Extended Abstract: A Semi-Decentralized Approach to Scalable Multiagent Control

Mahdi Al-Husseini, Avi Singh

[1]Stanford University, CS224R

Scalable multiagent control under stochastic communication remains an open challenge. Several real-world multiagent problems, including robot exploration teams (Saboia et al. 2022; Rouček et al. 2019), cooperative driving (Zhao et al. 2021; Li et al. 2023), and satellite collision avoidance (Dolan, Nayak, and Balakrishnan 2023; Zhao et al. 2025), feature stochastic communication dynamics conditioned on the underlying environment or actions taken. These *semi-decentralized* systems (Al-Husseini, Wray, and Kochenderfer 2026) (Figure 1) necessitate planning over the stochastic information structure under which actions are executed and observations are received. Planning approaches (RS-SDA*) scale poorly, sampling-based search methods (Dec-MCTS) assume fixed-period communication, and MARL methods (QMIX, MAPPO, MAZero) enforce strict execution-time decentralization, foreclosing coordination opportunities that arise stochastically at decision time.

We adopt the Semi-Decentralized POMDP, which unifies the Dec-POMDP and MPOMDP through a distribution over agent communication-sojourn times. We introduce SDecMCTS, a multiagent MCTS algorithm competitive with RS-SDA* in solution quality at comparable compute on small benchmarks. SDecMCTS uses a centralized belief-MDP tree as a high-quality critic, then solves a factored local-policy extraction problem over reachable decentralized stages. We extend SDecMCTS to Semi-Decentralized Zero (SDecZero), a scalable AlphaZero-style algorithm that trains centrally and executes semi-decentrally (Figure 2). SDecZero searches over the belief-MDP using a unified value and policy network. We add a communication head to the network to learn the communication-sojourn time distribution.

SDecMCTS is competitive with Approximate RS-SDA* and significantly outperforms DecMCTS across three canonical semi-decentralized benchmarks and deterministic and stochastic versions of six generated labyrinth environments. Similarly, SDecZero significantly outperforms QMIX, MAPPO, IPPO, and MASAC across two scenarios in large multi-drone search problems in real-world cave systems associated with the DARPA Subterranean Challenge. Overall, the results show a consistent progression across planning, learning with known communication, and learning with inferred communication. SDecMCTS establishes that semi-decentralized search can approach centralized planning quality on small and medium-scale benchmarks while avoiding the brittleness of fixed-period Dec-MCTS. SDecZero then amortizes this search procedure and scales it to realistic cave-system graphs. Finally, our learned-communication-head experiment demonstrates that the stochastic communication model itself can be recovered from data with sufficient accuracy to support high-return semi-decentralized execution.

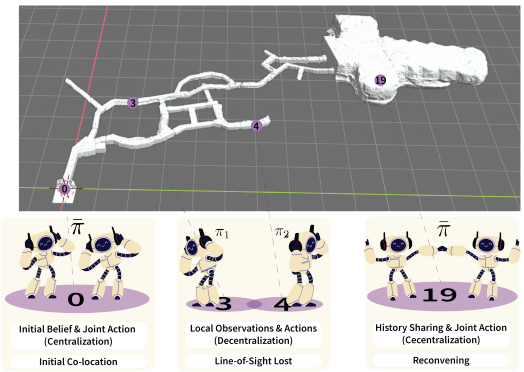


Figure 1: Illustrating semi-decentralization in the DARPA Subterranean Challenge environment.

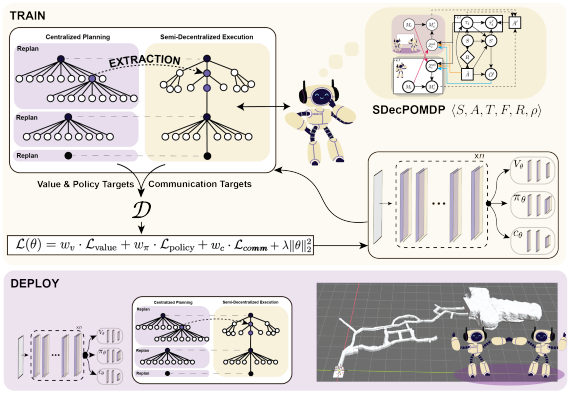


Figure 2: SDecZero algorithm outline.

A Semi-Decentralized Approach to Scalable Multiagent Control

Mahdi Al-Husseini, Avi Singh

[1]Stanford University, CS224R

Abstract

Scalable multiagent control under stochastic communication remains an open challenge. Planning approaches (RS-SDA*) scale poorly, sampling-based search methods (Dec-MCTS) assume fixed-period communication, and MARL methods (QMIX, MAPPO, MAZero) enforce strict execution-time decentralization, foreclosing coordination opportunities that arise stochastically at decision time. We adopt the Semi-Decentralized POMDP, which unifies the Dec-POMDP and MPOMDP through a distribution over agent communication-sojourn times. We introduce SDecMCTS, a multiagent MCTS algorithm competitive with RS-SDA* in solution quality at comparable compute on small benchmarks. SDecMCTS uses a centralized belief-MDP tree as a high-quality critic, then solves a factored local-policy extraction problem over reachable decentralized stages. We extend SDecMCTS to Semi-Decentralized Zero (SDecZero), a scalable AlphaZero-style algorithm that trains centrally and executes semi-decentrally. SDecZero searches over the belief-MDP using a unified value and policy network. We add a communication head to the network to learn the communication-sojourn time distribution. SDecZero significantly outperforms QMIX, MAPPO, IPPO, and MASAC across two scenarios in large multi-drone search problems in real-world cave systems associated with the DARPA Subterranean Challenge.

Introduction

Several real-world multiagent problems, including robot exploration teams (Saboia et al. 2022; Rouček et al. 2019), cooperative driving (Zhao et al. 2021; Li et al. 2023), and satellite collision avoidance (Dolan, Nayak, and Balakrishnan 2023; Zhao et al. 2025), feature stochastic communication dynamics conditioned on the underlying environment or actions taken. These *semi-decentralized* systems (Al-Husseini, Wray, and Kochenderfer 2026) necessitate planning over the stochastic information structure under which actions are executed and observations are received. Consider a robot team conducting a search and rescue mission in a subterranean cave with line-of-sight communication, depicted in Figure 1. The robots may temporarily share observations and coordinate a joint plan, lose contact while moving through different tunnels, and later reconnect with new local histories. Most MARL methods, including centralized training with decentralized execution (CTDE) (Amato 2024) and learned-communication approaches (Zhu, Dastani, and Wang 2024a), assume a fixed execution-time information structure. Agents either execute decentralized policies, possibly conditioned on learned messages, or operate under an always-centralized planner. Few methods perform online joint-action planning at intermittent communication opportunities; exceptions include the very classical Dec-POMDP planners (Al-Husseini et al. 2026) that inspire our work. Recent multiagent model-based tree-search algorithms (Liu et al. 2024a; Hao et al. 2024a) demonstrate improved sample efficiency, but again assume online decentralization. There is a well-defined gap between scalable and efficient MARL algorithms and domains in which communication opportunities are stochastic, recurrent, and action- or state-dependent. We address this by explicitly modeling the execution-time information structure as part of the control problem, enabling agents to plan over both actions and the future communication regimes those actions may induce.

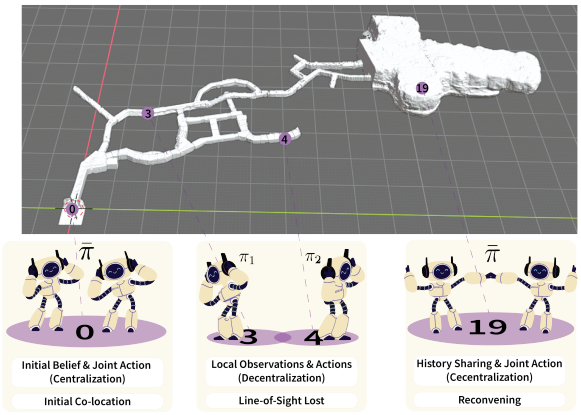


Figure 1: Illustrating semi-decentralization in the DARPA Subterranean Challenge environment.

We formalize the semi-decentralized setting using the *Semi-Decentralized POMDP* (Al-Husseini, Wray, and Kochenderfer 2026), a multiagent model in which communication and coordination opportunities evolve stochastically in the environment. We first introduce *SDecMCTS*, a classical

Monte Carlo tree search implementation that uses *Centralized Planning Semi-Decentralized Execution* (CPSDE). We then extend this planner to *SDecZero*, an AlphaZero-style method that amortizes semi-decentralized search through learned policy and value networks. Finally, we introduce *SDecMuZero*, which learns the stochastic communication-sojourn model during online search. Unlike prior CTDE and model-based MARL methods, our algorithms do not commit to a purely centralized or decentralized executor. Instead, they derive both joint coordination and decentralized fallback behavior from a single semi-decentralized search procedure. Across canonical semi-decentralized benchmarks and large-scale subterranean search-and-rescue simulations, our results show that search-based semi-decentralized policy learning can recover much of the value of centralized coordination under stochastic communication loss while remaining tractable.

Contributions

- We introduce SDecMCTS, a novel multiagent extension of MCTS which uses centralized planning and semi-decentralized execution (CPSDE). SDecMCTS demonstrates state-of-the-art performance across nine canonical semi-decentralized planning benchmarks.
- We further introduce SDecZero, which extends SDecMCTS to the AlphaZero framework, and enables significant scaling.
- We learn the underlying stochastic communication dynamics by adding a network communication head.
- We show that SDecZero significantly outperforms model-free (MAPPO, IPPO, QMIX, MASAC) methods in eight real-world semi-decentralized cave system simulations, to include that of the DARPA Subterranean Challenge site.

Related Work

Planning approaches. Multiagent planning is commonly formalized using the Dec-POMDP (Oliehoek, Amato et al. 2016), which provides a principled model for cooperative decentralized agents acting under partial observability. Dec-POMDP solvers such as MAA* (Szer, Charpillet, and Zilberstein 2005) and modern recursive small-step variants (Koops et al. 2023; Koops, Junges, and Jansen 2024a) optimize policy trees with admissible heuristic search and partial policy pruning, but remain limited by the combinatorial growth of joint policies and observation histories. More directly related to our setting, the SDec-POMDP formalism and RS-SDA* extend this line of work to semi-decentralized systems in which communication evolves stochastically per the environment or control strategy (Al-Husseini, Wray, and Kochenderfer 2026; Al-Husseini et al. 2026). Hybrid decentralized search techniques like Dec-MCTS (Best et al. 2019) and Dec-MCTS-SP (Li et al. 2019) improve scalability by maintaining agent-specific search-trees aided by prediction-based heuristics; they are however limited to the narrow case of fixed-period communication, and do not implement networks for scale.

Model-free multiagent reinforcement learning. Model-free MARL has produced scalable algorithms for cooperative and mixed multiagent domains, primarily through CTDE. Actor-critic methods such as MADDPG (Lowe et al. 2017) and COMA (Foerster et al. 2018) use centralized critics to stabilize learning and address multiagent credit assignment while deploying local policies at execution time. Value-factorization methods such as VDN (Su, Adams, and Beling 2021), QMIX (Rashid et al. 2020), and QPLEX (Wang et al. 2021) learn structured decompositions of the joint action-value function that permit decentralized action selection. Policy-gradient methods such as MAPPO (Yu et al. 2022) have proven performant across standard cooperative MARL benchmarks. Although scalable and generally effective, model-free approaches perform poorly in multiagent domains with stochastic information flow, sparse reward signals, and critical joint actions. Model-free approaches are sample inefficient, which degrades performance when interaction with the environment is limited.

Model-based multiagent reinforcement learning. Model-based MARL improves sample efficiency by exploiting known or learned dynamics models to enable imagined rollouts. MAMBA learns a multiagent communication world model to reduce environment interaction in cooperative tasks (Egorov and Shpilman 2022). MuZero-style methods that incorporate tree search into multiagent learning have become increasingly popular: MAZero uses a centralized model with MCTS to enhance policy search and introduces two search efficiency techniques (Liu et al. 2024b), MA Gumbel MuZero replaces the traditional PUCT exploration mechanism with Gumbel-top-k search to overcome combinatorial joint-action spaces (Hao et al. 2024b), and MALinZero cleverly projects joint-action returns into a low-dimensional space and solving a contextual linear bandit problem (Tang, Chen, and Lan 2026). Notably, these algorithms assume a fixed execution-time information structure. In contrast, our methods explicitly model stochastic transitions between communication regimes and derive centralized and decentralized behavior from a single semi-decentralized search procedure.

Communication and hybrid execution in multiagent reinforcement learning. The MARL-COMM literature (Zhu, Dastani, and Wang 2024b) augments decentralized policies with learned *explicit* communication mechanisms, including continuous differentiable messages, discrete protocols, gating, targeted communication, and attention-based aggregation. Representative methods such as DIAL (Foerster et al. 2016) learn communication channels jointly with policies, while later approaches such as IC3Net (Singh, Jain, and Sukhbaatar 2019), TarMAC (Das et al. 2019), and ATOC (Jiang and Lu 2018) learn when to communicate, whom to communicate with, or how to aggregate messages. Closest to our setting, MARO (Santos et al. 2025) studies *hybrid execution*, in which agents may encounter arbitrary communication levels at test time, ranging from fully decentralized to fully centralized execution, and proposes an autoregressive model for imputing missing observations under faulty or partial communication. These approaches improve coordination under partial observability and communication constraints, but they primarily learn message-conditioned policies or missing-observation predictors under a prescribed communication process.

Preliminaries

Definition 1. The *semi-Markov property for communication*, or *semi-decentralization*, admits a distribution over time for what information agents can store in memory.

Sojourn communication time τ is general continuous random variable representing the time for an agent to return to an information-sharing state. We here overload τ for sojourn communication time, distinct from sojourn control time.

eq:secondQ($\leq \bar{\tau}', s' \mid \bar{\tau}, s, \bar{a}, \bar{a}'$)

As with SMDPs, we can define Q as the product of $F(\bar{\tau}' \mid s', \bar{a}', \bar{\tau})$ and $T(s' \mid s, \bar{a}, \bar{\tau})$, where $\bar{\tau}'$ may be conditioned on the subsequent joint action set $\bar{a}'$. SMDPs have one agent with an implicit conditioned $\bar{\tau} = 0$. However, SDec-POMDPs have multiple agents with varied $\bar{\tau}$. Thus it is more general with $\bar{\tau}'$ conditioned on $\bar{\tau}$. Semi-decentralized models assume an initial $\bar{\tau}^0$, which can be interpreted as the communicating state of each agent when $\eta = 0$. When $\tau = 0$, information sharing can occur coinciding with a *communication epoch*. We assume noise-free instantaneous broadcast communication resulting in a single communicating agent set as in a blackboard (Erman et al. 1980; Craig 1988). Semi-decentralization may however incorporate multiple distinct communicating sets. Whereas as semi-Markov control systems toggle model transition dynamics using τ, semi-decentralized systems toggle updating histories using $\bar{\tau}$.

We pose the problem as a semi-decentralized partially observable Markov decision process (SDec-POMDP) (Al-Husseini, Wray, and Kochenderfer 2026). The SDec-POMDP is a principled model for cooperative multiagent teams subject to stochastic communication constraints. The SDec-POMDP unifies the Dec-POMDP and MPOMDP frameworks along a communication dimension.

SDec-POMDPs

Shown in Figure 2, the *semi-decentralized partially observable Markov decision process* (SDec-POMDP) is a semi-decentralized multiagent decision process for sequential decision-making under partial observability and probabilistic communication characterized by tuple $\langle I, S, \bar{A}, \bar{\mathcal{O}}, T, O, R, F \rangle$, where:

- I is a finite set of k agents,
- S is a finite set of states,
- $\bar{A} = \times_i A_i$ is a finite set of joint actions,
- $\bar{\mathcal{O}} = \times_i \mathcal{O}_i$ is a finite set of joint observations, and
- $T : S \times \bar{A} \times S \to [0, 1]$ is a state transition function where $T(s' \mid s, \bar{a})$ is the probability of transitioning into state s' given joint action $\bar{a}$ being performed in state s,
- $O : \bar{\mathcal{O}} \times S \times \bar{A}$ is a joint-observation function where $O(\bar{o}' \mid s', \bar{a})$ specifies the probability of attaining joint-observation $\bar{o}'$ when joint action $\bar{a}$ results in state s',
- $R : S \times \bar{A} \to \mathbb{R}$ is a reward function such that $R(s, \bar{a})$ is the immediate reward for performing joint action $\bar{a}$ in state s, and
- $F(\tau \mid s, \bar{a})$ is the *communication sojourn distribution*, defining the probability an agent remains decentralized for duration τ conditioned on the current state and joint action taken.

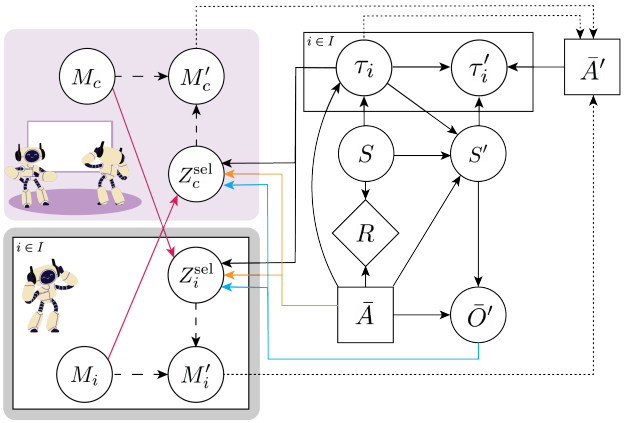


Figure 2: The SDec-POMDP Bayes net. Policy infrastructure is shown on the left and world model on the right. The purple backdrop is the blackboard with memory M_c generated from the histories of communicating agents. The gray backdrop with plate notation includes the individual agent memories M_i. Z selector nodes are selectively toggled by $\bar{\tau}$ to facilitate memory propagation η, represented by dashed lines. Policy ψ edges are represented by dotted lines.

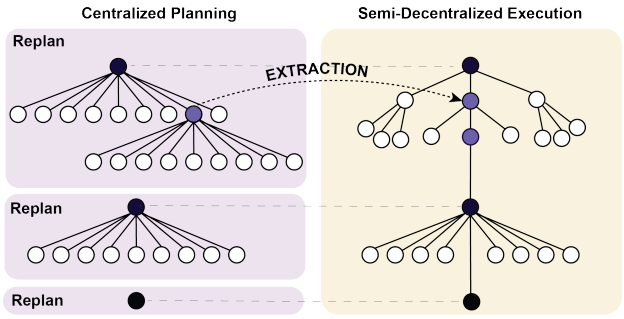


Figure 3: SDecMCTS algorithm outline.

Definition 4. Centralized Planning Semi-Decentralized Execution. Centralized Planning Semi-Decentralized Execution (CPSDE) is a multiagent planning regime in which agents conduct centralized offline planning, but toggle between centralized and decentralized policies online. Rather than executing only the next joint action, the agents periodically extract a semi-decentralized contingent policy over private histories, allowing them to continue acting during communication gaps. Execution proceeds open-loop with respect to centralized communication, but closed-loop with respect to each agent's action-observation history. When the agents return to a communication state, their histories are merged and centralized re-planning resumes. This formulation preserves the decision quality of centralized belief-space planning where communication permits it, while explicitly adhering to the constraints induced by semi-decentralized execution.

SDecMCTS

Overview. Following from CPSDE, SDecMCTS performs classical Monte Carlo tree search from a synchronized belief, then converts the resulting centralized search tree into a semi-decentralized execution policy (see Figure 3). Policy extraction maps private observation histories to local actions by aggregating the centralized tree's joint-action evidence over information-compatible branches. When extraction encounters unsupported or weakly visited histories, fallback mechanisms can use conservative replanning, heuristic completion, robust visit-count selection, or safe default actions. The method therefore preserves centralized search quality where evidence exists while maintaining executable local policies between synchronization events.

Policy Extraction. To extract a decentralized policy during execution, the extraction mechanism groups visited decentralized nodes by depth and their joint private-history cluster, then projects those clusters into per-agent information states conditioned on last synchronized belief plus local private history. We then solve for a factored policy mapping each agent's local information state to a local action. The extraction objective maximizes the expected centralized tree value under the empirical distribution of visited private-history clusters, using the backed-up joint-action values from the centralized tree with shrinkage toward heuristic or prior values in low-count regimes. Small factored optimization problems are solved using exhaustive enumeration over candidate local policy maps. Larger factored optimization problems use an iterative best-response solver.

Fallback Techniques. Unlike the analytical MAA* algorithms, SDecMCTS rarely generates a complete policy that considers every reachable joint observation history. This is because finite-budget search may not visit every private-history branch that can arise during execution, in planning. SDecMCTS therefore includes fallback mechanisms to make extracted policies executable without artificially re-centralizing the agents. During extraction, actions that were never evaluated for a private-history cluster can be disallowed, preventing the decentralized policy from selecting unsupported branches of the centralized tree. For low-support clusters, a configurable heuristic fallback ratio permits replacement of fragile extracted decisions with a tiebreak action derived from the model-based prior, such as POMDP, QMDP, lookahead, or, when a neural policy is active, the learned policy logits. Fallback is intended as a robustness mechanism rather than the primary decision rule; high-quality SDecMCTS runs should have low fallback fractions.

SDecZero

Overview. In the style of AlphaZero, BetaZero, and MuZero, SDecZero extends SDecMCTS with learned neural guidance to enable scaling (see Figure 4). SDecZero applies BetaZero to belief-space semi-decentralized planning while retaining a flat joint-action policy head. The core planner remains centralized MCTS with PUCT exploration operating over belief states. Beliefs may either be represented using an exact sparse Bayesian updater or a particle filter. Policy

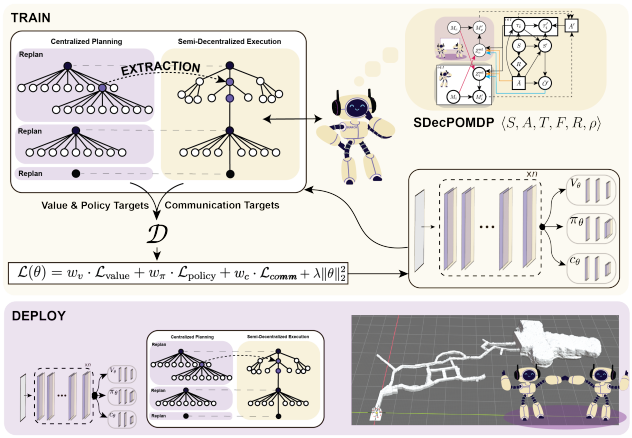


Figure 4: SDecZero algorithm outline.

targets are derived from search statistics at the corresponding belief, typically through MaxN-style visit distributions, with optional Q-weighted variants implemented. SDecZero can either train under centralized replanning and apply SDec extraction at evaluation time, or execute an SDec-aware data-generation path in which centralized trees are built at sync beliefs, semi-decentralized policies are extracted, and those policies are executed until the next sync or terminal state.

We also support auxiliary SDec-relevant training rows: centralized trigger episodes can identify private-history nodes that lie on extracted semi-decentralized paths, and those beliefs can be labeled with centralized PUCT policy targets and closed-loop target-mode values. We optionally consider the case of learned communication dynamics by adding a communication head to the network. Value and policy targets are gathered from the centralized inner planning loop, and where applicable, communication labels are gathered from the in-environment semi-decentralized outer loop. The resulting algorithm amortizes expensive centralized planning into a reusable neural guide while evaluating the policy under the same semi-decentralized communication constraints used at deployment.

Experiments

We conduct both planning (SDecMCTS) and learning (SDecZero) experiments to evaluate both algorithms across various problem types. SDecMCTS is evaluated against RS-MAA*, RS-SDA*, and Dec-MCTS on small semi-decentralized benchmarks including SDEC-TIGER, SDEC-MARS, MARITIMEMEDEVAC, and various SEARCH AND RETURN labyrinth problems. SDecZero is evaluated against IPPO, MAPPO, QMIX, and MASAC on large semi-decentralized benchmarks including SEARCH AND RETURN labyrinth problems from the DARPA Subterranean Site Challenge. Collectively, these baselines comprise the current state-of-the-art for decentralized and semi-decentralized multiagent planning and learning.

Setup. All experiments were conducted on an AMD Ryzen 9 9900X3D 12-Core Processor (4400 MHz), with timeout

occurring at 20 minutes. Approximate RS-SDA* is implemented in Python 3, and code is available at https://github.com/mahdial-husseini/RSSDA to support reproducibility. We compare directly against the RS-MAA* implementation provided by Koops, Junges, and Jansen (2024b). MAPPO, IPPO, and QMIX were sourced from the BenchMARL library (Bettini, Prorok, and Moens 2024), and MASAC sourced from the MASAC library (Felten 2023).

Canonical Benchmarks. We evaluate SDecMCTS on three canonical SDec-POMDP benchmarks: SDEC-TIGER, SDEC-MARS, and MARITIMEMEDEVAC (Al-Husseini, Wray, and Kochenderfer 2026). SDEC-TIGER uses action-based communication triggers and has stochastic observation dynamics; RS-SDA* interleaved results are therefore presented in Table 1 as average values with standard errors across 128 paired seeds. SDEC-MARS and MARITIMEMEDEVAC use state-based communication triggers. RS-MAA* is presented as the decentralized lower bound and RS-SDA* with all states and joint-actions centralized is the centralized upper bound.

We also test using LABYRINTH SEARCH & RETURN, a two-agent graph search problem, and LABYRINTH SEARCH & RESCUE, a more challenging variant with stochastic observations. In LABYRINTH SEARCH & RETURN, agents navigate a graph to locate a hidden target node selected uniformly from the environment and then return to the start node with the target information. The agents receive a step cost of -1 and a terminal reward of $+100$ if either returns to the start node having found the target. Agents can only share information and take joint actions when within line-of-sight. LABYRINTH SEARCH & RESCUE is a high-stakes perception and assistance problem. Agents are equipped with noisy sensors that return binary observations correlated with the target's presence at the current node. The action space is augmented with a terminal ASSIST action. Agents must locate the hidden target using uncertain sensor data and commit to assistance. Both problems are tested using the six generated labyrinth structures in Figure 5.

Real-World Cave Systems We use the *Prospector* simulator (Ward et al. 2025) to simulate caves from the DARPA Subterranean Challenge dataset, which features real world data from the Louisville Mega Cavern, a former limestone mine in Kentucky. Prospector re-meshes the 3D data provided by DARPA, and provides simulations of other real-world caves using data gathered from caving expeditions. We tranforms these maps into realistic traversability and coordination graphs, as seen in Figure 6.

Experimental Results

Table 1 evaluates SDecMCTS on canonical semi-decentralized planning benchmarks and on the larger Labyrinth Search & Return and Search & Rescue families. Across all SDec-POMDP benchmarks, SDecMCTS consistently recovers most of the value of centralized planning while preserving semi-decentralized execution. On SDEC-TIGER, SDecMCTS remains close to the Exact RS-SDA* centralized upper bound for all horizons, obtaining 60.46, 71.60, 92.10, and 121.04 at horizons 10, 12, 15, and

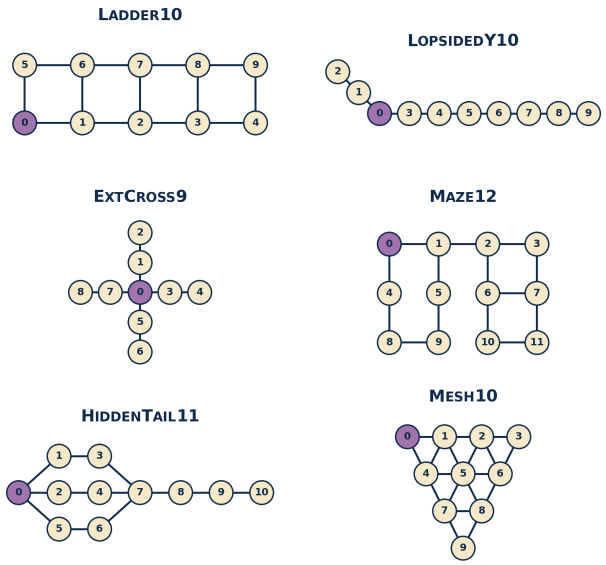


Figure 5: Simulated Labyrinth Layouts

20, respectively, while requiring less than one second per instance. In contrast, Dec-MCTS produces substantially lower returns, ranging from 22.83 to 30.71, and requires between 18 and 41 seconds. This gap is especially pronounced at longer horizons, where fixed-period communication fails to exploit the state and action-dependent coordination opportunities modeled by the SDec-POMDP.

A similar trend holds on SDEC-MARS and MARITIMEMEDEVAC. On SDEC-MARS, SDecMCTS tracks the centralized upper bound closely, achieving 27.41 at horizon 10 compared with 28.61 for Exact RS-SDA*, while Dec-MCTS reaches only 21.85. On MARITIMEMEDEVAC, SDecMCTS again remains near the centralized benchmark and outperforms Dec-MCTS on three of the four horizons. Small empirical exceedances of the exact point value, such as the horizon-8 MARITIMEMEDEVAC result, should be interpreted as practical parity under finite-seed evaluation rather than as a violation of the centralized upper-bound relationship.

The Labyrinth Search & Return results further demonstrate that SDecMCTS scales beyond the small canonical problems while retaining strong solution quality. Across all six graph families, SDecMCTS substantially improves over Dec-MCTS at every reported horizon. For example, on LABYRINTH MESH10, SDecMCTS achieves 85.67, 96.04, and 95.48 for horizons 5, 6, and 7, compared with 53.61, 75.23, and 80.13 for Dec-MCTS. SDecMCTS also remains close to Exact RS-SDA* on nearly all Search & Return instances, often matching the centralized value exactly or within a small absolute margin. This includes exact parity on EXTCROSS9 at horizon 6, LADDER10 at horizon 6, HIDDENTAIL11 at horizon 6, and MESH10 at horizon 5. The largest Search & Return gap occurs on MAZE12 at horizon 6, where the more complex topology leaves a larger gap to the centralized upper bound; nevertheless, SDecMCTS still

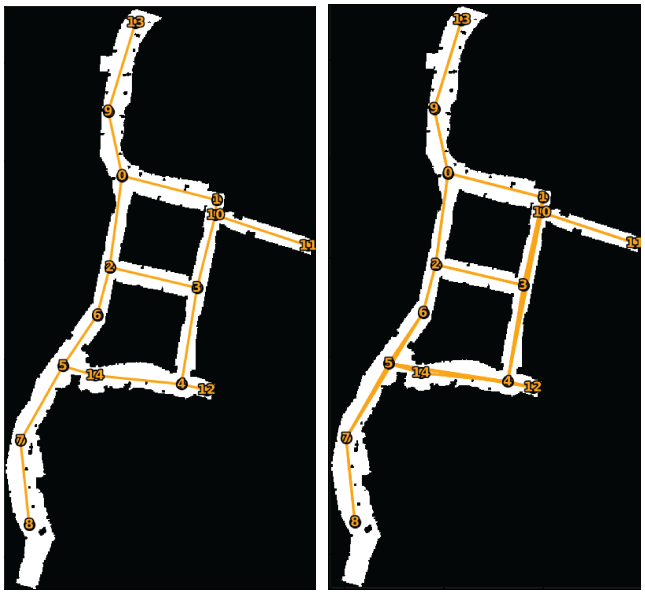
Table 1: SDecMCTS performance across canonical semi-decentralized planning benchmark categories. TO denotes timeout (>1200s). Averaged returns were performed over 128 paired seeds.

	decentralized		*periodic*		*semi-decentralized*				*centralized*	
	RS-MAA* (lower bound)		Dec-MCTS		App. RS-SDA*		SDecMCTS **(Our Approach)**		Exact RS-SDA* (upper bound)	
h	value	time	value	time	value	time	value	time	value	time
SDec-Tiger										
10	13.57	2	22.83 ± 6.33	18	60.26 ± 0.86	1	60.46 ± 2.39	<1	60.51	1
12	20.76	2	23.92 ± 7.41	27	73.61 ± 2.85	1	71.60 ± 3.09	<1	73.39	1
15	25.95	2	25.92 ± 8.60	37	94.76 ± 3.09	2	92.10 ± 3.64	<1	92.67	1
20	28.01	3	30.71 ± 10.78	41	120.94 ± 4.46	4	121.04 ± 4.23	<1	125.19	1
SDec-Mars (Right Band Rendezvous)										
7	20.90	2	15.40 ± 0.67	3	21.17	1	20.19 ± 0.56	<1	21.17	<1
8	22.48	2	17.56 ± 1.15	5	23.65	1	23.34 ± 0.27	<1	23.65	<1
9	24.31	3	19.04 ± 1.00	5	25.21	2	25.03 ± 0.30	1	25.70	<1
10	26.31	4	21.85 ± 1.11	7	28.28	2	27.41 ± 0.54	1	28.61	<1
MaritimeMEDEVAC										
7	3.25	<1	4.83 ± 1.12	16	6.36	<1	5.74 ± 0.58	<1	6.62	<1
8	8.03	<1	9.56 ± 1.37	22	10.61	<1	11.26 ± 0.52	<1	10.88	<1
9	10.79	<1	11.78 ± 1.11	27	12.74	<1	12.48 ± 0.43	<1	13.01	<1
10	12.15	<1	13.51 ± 0.88	30	13.35	<1	13.26 ± 0.32	<1	13.61	<1
Labyrinth ExtCross9 Search & Return (N = 2)										
6	71.25	15	79.83 ± 5.42	80	84.38	1	84.38 ± 0.00	2	84.38	1
7	-	TO	77.25 ± 9.41	104	84.25	2	83.60 ± 0.03	3	84.25	5
8	-	TO	86.17 ± 11.45	108	96.75	1	96.75 ± 0.08	5	96.75	1
Labyrinth LopsidedY10 Search & Return (N = 2)										
5	40.78	<1	46.43 ± 18.72	44	74.67	<1	74.38 ± 0.02	<1	74.67	<1
6	51.44	<1	63.00 ± 17.77	66	85.67	<1	85.30 ± 0.04	<1	85.67	<1
7	51.00	<1	76.03 ± 15.13	87	96.78	<1	95.94 ± 0.06	<1	96.78	<1
Labyrinth Ladder10 Search & Return (N = 2)										
5	40.78	<1	49.47 ± 18.55	50	74.56	<1	74.37 ± 0.02	<1	74.67	<1
6	62.67	<1	55.67 ± 18.33	79	95.89	<1	96.33 ± 0.00	1	96.33	<1
7	62.33	<1	75.83 ± 15.09	95	96.44	<1	96.07 ± 0.12	4	96.78	<1
Labyrinth Maze12 Search & Return (N = 2)										
5	32.45	2	39.47 ± 18.51	65	60.09	<1	59.90 ± 0.00	1	60.18	1
6	59.35	2	60.18 ± 12.53	94	77.64	<1	77.62 ± 0.02	3	86.90	1
7	-	TO	80.16 ± 10.67	135	94.82	<1	95.46 ± 0.00	12	96.00	1
Labyrinth HiddenTail11 Search & Return (N = 2)										
4	36.80	<1	33.23 ± 17.81	50	57.20	<1	57.06 ± 0.03	<1	57.20	<1
5	36.20	13	36.00 ± 18.28	90	66.90	<1	66.10 ± 0.05	2	66.90	1
6	65.90	2	48.80 ± 18.66	128	95.80	2	96.20 ± 0.00	4	96.20	1
Labyrinth Mesh10 Search & Return (N = 2)										
5	52.00	<1	53.61 ± 18.42	23	85.67	1	85.67 ± 0.00	4	85.67	3
6	73.89	6	75.23 ± 17.24	91	96.67	2	96.04 ± 0.03	14	96.78	7
7	-	TO	80.13 ± 12.53	111	96.67	3	95.48 ± 0.15	37	96.78	8
Labyrinth ExtCross9 Search & Rescue (N = 2)										
10	17.62	3	-	-	80.11 ± 4.80	14	81.19 ± 3.03	14	-	TO
Labyrinth LopsidedY10 Search & Rescue (N = 2)										
10	39.91	62	-	-	65.00 ± 5.55	4	59.89 ± 6.32	11	-	TO
Labyrinth Ladder10 Search & Rescue (N = 2)										
10	36.03	59	-	-	83.00 ± 4.29	19	86.63 ± 2.76	13	-	TO
Labyrinth Maze12 Search & Rescue (N = 2)										
10	21.53	62	-	-	43.75 ± 5.24	35	64.67 ± 5.40	16	-	TO
Labyrinth HiddenTail11 Search & Rescue (N = 2)										
10	21.02	65	-	-	73.26 ± 5.10	41	61.50 ± 4.45	11	-	TO
Labyrinth Mesh10 Search & Rescue (N = 2)										
10	48.20	17	-	-	88.10 ± 2.54	115	88.00 ± 2.53	33	-	TO

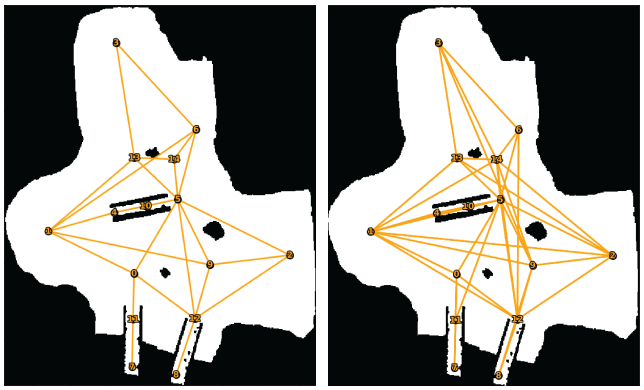
improves substantially over both RS-MAA* and Dec-MCTS.

The Search & Rescue variant introduces noisy observations and terminal assist decisions, making the value of semi-decentralized coordination more visible. Exact RS-SDA* times out on all six Search & Rescue instances, indicating that exact centralized solution methods do not scale to this setting within the 20-minute limit. SDecMCTS remains tractable on every instance, producing policies in 11–33 seconds. It is competitive with Approximate RS-SDA*, outperforming it on ExtCross9, Ladder10, and Maze12, matching it closely on Mesh10, and underperforming it on LopsidedY10 and HiddenTail11. The strongest improvement appears on Maze12, where SDecMCTS obtains 64.67 compared with 43.75 for Approximate RS-SDA*. These results suggest that the extracted semi-decentralized policy is especially useful when stochastic observations and graph structure make premature local commitment costly.

Table 2 evaluates the learned SDecZero variants on two DARPA Subterranean cave-system tasks. SDecZero with



(a) *Tunnels* traversability graph. (b) *Tunnels* coordination graph.



(c) *Chamber* traversability graph. (d) *Chamber* coordination graph.

Figure 6: Fifteen node two-dimensional traversability and coordination graphs for two DARPA Subterranean Challenge cave sites: *Tunnels* and *Chamber*. The traversability graph has an edge between two nodes if they are spatial neighbors, and the coordination graph has an edge between two nodes if they have a line-of-sight connection.

known communication dynamics obtains 92.31 on DARPA 3D Tunnels-15 and 95.11 on DARPA 3D Chambers-15 after 50K environment steps. These returns exceed the strongest model-free baseline by large absolute margins: on Tunnels, the best baseline is IPPO at 62.10, and on Chambers, the best baseline is MAPPO at 62.50. Thus, SDecZero improves over the strongest baseline by 30.21 and 32.61 return points while using one third of the reported environment interaction budget.

The learned-communication variant, SDecZero with a communication head, remains close to the known-communication version despite having to infer the communication-sojourn

Table 2: SDecZero performance across DARPA Subterranean Cave Site Tunnel and Chamber Systems.

	SDecZero (Our Approach)		SDecZero (learned communication) (Our Approach)		MAPPO		IPPO		QMIX		MASAC	
h	value	env steps	value	env steps	value	env steps	value	env steps	value	env steps	value	env steps
DARPA 3D Tunnels-15 Search & Return (N = 2)												
11	92.31	50K	90.31	150K	57.30	150K	62.10	150K	-3.14	150K	44.30	150K
DARPA 3D Chambers-15 Search & Return (N = 2)												
11	95.11	50K	94.09	150K	62.50	150K	42.60	150K	-3.14	150K	53.33	150K

model from interaction. It obtains 90.31 on Tunnels and 94.09 on Chambers, only 2.00 and 1.02 points below the corresponding known-communication SDecZero results. At the same 150K environment-step budget used by MAPPO, IPPO, QMIX, and MASAC, the learned-communication variant still outperforms all model-free baselines by wide margins. QMIX fails to learn useful behavior in both DARPA tasks, producing a return of -3.14, while MASAC, MAPPO, and IPPO learn partially successful policies but remain far below either SDecZero variant. These results support the central claim that amortized semi-decentralized search provides a more effective inductive bias than strict decentralized execution in domains with intermittent line-of-sight coordination.

Figure 7 evaluates whether the communication head learns the underlying communication dynamic rather than merely improving task return. The heatmaps visualize the predicted communication probability $p_\theta(\text{comm} \mid b, a)$ across training iterations and compares it with the oracle communication map. On SDec-Tiger, SDec-Mars, and MaritimeMedevac, the learned communication head rapidly suppresses false positives and recovers the sparse oracle structure within the first few training iterations. The accompanying Brier-score curve confirms this visual trend: calibration error drops sharply for the compact benchmarks and remains low thereafter.

The Labyrinth ExtCross9 communication function is more difficult. Its oracle map has a structured, graph-induced pattern rather than a small set of isolated triggers, and the learned heatmaps remain visibly noisier than the compact-domain predictions. Even in this harder setting, however, training steadily reduces the Brier score and moves the predicted communication map toward the oracle structure. This result is important for two reasons. First, it shows that the communication head is learning a reusable model of the semi-decentralized information process, not only overfitting to return. Second, it explains the modest performance gap between SDecZero and SDecZero with learned communication in Table 2: learning communication dynamics introduces estimation error, but the learned model is accurate enough to preserve most of the benefit of semi-decentralized search.

Overall, the results show a consistent progression across planning, learning with known communication, and learning with inferred communication. SDecMCTS establishes that semi-decentralized search can approach centralized planning quality on small and medium-scale benchmarks while avoiding the brittleness of fixed-period Dec-MCTS. SDecZero then amortizes this search procedure and scales it to realistic cave-system graphs, substantially outperforming standard model-free MARL baselines. Finally, the learned-communication-head experiment demonstrates that the stochastic communication model itself can be recovered from data with sufficient accuracy to support high-return semi-decentralized execution.

Limitations and Future Work

There are limitations to our existing work, however promising, that remain to be addressed. Current evidence suggests that *centralized performance alone can have difficulty generating high-quality semi-decentralized policies in many benchmarks that require strategic coordination*! Although the centralized inner planning tree performs well in fully centralized regimes, including 25 and 30-node DARPA Subterranean Challenge Site instances, extracting high-quality semi-decentralized policies for these larger problems remains elusive. One issue is that the centralized root action may be selected without accounting for the semi-decentralized continuation induced by downstream communication loss. This can degrade performance in tasks requiring tightly coordinated joint actions. Addressing this mismatch through SDec-aware backups and extracted-continuation value targets is an important direction for scaling SDecZero. We also look forward to implementing our methods in SMAC and SMAClite, which should further challenge the algorithm at higher agent counts.

Conclusion

We introduce two new scalable algorithmic approaches to semi-decentralized multiagent control, **SDecMCTS** and **SDecZero**, evaluated across canonical benchmarks and complex labyrinth layouts. SDecMCTS shows that centralized belief-space search can be converted into executable semi-decentralized policies that closely match centralized solution quality while avoiding the combinatorial failures of exact planning in larger domains. SDecZero further amortizes this search with learned policy, value, and communication models, enabling strong performance on realistic DARPA subterranean cave-system tasks where standard model-free MARL baselines struggle under sparse rewards and stochastic information flow. Together, these results demonstrate that explicitly modeling intermittent communication is a practical and effective path toward scalable multiagent control in environments that are neither fully centralized nor fully decentralized.

Collaboration

Both team members contributed meaningfully and equally towards all aspects of this research project. Mahdi prioritized the algorithm design and literature review, and Avi prioritized baseline development and experiments.

References

Al-Husseini, M.; Wray, K. H.; and Kochenderfer, M. J. 2026. A Semi-Decentralized Approach to Multiagent Control. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Al-Husseini, M.; Wray, K. H.; Ward, I. R.; and Kochenderfer, M. J. 2026. Approximate Heuristic Search for Semi-

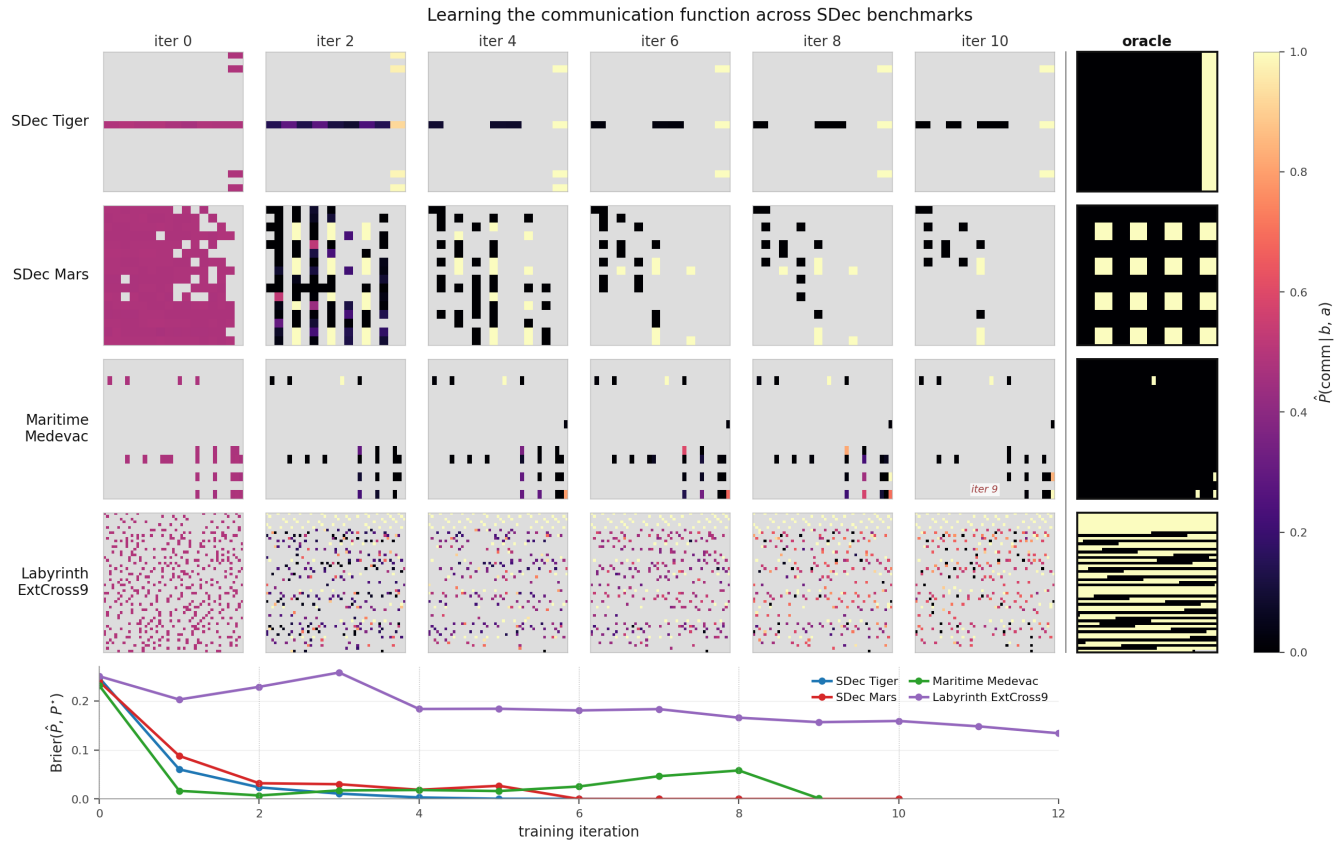


Figure 7: Learning the communication dynamics across SDec benchmarks. Heatmaps show predicted communication probabilities over training, compared with the oracle communication map. The bottom panel reports Brier calibration error.

Decentralized Systems. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Amato, C. 2024. An Introduction to Centralized Training for Decentralized Execution in Cooperative Multi-Agent Reinforcement Learning. *ArXiv*, abs/2409.03052.

Best, G.; Cliff, O. M.; Patten, T.; Mettu, R. R.; and Fitch, R. 2019. Dec-MCTS: Decentralized planning for multi-robot active perception. *Int. J. Rob. Res.*, 38(2–3): 316–337.

Bettini, M.; Prorok, A.; and Moens, V. 2024. BenchMARL: Benchmarking Multi-Agent Reinforcement Learning. *Journal of Machine Learning Research*, 25(217): 1–10.

Craig, I. D. 1988. Blackboard systems. *Artificial Intelligence Review*, 2(2): 103–118.

Das, A.; Gervet, T.; Romoff, J.; Batra, D.; Parikh, D.; Rabbat, M.; and Pineau, J. 2019. TarMAC: Targeted Multi-Agent Communication. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 1538–1546. PMLR.

Dolan, S.; Nayak, S.; and Balakrishnan, H. 2023. Satellite Navigation and Coordination with Limited Information Sharing. In Matni, N.; Morari, M.; and Pappas, G. J., eds., *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, 1058–1071. PMLR.

Egorov, V.; and Shpilman, A. 2022. Scalable Multi-Agent Model-Based Reinforcement Learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, 381–390. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

Erman, L. D.; Hayes-Roth, F.; Lesser, V. R.; and Reddy, D. R. 1980. The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys (CSUR)*, 12(2): 213–253.

Felten, F. 2023. MASAC: A Multi-Agent Soft-Actor-Critic implementation for PettingZoo. https://github.com/ffelten/MASAC.

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).

Foerster, J. N.; Assael, Y. M.; de Freitas, N.; and Whiteson, S. 2016. Learning to communicate with Deep multi-agent reinforcement learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*,

NIPS'16, 2145–2153. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510838819.

Hao, X.; Hao, J.; Xiao, C.; Li, K.; Li, D.; and Zheng, Y. 2024a. Multiagent gumbel MuZero: efficient planning in combinatorial action spaces. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press. ISBN 978-1-57735-887-9.

Hao, X.; Hao, J.; Xiao, C.; Li, K.; Li, D.; and Zheng, Y. 2024b. Multiagent gumbel MuZero: efficient planning in combinatorial action spaces. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press. ISBN 978-1-57735-887-9.

Jiang, J.; and Lu, Z. 2018. Learning attentional communication for multi-agent cooperation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, 7265–7275. Red Hook, NY, USA: Curran Associates Inc.

Koops, W.; Jansen, N.; Junges, S.; and Simao, T. D. 2023. Recursive small-step multi-agent A* for dec-POMDPs.

Koops, W.; Junges, S.; and Jansen, N. 2024a. Approximate dec-POMDP solving using multi-agent A*. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24. ISBN 978-1-956792-04-1.

Koops, W.; Junges, S.; and Jansen, N. 2024b. Approximate Dec-POMDP solving using multi-agent A*.

Li, J.; Xu, R.; Liu, X.; Ma, J.; Chi, Z.; Ma, J.; and Yu, H. 2023. Learning for Vehicle-to-Vehicle Cooperative Perception Under Lossy Communication. *IEEE Transactions on Intelligent Vehicles*, 8(4): 2650–2660.

Li, M.; Yang, W.; Cai, Z.; Yang, S.; and Wang, J. 2019. Integrating decision sharing with prediction in decentralized planning for multi-agent coordination under uncertainty. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, 450–456. AAAI Press. ISBN 9780999241141.

Liu, Q.; Ye, J.; Ma, X.; Yang, J.; Liang, B.; and Zhang, C. 2024a. Efficient Multi-agent Reinforcement Learning by Planning. In Kim, B.; Yue, Y.; Chaudhuri, S.; Fragkiadaki, K.; Khan, M.; and Sun, Y., eds., *International Conference on Learning Representations*, volume 2024, 49499–49520.

Liu, Q.; Ye, J.; Ma, X.; Yang, J.; Liang, B.; and Zhang, C. 2024b. Efficient Multi-agent Reinforcement Learning by Planning. In *The Twelfth International Conference on Learning Representations*.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 6382–6393. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.

Oliehoek, F. A.; Amato, C.; et al. 2016. *A concise introduction to decentralized POMDPs*. Springer.

Rashid, T.; Samvelyan, M.; De Witt, C. S.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *J. Mach. Learn. Res.*, 21(1).

Rouček, T.; Pecka, M.; Čížek, P.; Petříček, T.; Bayer, J.; Šalanský, V.; Heřt, D.; Petrlík, M.; Báča, T.; Spurný, V.; Pomerleau, F.; Kubelka, V.; Faigl, J.; Zimmermann, K.; Saska, M.; Svoboda, T.; and Krajník, T. 2019. DARPA Subterranean Challenge: Multi-robotic Exploration of Underground Environments. In *Modelling and Simulation for Autonomous Systems: 6th International Conference, MESAS 2019, Palermo, Italy, October 29–31, 2019, Revised Selected Papers*, 274–290. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-030-43889-0.

Saboia, M.; Clark, L.; Thangavelu, V.; Edlund, J. A.; Otsu, K.; Correa, G. J.; Varadharajan, V. S.; Santamaria-Navarro, A.; Touma, T.; Bouman, A.; Melikyan, H.; Pailevanian, T.; Kim, S.-K.; Archanian, A.; Vaquero, T. S.; Beltrame, G.; Napp, N.; Pessin, G.; and Agha-mohammadi, A.-a. 2022. ACHORD: Communication-Aware Multi-Robot Coordination With Intermittent Connectivity. *IEEE Robotics and Automation Letters*, 7(4): 10184–10191.

Santos, P. P.; Carvalho, D. S.; Vasco, M.; Sardinha, A.; Santos, P. A.; Paiva, A.; and Melo, F. S. 2025. Centralized training with hybrid execution in multi-agent reinforcement learning via predictive observation imputation. *Artif. Intell.*, 348(C).

Singh, A.; Jain, T.; and Sukhbaatar, S. 2019. Individualized Controlled Continuous Communication Model for Multiagent Cooperative and Competitive Tasks. In *International Conference on Learning Representations*.

Su, J.; Adams, S.; and Beling, P. 2021. Value-Decomposition Multi-Agent Actor-Critics. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13): 11352–11360.

Szer, D.; Charpillet, F.; and Zilberstein, S. 2005. MAA*: a heuristic search algorithm for solving decentralized POMDPs. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, 576–583. Arlington, Virginia, USA: AUAI Press. ISBN 0974903914.

Tang, S.; Chen, J.; and Lan, T. 2026. MALinZero: Efficient Low-Dimensional Search for Mastering Complex Multi-Agent Planning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Wang, J.; Ren, Z.; Liu, T.; Yu, Y.; and Zhang, C. 2021. {QPLEX}: Duplex Dueling Multi-Agent Q-Learning. In *International Conference on Learning Representations*.

Ward, I. R.; Paral, M.; Riordan, K.; Ho, M.; Adang, M.; and Kochenderfer, M. J. 2025. Prospector: a Cave Simulation Environment for Rotorcraft. https://github.com/isaac-ward/prospector.

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. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713871088.

Zhao, C.; Duan, X.; Cai, L.; and Cheng, P. 2021. Vehicle Platooning With Non-Ideal Communication Networks. *IEEE Transactions on Vehicular Technology*, 70(1): 18–32.

Zhao, Z.; Zuo, H.; Shen, Q.; and Wang, W. 2025. Event-triggered Distributed Model Predictive Control for Spacecraft Formation with Collision Avoidance. *IFAC-PapersOnLine*, 59(20): 905–910. 23th IFAC Symposium on Automatic Control in Aerospace ACA 20255.

Zhu, C.; Dastani, M.; and Wang, S. 2024a. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, 38(1).

Zhu, C.; Dastani, M.; and Wang, S. 2024b. A Survey of Multi-Agent Deep Reinforcement Learning with Communication. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, 2845–2847. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400704864.