $p_{goal,i}$ in a *bounding region* $\mathcal{G}_i$ around the corresponding start locations $p_{start,i}$ within an *allocation radius* $R_{alloc} \geq 1$ for training (Phan et al. 2024a) (Fig. 2). A *curriculum threshold* $U \in (0, 1)$ and *deviation factor* $\eta > 0$ are used to increment R_{alloc}, if the average completion rate μ and standard deviation σ satisfy $\mu - \eta\sigma > U$. This corresponds to a significance test if the expected completion rate is at least U. Unlike other RL-based MAPF methods, CACTUS uses CTDE methods like value factorization (Rashid et al. 2020) and achieves superior effectiveness with less than 5% of the data and compute of prior methods with ad hoc measures (Fig. 1b).

4 Spatially Grouped Curriculum Learning

We now introduce *Partitioned Attention-based Reverse Curricula for Enhanced Learning (PARCEL)* for grouped curriculum learning in MAPF (Fig. 3). PARCEL builds on the bounding regions of CACTUS (Fig. 2), but we note that *any* (future) spatial curriculum approach could be used as well. Algorithm 1 summarizes PARCEL, where U is the curriculum threshold and η is the deviation factor of CACTUS. W is the epoch count, and Y is the episode count per epoch.

4.1 Grouping via Episodic Preprocessing

Given a *curriculum stage* defined by the allocation radius $R_{alloc} \geq 1$, we define the bounding region of an agent $i \in \mathcal{D}$ as $\mathcal{G}_i = \{p \in \mathcal{V} | d(p, p_{start,i}) \leq R_{alloc}\}$, from which we sample a goal location $p_{goal,i}$ that is reachable within R_{alloc} steps, even in the presence of other agents (Phan et al. 2024a)[1].

We can safely assume that two agents $i, j \in \mathcal{D}$ can reach their respective goals within R_{alloc} steps independently, without explicit coordination, if their bounding regions do not overlap, i.e., $\mathcal{G}_i \cap \mathcal{G}_j = \emptyset$ (Wagner and Choset 2011). However, when $\mathcal{G}_i \cap \mathcal{G}_j \neq \emptyset$, we assume a potential risk of conflicts. Thus, if an agent i has an overlapping region with any agent j of *spatial group* C_x, i.e., $j \in C_x \subseteq \mathcal{D}$, then agent i also belongs to this spatial group, i.e., $i \in C_x$.

We denote $\mathcal{C} = \{C_1, ..., C_X\}$ as *group assignment* and $X \geq 1$ as *group count*. $\mathcal{C}$ is defined in a global *preprocessing step* at the beginning of each episode to determine which agents should be trained jointly for spatial coordination.

4.2 Masked Attention-Based Critics

To consider the spatial groups in $\mathcal{C}$ via MARL, we use masked self-attention in the critics $\hat{Q}_i$ for coordinated and partially independent learning, where all groups can be trained separately (Fig. 3). We define a *grouping mask* $\mathcal{M} \in \mathbb{R}^{N \times N}$, with each entry $\mathcal{M}_{i,j} = c \in \mathbb{R}$ iff $i, j \in \mathcal{D}$ belong to the same spatial group $C_x \in \mathcal{C}$, and $\mathcal{M}_{i,j} = -\infty$ otherwise.

Given three neural networks, i.e., *multilayer perceptrons (MLP)* q, k, v, we obtain the individual row vectors $W_i^q = q(\tau_{t,i}), W_i^k = k(\tau_{t,i}), W_i^v = v(\tau_{t,i}) \in \mathbb{R}^{1 \times z}$, respectively, where $z > 0$ is the *embedding dimension*. By arranging the respective rows as matrices, we obtain $W^q, W^k, W^v \in \mathbb{R}^{N \times z}$ to compute their *self-attention* (Vaswani et al. 2017):

$$att(W^q, W^k, W^v, \mathcal{M}) = softmax\left(\frac{W^q (W^k)^\intercal + \mathcal{M}}{\sqrt{z}}\right) W^v \quad (3)$$

[1]We can ensure this efficiently, e.g., by simulating a joint random walk for R_{alloc} steps while avoiding vertex and edge conflicts.

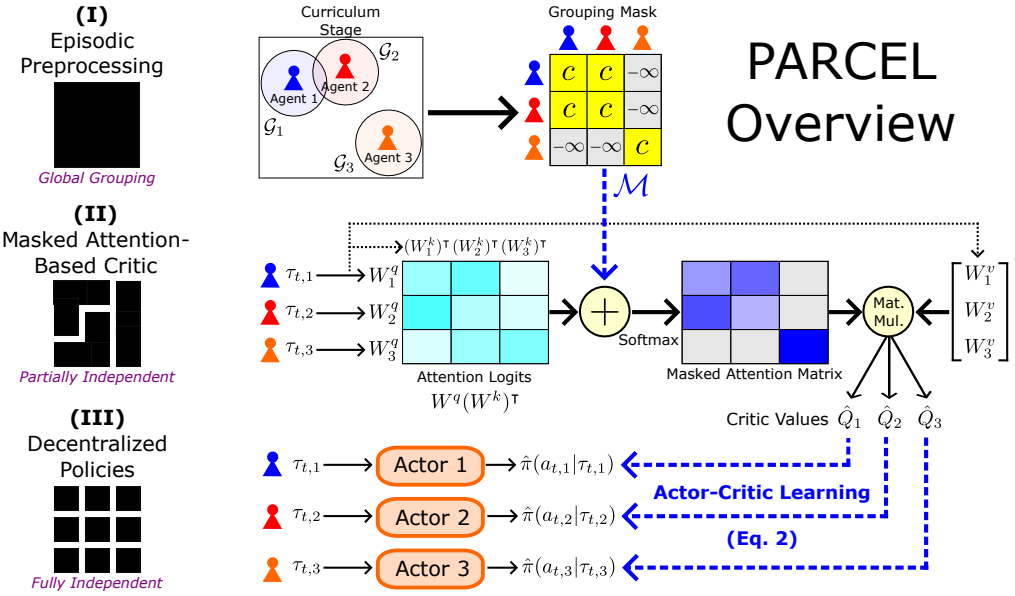


Figure 3: Overview of PARCEL. The blue dashed arrows indicate interactions between the three phases. **(I)** At the beginning of each episode, all agents are grouped according to the overlaps of their bounding regions $\mathcal{G}_i$ (shaded circles) w.r.t. the curriculum stage, defined by R_{alloc} (Fig. 2). A symmetric grouping mask $\mathcal{M}$ is defined, where a constant $c \in \mathbb{R}$ is assigned to each pair $i, j \in \mathcal{D}$ if they belong to the same spatial group $i, j \in C_x \subseteq \mathcal{D}$, and $-\infty$ otherwise. **(II)** Attention-based critic using the matrices $W^q, W^k, W^v \in \mathbb{R}^{N \times z}$ and the grouping mask $\mathcal{M}$ from (I) to calculate the critic value $\hat{Q}_i$ for each agent $i \in \mathcal{D}$, according to Eq. 3 or Eq. 4. **(III)** Decentralized policies or actors $\hat{\pi}_i$ trained with the critics $\hat{Q}_i$ from (II), according to Eq. 2.

where $softmax(\xi) = \frac{e^{\xi}}{\sum_{\xi_h \in \xi} e^{\xi_h}} \in [0,1]^{1 \times N}$ for each row vector $\xi \in \mathbb{R}^{1 \times N}$. The mask $\mathcal{M}$ ensures that agents of different groups are not considered in each other's rows as their mutual *softmax* would be zero due to $\lim_{\alpha \to -\infty} e^{\alpha} = 0$.

Eq. 3 can be calculated row-wise for each agent $i \in C_x$:

$$agent\_att_i(W_i^q, W^k, W_i^v, \mathcal{M}_i) = \sum_{j \in C_x} \mathbb{P}_{i,j} W_{i,j}^v \quad (4)$$

where $\mathbb{P}_{i,j}$ is the jth entry of $softmax((W_i^q (W^k)^\intercal + \mathcal{M}_i)/\sqrt{z})$ w.r.t. Eq. 3 and $\mathcal{M}_i$ is the ith row of the grouping matrix $\mathcal{M}$. Thus, we only need to train agents i, j jointly, when they belong to the same group C_x, i.e., $\mathcal{M}_{i,j} = c \in \mathbb{R}$. Since $softmax(\xi + \bar{c}) = softmax(\xi)$, where $\bar{c} \in \{c\}^{1 \times N}$, we can set the constant c to an arbitrary real number, e.g., zero.

In practice, multiple attention heads are used for Eq. 3, which are summed and processed to critic values $\hat{Q}_i$ afterwards (Iqbal et al. 2021; Vaswani et al. 2017). Given Y episodes y of T time steps each, the attention-based critic $\hat{Q}_i$ of agent $i \in \mathcal{D}$ is learned by minimizing the loss $\mathcal{L}_i^{\hat{Q}_i}$:

$$\mathcal{L}_i^{\hat{Q}_i} = \frac{1}{YT} \sum_{y=1}^{Y} \sum_{t=0}^{T-1} \left[\left(\hat{Q}_i(\tau_{t,i}^{(y)}, a_{t,i}^{(y)}) - \sum_{\alpha=0}^{T-1} r_{t+\alpha,i}^{(y)} \right)^2 \right] \quad (5)$$

Alternatively, the critics $\hat{Q}_i$ could be processed by a centralized factorization operator (Phan et al. 2023; Rashid et al. 2020), whose investigation we defer to future work.

4.3 Learning Decentralized Policies

Through $\hat{Q}_i$, our goal is to empower the local policies or actors $\hat{\pi}_i$ with the necessary spatial coordination for independent execution after training, without further ad hoc measures. The critics $\hat{Q}_i$ are only used during training to learn $\hat{\pi}_i$ using actor-critic methods (Section 2.2 and Eq. 2).

We employ the CACTUS curriculum (Fig. 2) for the bounding regions $\mathcal{G}_i$ to define spatial groups C_x (Section 4.1) and to improve sample efficiency (Phan et al. 2024a).

After training, each local policy $\hat{\pi}_i$ can be executed solely based on its individual history $\tau_{t,i}$, without additional ad hoc measures like replanning and communication, thus ensuring compliance with the MAPF stochastic game (Section 2.2).

4.4 Conceptual Discussion

The key idea of PARCEL is to consider bounding regions to partition MAPF tasks for tractable and partially independent learning. In contrast, most prior RL-based MAPF methods assume *unbounded regions*, i.e., where goals can be distributed arbitrarily across the map. Thus, most prior work cannot leverage such independence structures (Sartoretti et al. 2019; Skrynnik et al. 2024; Wang et al. 2023) and, therefore, use ad hoc measures like replanning and communication that complicate the approach and require knowledge beyond the observations $o_{t,i}$ in stochastic games, limiting generalizability, effectiveness, and sample efficiency.

The bounding regions enable *modular training*, where each spatial group can be trained separately and in parallel without non-stationarity issues, specific assumptions about

Algorithm 1: PARCEL for Partitioned Curriculum Learning

procedure *PARCEL*(U, η, W, Y)
  Initialize parameters of $\hat{\pi}_i, \hat{Q}_i$ for each agent $i \in \mathcal{D}$.
  $\hat{\pi} \leftarrow \langle \hat{\pi}_1, ..., \hat{\pi}_N \rangle$
  $R_{alloc} \leftarrow 1$ ▷ Initial curriculum stage, Section 3.4
  for epoch $w \leftarrow 1, W$ **do**
    for episode $y \leftarrow 1, Y$ **do**
      Generate map G with random start locations
      Define bounding regions $\mathcal{G}_i$ via G and R_{alloc}
      Generate instance I via bounding regions $\mathcal{G}_i$
      Form spatial groups C_x for all $i, j \in \mathcal{D}$ w.r.t. $\mathcal{G}_i, \mathcal{G}_j \subseteq \mathcal{V}$ ▷ **(I)** Episodic Preprocessing, Section 4.1
      Run episode with $\hat{\pi}$ for T time steps at most
    end for
    Define grouping mask $\mathcal{M}$ via $\mathcal{C} = \{C_1, ..., C_X\}$
    Train critic $\hat{Q}_i$ for each agent i with $\mathcal{M}$ (Eq. 4) ▷ **(II)** Masked Attention-Based Critic, Section 4.2
    Train actor $\hat{\pi}_i$ for each agent i with $\hat{Q}_i$ (Eq. 2) ▷ **(III)** Decentralized Policies, Section 4.3 and Fig. 3
    $R_{alloc} \leftarrow$ *CheckCurriculumStage*$(\hat{\pi}, U, \eta)$
  end for
  return $\langle \hat{\pi}_1, ..., \hat{\pi}_N \rangle$
end procedure

the map, extensive human knowledge, or expert data.

PARCEL draws inspiration from interaction-focused search algorithms like M* (Wagner and Choset 2011) and CBS (Sharon et al. 2012). Unlike most RL-based MAPF methods, which merely invoke search algorithms in an ad hoc manner, PARCEL is *purely learning-based*, providing an analogous RL concept without running the actual search.

Using the CACTUS regime (Fig. 2), the bounding regions grow over the course of training and eventually merge into the whole training map, resulting in a single spatial group $C_x \rightarrow \mathcal{D}$. At this stage, all agent policies are sufficiently proficient at spatial coordination, as ensured by the confidence-based assessment of CACTUS (Phan et al. 2024a). Intriguing directions for future work are to tighten these bounding regions further for better modularity and to consider asymmetric matrices $\mathcal{M}$ w.r.t. unilateral agent dependencies.

Alternatively, we could also equip the local policies $\hat{\pi}_i$ with self-attention, but this would require ad hoc communication during execution, similar to SCRIMP (Wang et al. 2023), causing additional overhead and limited scalability.

5 Experiments

MARL Algorithms We implemented PARCEL based on the public code of (Phan et al. 2024a). We re-implemented CACTUS (spatial curriculum), PRIMAL (reward shaping), SCRIMP (attention-based communication), and CostTracer (observation shaping) from (Damani et al. 2021; Phan et al. 2024a; Sartoretti et al. 2019; Skrynnik et al. 2024), excluding heuristic search additions which leverage human knowledge beyond the observations $o_{t,i}$ in stochastic games. This allows us to assess the general learning (in-)capability of each approach regarding spatial coordination in a fair way.

Architecture and Hyperparameters For all RL-based MAPF methods, we use deep neural networks to implement $\hat{\pi}_i$ and $\hat{Q}_i$ for each agent i. The neural networks are updated after every $Y = 10$ episodes using ADAM optimization with a learning rate of 0.001. We always train $\hat{\pi}_i$ with PPO.

Since most evaluation domains are gridworlds, the observations are encoded as 7×7 sub-grids, as suggested in (Sartoretti et al. 2019). We implement all neural networks as MLPs and flatten the observations before feeding them into the MLPs. $\hat{\pi}_i$ has two hidden layers of 64 units with ELU activation. The output of $\hat{\pi}_i$ has $|\mathcal{A}_i|$ units with softmax activation. The critic $\hat{Q}_i$ of PARCEL has two masked attention heads with an intermediate dimension of $z = 64$. The attention outputs of Eq. 3 are summed and processed by a linear layer with $|\mathcal{A}_i|$ output units. The critics of all other baselines (except PARCEL ablations) have a similar architecture as the actor $\hat{\pi}_i$. CACTUS uses QMIX from (Rashid et al. 2020) for value factorization via hypernetworks with two hidden layers of 128 units with ELU activation and one linear output unit. The complete architectures have fewer than 600,000 parameters, which is *less than 5%* of common prior models (Damani et al. 2021; Sartoretti et al. 2019; Wang et al. 2023), and should prevent overfitting via overparameterization (LeCun, Bengio, and Hinton 2015).

PARCEL and CACTUS use a curriculum threshold of $U = 50\%$ and $\eta = 2$, which corresponds to a confidence level of about 97% in one-tailed tests (Phan et al. 2024a). Both use the shortest path distance to measure R_{alloc}.

Training Setup Despite the availability of pretrained models, we run each learning algorithm from scratch because **(1)** prior models have been trained under different circumstances (maps, number of agents, model sizes, etc.), which makes a fair comparison difficult, and **(2)** to assess their sample efficiency, i.e., the learning progress over the total number of episodes, which has been widely ignored. For each experiment, all algorithms are run 20 times to report the average performance, e.g., the completion rate of each agent reaching its goal, as well as the 95% confidence interval.

5.1 Experiment – Simple Coordination

Setting We first study two simple 2-agent tasks, namely `Stag Hunt` as a matrix coordination game and a small MAPF task in a 3×3 grid world, as illustrated in Fig. 4. Focusing on such simple tasks, allows us to assess the potential of each algorithm to actually learn coordinated behavior and to transfer to non-MAPF tasks, such as `Stag Hunt`, where both agents are rewarded with $+2$ if they jointly hunt the stag (S), $+1$ if they both hunt hares (H). Hunting different animals results in $+\frac{1}{2}$ for the hare-hunting agent, and 0 for the stag-hunting agent (Fig. 4, top-left payoff matrix). Since `Stag Hunt` has no spatial features, we only compare the relevant architectures and consider the previous joint action a_{t-1} as the current observation $o_{t,i}$. We train each algorithm for $W = 100$ epochs à $Y = 10$ episodes with $T = 50$.

Results The results are given in Fig. 4. In `Stag Hunt` (top row), our attention-based critic for PARCEL achieves

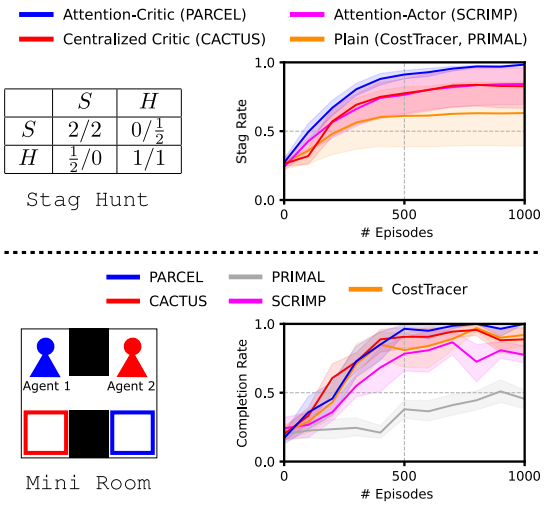
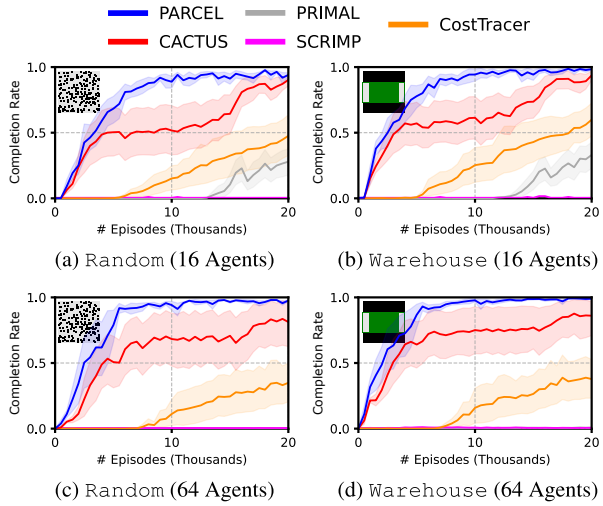


Figure 4: **Top**: Progress of different learning architectures in the one-shot `Stag Hunt` game. **Bottom**: Progress of PARCEL and other RL-based methods in a small MAPF task.



(a) `Random` (16 Agents) (b) `Warehouse` (16 Agents)

(c) `Random` (64 Agents) (d) `Warehouse` (64 Agents)

Figure 5: Training progress of PARCEL and state-of-the-art baselines with $N = \{16, 64\}$ agents over 20 runs. Shaded areas show the 95% confidence interval.

the highest stag rate over the attention-based actor used in SCRIMP and the centralized critic used in CACTUS. Plain actor-critic/PPO learning (Eq. 2) without any coordination technique, e.g., PRIMAL or CostTracer, performs worst.

In the MAPF tasks (bottom row), most algorithms manage to learn coordinated goal-reaching policies, except for PRIMAL, which only completes 50% agents on average.

Discussion These simple tasks demonstrate the basic capability of each algorithm to learn coordinated behavior in non-spatial (e.g., `Stag Hunt`) and spatial (e.g., MAPF) tasks, without considering scalability yet. PARCEL, SCRIMP, and CACTUS are most promising in both scenarios, indicating general algorithmic advantages compared to CostTracer and PRIMAL, which depend on specific spatial aids, e.g., observation shaping and expert data, respectively.

Although each baseline could be specifically engineered toward outperforming PARCEL, e.g., via larger neural networks, expert data, or ad hoc replanning, we regard such specializations as unreasonable for these simple tasks.

5.2 Experiment – Sample Efficiency (Training)

Setting Next, we evaluate how these algorithms scale to larger maps with $N \in \{16, 64\}$ agents. We consider two maps from the MAPF benchmark set of (Stern et al. 2019), namely a `Random` map (*Random-64-64-10*) and a `Warehouse` map. For training, we crop random 64×64 sub-grids of `Random` and `Warehouse`, which are rotated and mirrored randomly. We train each algorithm for $W = 2000$ epochs à $Y = 10$ episodes with horizon $T = 256$.

Results The training progress is shown in Fig. 5. In all settings, PARCEL is the most sample-efficient approach, progressing significantly faster than any other algorithm during training. CACTUS progresses second fastest with notably high variance, eventually matching the completion rate of PARCEL in the 16-agent setting, but failing to do so in the 64-agent setting. CostTracer progresses third fastest, performing slightly better in the 16-agent setting. PRIMAL and SCRIMP perform worst, with PRIMAL making at least some progress in the 16-agent setting.

Discussion The results demonstrate that PARCEL is sample-efficient and also scales well w.r.t. the number of agents N and map size. Despite most algorithms having the potential to learn spatial coordination, they are not sample-efficient enough to scale to larger scenarios within 20,000 episodes (and without any expert data). SCRIMP struggles with the quadratic scale in communication effort, despite faring well in smaller tasks (Section 5.1). While training each algorithm with more episodes could eventually lead to some progress, the results highlight a severe limitation of most prior work on RL-based MAPF, and thus the need for better sample and compute efficiency to ensure sustainability.

5.3 Experiment – Generalization (Test)

Setting We now test the policies trained with $N = 64$ in Section 5.2. We consider two maps from the MAPF benchmark set of (Stern et al. 2019) with their original sizes and orientations, namely a `Game` map (*Den520d*) and a `City` map (*Paris_1_256*), and use trimmed versions of the 25 random instances for evaluation, where we run the LaCAM* algorithm (Okumura 2023) to place the goals closer to the start locations, such that they are reachable within the horizon used in Section 5.2. We assess the completion rate and average travel time for different numbers of agents, i.e., $N > 64$, where unsuccessful paths ϕ_i are considered via $sup\,(l(\phi_i)) = T$. We also compare with the search-based and non-learning MAPF solver LaCAM* (Okumura 2023).

Results The test results are shown in Fig. 6. In all settings, PARCEL achieves a higher completion rate and lower average travel time than all other baselines except LaCAM*. CACTUS is the second-best learning approach with notably

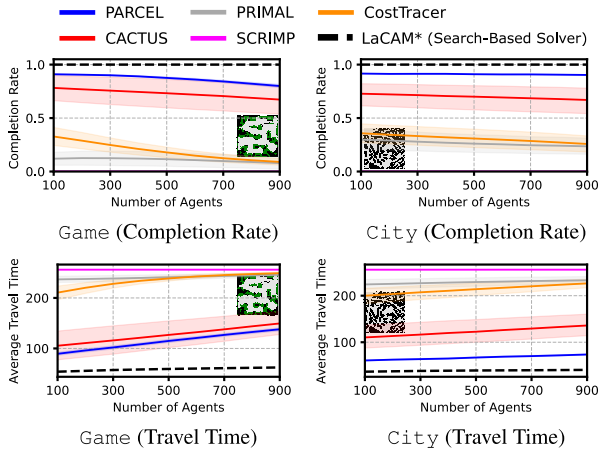
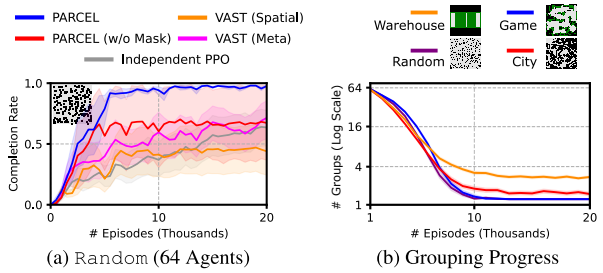


Figure 6: Test completion rate (top row) and average travel time (bottom row) of PARCEL and state-of-the-art baselines for different numbers of agents N in maps not seen during training. Shaded areas show the 95% confidence interval.



(a) Random (64 Agents) (b) Grouping Progress

Figure 7: **Left**: Training progress of PARCEL and ablations using CACTUS with $N = 64$ agents over 20 runs. Shaded areas show the 95% confidence interval. **Right**: Evolution of the group count X for different maps over the course of training. Note the logarithmic scale of the y-axis.

higher variance, reflecting its training progress from Section 5.2. PRIMAL, SCRIMP, and CostTracer perform poorly in the test maps, never achieving a completion rate over 50% or an average travel time clearly below 200 time steps.

Discussion The results show that the PARCEL policies are not only sample-efficient to train, but can also generalize better to maps with structures and sizes that are different from the training maps, as well as different numbers of agents, than the RL-based alternatives. This supports the effectiveness and generalizability of our partially independent training approach, using spatial grouping and masked attention, over additional ad hoc mechanisms. Compared with LaCAM*, PARCEL tends to make unnecessary detours due to the probabilistic policies, as defined in Section 2.2.

5.4 Experiment – Ablations and Group Count

Setting Finally, we compare PARCEL with different ablations by omitting the grouping mask $\mathcal{M}$, replacing the attention mechanism with linear grouping, i.e., VAST (Phan et al. 2021), or entirely relying on independent PPO training without additional ad hoc measures, unlike prior approaches. We compare with the original formulation of VAST, which attempts to learn subteams via a centralized meta-policy (Section 3.1), as well as a VAST variant with spatial groups similar to PARCEL. All variants use the CACTUS curriculum. Experiments with REFIL (Iqbal et al. 2021), i.e., using random groups, led to similar results as just omitting $\mathcal{M}$. Thus, we only report the PARCEL ablation without the mask $\mathcal{M}$.

To assess the evolution of the group count X over the course of training, we trained PARCEL on each test map separately and recorded X for all training episodes.

Results The results are shown in Fig. 7 (left). PARCEL is the most sample-efficient variant as it progresses fastest. Omitting the grouping mask $\mathcal{M}$, the attention mechanism, or replacing the attention mechanism with linear grouping, i.e., VAST, results in less efficient and unstable learning.

Fig. 7 (right) shows the evolution of the group count X over the course of training, which differs according to the map structure. In Random and Game, PARCEL converges to a single coordinated group after 10,000 episodes. In City, PARCEL eventually forms one or two groups, and in Warehouse, PARCEL eventually forms 3 or 4 groups.

Discussion The PARCEL ablation without the grouping mask $\mathcal{M}$ highlights the importance of spatial grouping (Section 4.1) in our attention-based critics. Both VAST variants confirm the importance of self-attention (Section 4.2), as a more expressive way of considering spatial groups. The results of independent PPO indicate that PARCEL effectively empowers the decentralized policies (Section 4.3) with spatial coordination capabilities that naive independent training (without any ad hoc measures) lacks.

The evolution of X for different maps shows that PARCEL can flexibly adapt to different map structures without specific ad hoc measures. In the Warehouse map, PARCEL still has to balance between group-wise separation and joint training, as the reverse curriculum requires more epochs to fully expand the bounding regions, i.e., $\mathcal{G}_i \to \mathcal{V}$.

6 Conclusion

In this paper, we presented PARCEL as a spatial grouping-based alternative to common independent training approaches to MAPF. PARCEL considers bounding regions for all agents and only trains them jointly via self-attention if their regions overlap to avoid potential conflicts. By employing a reverse curriculum, such as CACTUS, where the bounding regions grow as the policies improve, all agents will eventually merge into a single coordinated group.

We evaluated PARCEL in various benchmark tasks and demonstrated superior effectiveness and sample efficiency to prior RL-based MAPF methods, which mostly rely on ad hoc measures to cope with spatial coordination. PARCEL demonstrated faster training progress as well as better scalability and generalization to large and structured test maps than the RL-based baselines, as well as its own ablations.

Future work includes investigating tighter bounding regions, asymmetric grouping matrices modelling unilateral agent dependencies, and the transfer to other NP-hard problems, such as TSP, MILP, or SAT, via reduction.

Acknowledgments

The research at the University of California, Irvine was supported by the National Science Foundation (NSF) under grant numbers 2544613, 2434916, 2321786, 2112533, as well as gifts from Amazon Robotics and the Donald Bren Foundation. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, or the U.S. government. Sven Koenig was awarded a WASP Distinguished Guest Professorship at Örebro University.

References

Alkazzi, J.-M.; and Okumura, K. 2024. A Comprehensive Review on Leveraging Machine Learning for Multi-Agent Path Finding. *IEEE Access*.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum Learning. In *26th International Conference on Machine Learning*.

Cohen, L.; and Koenig, S. 2016. Bounded Suboptimal Multi-Agent Path Finding using Highways. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 3978–3979.

Damani, M.; Luo, Z.; Wenzel, E.; and Sartoretti, G. 2021. PRIMAL2: Pathfinding via Reinforcement and Imitation Multi-Agent Learning-Lifelong. *IEEE Robotics and Automation Letters*, 6(2): 2666–2673.

Emery-Montemerlo, R.; Gordon, G.; Schneider, J.; and Thrun, S. 2004. Approximate Solutions for Partially Observable Stochastic Games with Common Payoffs. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.*, 136–143. IEEE.

Florensa, C.; Held, D.; Wulfmeier, M.; Zhang, M.; et al. 2017. Reverse Curriculum Generation for Reinforcement Learning. In *Conference on Robot Learning*, 482–495. PMLR.

Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual Multi-Agent Policy Gradients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Huang, T.; Dilkina, B.; and Koenig, S. 2021. Learning Node-Selection Strategies in Bounded Suboptimal Conflict-Based Search for Multi-Agent Path Finding. In *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Huang, T.; Li, J.; Koenig, S.; and Dilkina, B. 2022. Anytime Multi-Agent Path Finding via Machine Learning-Guided Large Neighborhood Search. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, 9368–9376.

Iqbal, S.; De Witt, C. A. S.; Peng, B.; Boehmer, W.; Whiteson, S.; and Sha, F. 2021. Randomized Entity-wise Factorization for Multi-Agent Reinforcement Learning. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 4596–4606. PMLR.

Iqbal, S.; and Sha, F. 2019. Actor-Attention-Critic for Multi-Agent Reinforcement Learning. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 2961–2970. PMLR.

Kaduri, O.; Boyarski, E.; and Stern, R. 2020. Algorithm Selection for Optimal Multi-Agent Pathfinding. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, 161–165.

Laurent, G. J.; Matignon, L.; Fort-Piat, L.; et al. 2011. The World of Independent Learners is not Markovian. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 15(1): 55–64.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep Learning. *Nature*, 521(7553): 436–444.

Li, J.; Chen, Z.; Harabor, D.; Stuckey, P. J.; and Koenig, S. 2021a. Anytime Multi-Agent Path Finding via Large Neighborhood Search. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 4127–4135.

Li, J.; Tinka, A.; Kiesel, S.; Durham, J. W.; Kumar, T. K. S.; and Koenig, S. 2021b. Lifelong Multi-Agent Path Finding in Large-Scale Warehouses. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13): 11272–11281.

Li, M.; Wang, Q.; and Xu, Y. 2025. GTDE: Grouped Training with Decentralized Execution for Multi-Agent Actor-Critic. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 18368–18376.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-Level Control through Deep Reinforcement Learning. *Nature*, 518(7540): 529–533.

Okumura, K. 2023. Improving LaCAM for Scalable Eventually Optimal Multi-Agent Pathfinding. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Oliehoek, F. A.; and Amato, C. 2016. *A Concise Introduction to Decentralized POMDPs*. Springer.

Phan, T.; Driscoll, J.; Romberg, J.; and Koenig, S. 2024a. Confidence-Based Curriculum Learning for Multi-Agent Path Finding. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 1558–1566.

Phan, T.; Driscoll, J.; Romberg, J.; and Koenig, S. 2025a. Confidence-Based Curricula for Multi-Agent Path Finding via Reinforcement Learning. *Preprint at Research Square*.

Phan, T.; Huang, T.; Dilkina, B.; and Koenig, S. 2024b. Adaptive Anytime Multi-Agent Path Finding Using Bandit-Based Large Neighborhood Search. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 38(16): 17514–17522.

Phan, T.; Phan, T.; and Koenig, S. 2025. Generative Curricula for Multi-Agent Path Finding via Unsupervised and

Reinforcement Learning. *Journal of Artificial Intelligence Research*, 82: 2471–2534.

Phan, T.; Ritz, F.; Altmann, P.; Zorn, M.; Nüßlein, J.; Kölle, M.; Gabor, T.; and Linnhoff-Popien, C. 2023. Attention-Based Recurrence for Multi-Agent Reinforcement Learning under Stochastic Partial Observability. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 27840–27853. PMLR.

Phan, T.; Ritz, F.; Belzner, L.; Altmann, P.; Gabor, T.; and Linnhoff-Popien, C. 2021. VAST: Value Function Factorization with Variable Agent Sub-Teams. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 24018–24032. Curran Associates, Inc.

Phan, T.; Zhang, B.; Chan, S.-H.; and Koenig, S. 2025b. Anytime Multi-Agent Path Finding with an Adaptive Delay-Based Heuristic. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, 23286–23294.

Rashid, T.; Samvelyan, M.; de Witt, C. S.; Farquhar, G.; Foerster, J.; et al. 2020. Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *Journal of Machine Learning Research*, 21(178): 1–51.

Ratner, D.; and Warmuth, M. 1986. Finding a Shortest Solution for the NxN Extension of the 15-Puzzle is Intractable. In *Proceedings of the Fifth AAAI National Conference on Artificial Intelligence*, AAAI'86, 168–172. AAAI Press.

Sartoretti, G.; Kerr, J.; Shi, Y.; Wagner, G.; Kumar, T. S.; Koenig, S.; and Choset, H. 2019. PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning. *IEEE Robotics and Automation Letters*, 4(3): 2378–2385.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. In *arXiv preprint arXiv:1707.06347*.

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. 2012. Conflict-Based Search For Optimal Multi-Agent Path Finding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1): 563–569.

Skrynnik, A.; Andreychuk, A.; Yakovlev, K.; and Panov, A. 2024. Decentralized Monte Carlo Tree Search for Partially Observable Multi-Agent Pathfinding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17531–17540.

Stern, R.; Sturtevant, N.; Felner, A.; Koenig, S.; Ma, H.; Walker, T.; Li, J.; Atzmon, D.; Cohen, L.; Kumar, T.; et al. 2019. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. In *Proceedings of the International Symposium on Combinatorial Search*, volume 10, 151–158.

Sutton, R. 2019. The Bitter Lesson. *Incomplete Ideas (Blog)*, 13(1): 38.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In Solla, S.; Leen, T.; and Müller, K., eds., *Advances in Neural Information Processing Systems*, volume 12, 1057–1063. MIT Press.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All You Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wagner, G.; and Choset, H. 2011. M*: A Complete Multirobot Path Planning Algorithm with Performance Bounds. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3260–3267.

Wang, Y.; Xiang, B.; Huang, S.; and Sartoretti, G. 2023. SCRIMP: Scalable Communication for Reinforcement-and Imitation-Learning-Based Multi-Agent Pathfinding. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9301–9308. IEEE.

Watkins, C. J.; and Dayan, P. 1992. Q-Learning. *Machine Learning*, 8(3-4): 279–292.

Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; and Wu, Y. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. *Advances in Neural Information Processing Systems*, 35: 24611–24624.

Zang, Y.; He, J.; Li, K.; Fu, H.; Fu, Q.; Xing, J.; and Cheng, J. 2023. Automatic Grouping for Efficient Cooperative Multi-Agent Reinforcement Learning. *Advances in neural information processing systems*, 36: 46105–46121.

Zhang, Y.; Fontaine, M. C.; Bhatt, V.; Nikolaidis, S.; and Li, J. 2023. Arbitrarily Scalable Environment Generators via Neural Cellular Automata. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, 57212–57225.