SDEC-MAT: SEQUENCE MODELING FOR SEMI-DECENTRALIZED MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In real-world multi-agent systems, imperfect communication makes purely centralized execution challenging. Centralized training with decentralized execution is a common approach under constrained communication, typically with parameter sharing and often with observation sharing among connected neighbors. These methods perform strongly on common benchmarks, which rarely demand strict coordination. Their tasks are either solvable without it or expose information that lets independently acting agents coordinate implicitly. We introduce the Multi-Agent Counting (MAC) diagnostic suite, in which an exact number of agents must take a specified action while communication links change over time. We prove that parameter-shared decentralized policies cannot reliably solve such tasks when agents' observations are identical, and that, when team membership varies, even agents with distinct observations cannot in general learn to guarantee the exact count. We propose the Semi-Decentralized Multi-Agent Transformer (SDec-MAT), which builds on the Multi-Agent Transformer and the Semi-Decentralized POMDP framework. SDec-MAT masks its attention with the current communication graph, so each agent sees only its neighbors' observations and chooses its action based on the actions of neighbors that act before it, while agents without links act independently. Because connected sub-teams decide jointly, SDec-MAT attains high success across the MAC suite, where baselines given agent IDs succeed only inconsistently, and generalizes zero-shot to agent dropouts, team partitions and larger multi-target teams better than these baselines. We further extend a graph-based particle navigation task into a joint decision-making task in which targets are captured only by simultaneous actions of at least or exactly three agents. Decentralized baselines complete the task under the at-least rule but with frequent false captures, whereas among methods with nonzero success, SDec-MAT makes the fewest false captures in every condition and is the only method to succeed under the exact rule with a high false-capture penalty. SDec-MAT also remains competitive on the StarCraft Multi-Agent Challenge.

1 INTRODUCTION

Cooperative multi-agent systems often aim for centralized control, where a single global policy controls all agents of a team (Claus & Boutilier, 1998; Oliehoek & Amato, 2016). However, real-world issues such as delayed or broken communication make pure centralization challenging. The standard approach to address this limitation is the Centralized Training with Decentralized Execution (CTDE) paradigm, which lets agents use global information in training, but act on their local observations at deployment (Rashid et al., 2020; Yu et al., 2022). Communicative CTDE variants reduce this isolation by letting agents share local messages (Sukhbaatar et al., 2016; Das et al., 2019).

CTDE methods perform strongly on standard benchmarks such as the StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019) for decentralized combat control and the Multi-Agent Particle Environment (MPE) (Lowe et al., 2017) for 2D physics-based navigation, and so are the default for teams under unreliable communication. Yet these benchmarks do not test strict coordination: their tasks are either solvable without it or expose information that lets independently acting agents coordinate (e.g., SMAC units differ in type, position and health, so they can take different roles

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

without communicating). We thus propose the Multi-Agent Counting (MAC) benchmark, in which exactly a specified number of agents must take a target action, in states where every agent receives an identical observation. We show, first, that parameter sharing (Gupta et al., 2017), the standard route to scalable and generalizable CTDE training, cannot reliably solve MAC, as a property of the shared policy rather than an artifact of training. Second, naive remedies such as appending agent IDs (Yu et al., 2022) break the symmetry and can be learned in training. However, we prove that when an agent's observation stays the same across team compositions, this remedy cannot in general learn to guarantee the exact count, and it degrades sharply in zero-shot tests with changed teams.

To achieve exact coordination under intermittent communication, we propose the Semi-Decentralized Multi-Agent Transformer (SDec-MAT), which explicitly tracks the live physical connectivity graph. We build on the Multi-Agent Transformer (MAT) (Wen et al., 2022), whose sequential action generation conditions each action on the previous ones in the chain, enabling joint decision-making while avoiding the exponential complexity of joint action spaces. Its decode order breaks symmetry among identical agents by introducing a dynamic asymmetry, at a cost of up to one sequential communication round per agent in a sub-team at each step. Moreover, MAT assumes full connectivity, so it cannot be directly applied to dynamic network topologies.

To adapt this paradigm to physical realities, we ground our method in the Semi-Decentralized POMDP (SDec-POMDP) framework (Al-Husseini et al., 2026). Shifting from CTDE to a Centralized Training, Semi-Decentralized Execution (CTSDE) paradigm, SDec-MAT feeds the real-time communication graph into the transformer's attention via hard masking, restricting joint decision-making to each agent's communication neighborhood. It thus coordinates exactly and sequentially where links exist and shifts smoothly to independent, decentralized execution when connections fail.

Our main contributions are: **(1)** We propose the MAC benchmark to evaluate the limitations of decentralized execution in strict symmetry-breaking coordination and intermittent communication tasks. **(2)** We introduce SDec-MAT, a novel sequence model that operationalizes the SDec-POMDP framework. It dynamically adjusts autoregressive decision-making to match real-time communication limits without requiring extra parameters. **(3)** We empirically demonstrate that SDec-MAT resolves exact-coordination bottlenecks and achieves zero-shot fault tolerance to agent dropouts and network partitions while maintaining competitive performance on SMAC.

2 Related Work

Parameter sharing and behavioral symmetry. CTDE answers non-stationarity with centralized critics (Lowe et al., 2017), including multi-agent proximal policy optimization (MAPPO; Yu et al., 2022), or value factorization (Sunehag et al., 2018; Rashid et al., 2020). What is deployed is still a product of independent local policies (Claus & Boutilier, 1998). Parameter sharing (Gupta et al., 2017) makes that product symmetric: on coinciding local inputs the joint action law is a product of identical marginals, so a target count is sampled rather than decided (Section 4.1). Likewise, deterministic leader election is impossible in anonymous symmetric networks (Angluin, 1980). A centralized planner exploits the same anonymity, reasoning over counts alone (Robbel et al., 2016), and any subset of the target size attains its optimum. MAC therefore tests which agents act, not how many. Agent indication and selective sharing restore asymmetry but bind the policy to a fixed roster (Terry et al., 2020; Christianos et al., 2021). The universal policy decoupling transformer frees the input and role learning frees the behavioral marginal (Hu et al., 2021; Wang et al., 2020; Jiang & Lu, 2021), yet in every case the joint action stays factorized.

Routing information over a graph. Communication enriches what an agent conditions on, through pooled (Sukhbaatar et al., 2016), differentiable (Foerster et al., 2016), gated (Jiang & Lu, 2018) and targeted (Das et al., 2019) channels, while leaving actions conditionally independent given messages: under deterministic messages and agreeing histories the product structure survives any number of rounds. Stochastic messages from a finite alphabet break it only through protocols that may run forever (Itai & Rodeh, 1990), whereas SDec-MAT breaks ties by decoding order within at most one sequential message per agent. Learned topologies place the graph inside the optimization (Hu et al., 2024; Niu et al., 2021) and assume every selected link is available, so execution-time partitions lie outside that optimization. InforMARL takes an exogenous proximity graph (Nayak et al., 2023) and is our closest baseline, but it leaves connected agents conditionally independent; we route the autoregressive decision chain itself along that graph.

Action dependence at execution. Coordination graphs make joint action selection dependent by maximizing a factored payoff (Guestrin et al., 2002). Max-Plus action selection inside Monte Carlo tree search admits state-dependent coordination graphs, such as edges between nearby drones (Choudhury et al., 2022). These graphs couple payoffs, not information, since every action is selected by online search over the full joint state of a multi-agent Markov decision process. Multi-agent common knowledge reinforcement learning delegates to subgroups sharing common knowledge over a fixed roster, and the Bayesian action decoder conditions on a public belief (Schroeder de Witt et al., 2019; Foerster et al., 2019). A second line makes the dependence sequential through the multi-agent advantage decomposition reviewed in Section 3.2, exploited by heterogeneous-agent proximal policy optimization only through a serial update of factorized policies (Kuba et al., 2022), and by MAT through a causal mask admitting parallel training (Wen et al., 2022); bidirectional action-dependent Q-learning (ACE; Li et al., 2023) reaches the same sequential structure without the decomposition. Unlike all of them, SDec-MAT routes this chain along a communication graph, reducing to MAT when that graph is complete.

Intermittent communication. Robustness methods target execution-time unreliability without changing the decision structure, through recurrent memory (Wang et al., 2019), message dropout (Kim et al., 2019), variational prediction (Fu et al., 2025), centrally trained autoregressive imputation (Santos et al., 2025) and value-aware weighting of that objective (Kafadar et al., 2026). A perfect imputer returns each agent to whatever policy consumes the restored input, leaving the symmetry of a shared factorized policy intact. Planning models instead price communication actions, lag a joint belief under delay, or place a distribution over the time until an agent regains an information-sharing state (Goldman & Zilberstein, 2003; Nayyar et al., 2011; Al-Husseini et al., 2026). By Corollary 1 of Al-Husseini et al. (2026), the finite-horizon SDec-POMDP is NEXP-complete. SDec-MAT learns a policy whose attention follows a communication graph it does not select.

3 Background

3.1 The SDec-POMDP

A decentralized partially observable Markov decision process (Dec-POMDP; Oliehoek & Amato, 2016; Kochenderfer et al., 2022) is a tuple $\langle I, S, \bar{A}, \bar{\mathcal{O}}, T, O, R\rangle$ with N agents $I = \{1, \dots, N\}$, a finite state set S, and joint sets $\bar{A} = \times_{i\in I} A_i$ and $\bar{\mathcal{O}} = \times_{i\in I} \mathcal{O}_i$ of each agent's finite action set A_i and observation set $\mathcal{O}_i$. An overline marks a tuple indexed by agents, $\bar{a} = (a_i)_{i\in I}$ and $\bar{o} = (o_i)_{i\in I}$; for $J \subseteq I$, $\bar{a}_J = (a_j)_{j\in J}$ and $\bar{o}_J = (o_j)_{j\in J}$ in increasing index order. The transition function $T(s' \mid s, \bar{a})$ gives the probability of next state $s' \in S$ after joint action $\bar{a} \in \bar{A}$ in state $s \in S$, the observation function $O(\bar{o}' \mid s', \bar{a})$ that of joint observation $\bar{o}' \in \bar{\mathcal{O}}$, and $R(s, \bar{a})$ is the shared reward. Agents cannot share information explicitly, so agent i draws $a_i \sim \pi_i(\cdot \mid h_i)$ from a local policy π_i conditioned only on its action-observation history $h_i \in (A_i \times \mathcal{O}_i)^\star$, and the joint policy $\bar{\pi} = \langle \pi_i \rangle_{i\in I}$ factorizes as $\bar{\pi}(\bar{a} \mid \bar{h}) = \prod_{i\in I} \pi_i(a_i \mid h_i)$ with $\bar{h} = (h_i)_{i\in I}$. The agents seek the $\bar{\pi}$ maximizing the expected return over decision epochs $t \in \mathbb{N}$, discounted by $\gamma \in [0, 1)$ and started from a state drawn from the initial belief b^0, a distribution over S. The multi-agent partially observable Markov decision process (MPOMDP) shares this tuple and lets every agent condition on $\bar{h}$.

Semi-decentralization. Al-Husseini et al. (2026) make communication semi-Markov through a sojourn communication time $\mathcal{T}_i$ for each agent $i \in I$, whose realization τ_i is the time until agent i returns to an information-sharing state. We write τ_i^t for its value at decision epoch t and, as for states and actions, mark the next epoch with a prime, so $\tau_i' = \tau_i^{t+1}$ when $\tau_i = \tau_i^t$. We take $\mathbb{P}(\mathcal{T}_i = 0) > 0$ for every $i \in I$, so $\tau_i^t = 0$ reads: agent i is in a communication epoch at epoch t. Writing $\bar{\tau} = (\tau_i)_{i\in I}$ for the joint realization, the SDec-POMDP $\langle I, S, \bar{A}, \bar{\mathcal{O}}, F, T, O, R\rangle$ adds the distribution $F(\bar{\tau}' \mid s', \bar{a}', \bar{\tau})$ of the joint sojourn communication time $\bar{\mathcal{T}} = (\mathcal{T}_i)_{i\in I}$, where $\bar{a}'$ is the next joint action, and conditions its dynamics on the communication state, $T(s' \mid s, \bar{a}, \bar{\tau})$ and $O(\bar{o}' \mid s', \bar{a}, \bar{\tau}')$. It propagates memories, actions and observations into agent memories and a blackboard through selector functions gated on $\bar{\tau}$, of which we use only the observation selector (Section A). Section 4.3 adopts a graph view rather than the full formalism: it represents the communication state at epoch t by a communication graph G^t, models neither F nor reconnection times, and restricts agent i to its one-hop neighborhood $\mathcal{N}^t(i)$, which equals the communicating set containing i for every $i \in I$ exactly when the connected components of G^t are cliques (Section A). A complete G^t at every epoch recovers the MPOMDP, and one with only self-loops at every epoch recovers the Dec-POMDP.

3.2 AUTOREGRESSIVE MULTI-AGENT REINFORCEMENT LEARNING

Following Wen et al. (2022) we write value functions over joint observations $\bar{o} \in \bar{\mathcal{O}}$ for a joint policy $\bar{\pi}(\bar{a} \mid \bar{o})$, which need not factorize over agents. For a permutation $i_{1:N} = (i_1, \ldots, i_N)$ of I, we write $\bar{a}_{i_{1:m}} = (a_{i_1}, \ldots, a_{i_m})$ and $\bar{o}_{i_{1:m}} = (o_{i_1}, \ldots, o_{i_m})$ for $m \in \{0, \ldots, N\}$, with $\bar{a}_{i_{1:0}}$ empty. The multi-agent advantage decomposition (Kuba et al., 2022) states that, for any such permutation and with no further assumption, the joint advantage of $\bar{a}$ at $\bar{o}$ equals the sum over m of the advantage of a_{i_m} given predecessor actions $\bar{a}_{i_{1:m-1}}$. Since it holds for every order, any fixed order is valid; Wen et al. (2022) re-draw the order at every training iteration for a Nash-equilibrium convergence argument. Conditioning on $\bar{a}_{i_{1:m-1}}$ removes the independence a factorized policy imposes, so ordered decision-making breaks symmetry among interchangeable agents.

MAT and the connectivity assumption. MAT (Wen et al., 2022) encodes $\bar{o}_{i_{1:N}}$ by unmasked self-attention into representations $\hat{o}_{i_1}, \ldots, \hat{o}_{i_N}$ and decodes $\pi_\theta(a_{i_m} \mid \hat{o}_{i_m}, \bar{a}_{i_{1:m-1}})$ with parameters θ under a lower-triangular causal mask, fitting θ with a clipped proximal policy optimization (PPO) surrogate (Schulman et al., 2017). Because the encoder attention is unmasked, every $\hat{o}_{i_m}$ depends on the joint observation $\bar{o}$, so executing π_θ for agent i_m requires $\bar{o}$ and every predecessor action regardless of the decoder's causal mask. MAT is therefore executable only when G^t is complete.

4 METHODOLOGY

4.1 MULTI-AGENT COUNTING PROBLEM

To formalize the failure of standard decentralized execution in strict coordination, consider a cooperative multi-agent Markov game with a team of N interchangeable agents choosing between a coordinating action $a^\bullet$ and a default a°. Letting the global coordination count be $X = \sum_{i \in I} \mathbf{1}[a_i = a^\bullet]$, standard CTDE frameworks are bound by three structural conditions: **(1) Count-only reward:** the shared reward $r(X)$ depends exclusively on X and is strictly maximized at an interior target K ($1 < K < N$); **(2) Symmetric observation:** all agents receive an identical observation or broadcast $o_1 = \cdots = o_N \equiv o$ at coordination states; and **(3) Shared policy:** agents act independently through a single parameter set θ, so the joint policy factorizes as $\bar{\pi}_\theta(\bar{a} \mid o) = \prod_{i \in I} \pi_\theta(a_i \mid o)$.

Under these conditions, the global count X reduces to a Binomial distribution $X \sim \mathrm{Binomial}(N, p)$, where $p = \pi_\theta(a^\bullet \mid o)$ is the shared probability of taking the coordinating action. The expected global reward is evaluated directly against this Binomial law as $\mathbb{E}[r \mid p] = \sum_{k=0}^{N} r(k)\binom{N}{k} p^k (1-p)^{N-k}$. Consequently, the probability of achieving exactly the target count K peaks at $p = K/N$, yielding the strict upper bound $\max_{p \in [0,1]} \mathbb{P}(X = K) = \binom{N}{K}(K/N)^K (1 - K/N)^{N-K}$.

Theorem 4.1. *Under Conditions 1–3, every shared policy incurs a strictly positive optimality gap:* $\sup_{p \in [0,1]} \mathbb{E}[r \mid p] < r(K)$. *In particular, a team constrained by such a shared policy can never perfectly coordinate, yielding its maximum expected return strictly below* $r(K)$.

Theorem 4.2. *Under Conditions 1–3, every maximizer* $p^\dagger$ *of* $\mathbb{E}[r \mid p]$ *over* $p \in [0, 1]$ *is a suboptimal shared policy of one of two types:* ***(i) A deterministic policy*** *(*$p^\dagger \in \{0, 1\}$*), if at least one boundary case yields a return strictly greater than the expectation over all* $p \in (0, 1)$ *due to severe miscoordination penalties (e.g.,* $r(0) > \mathbb{E}[r \mid p]$*);* ***(ii) A stochastic policy*** *(*$p^\dagger \in (0, 1)$*), if no such dominating boundary exists, with* $p^\dagger = K/N$ *as the unique maximizer under uniform miscoordination penalties (*$r(k) = c$ *for all* $k \neq K$*).*

To address the limitations established in Theorem 4.1 and Theorem 4.2, a common approach is to assign unique identifiers to agents (Yu et al., 2022; Terry et al., 2020). This breaks the observation symmetry in Condition 2. Such methods can solve the MAC problem for a fixed roster, but, as we show next, their solutions do not transfer when the roster changes. Let s denote a coordination state, and $H_i(s)$ denote agent i's complete policy input, which now includes its identifier. Let $M(s) \subseteq I$ represent the active sub-team that must coordinate to reach a target count K. Agents can form distinct probability distributions $p_i = \pi_\theta(a^\bullet \mid H_i(s))$ with identifiers, taking different roles to solve the problem. However, assuming conditionally independent action selection, the variance of the total action count $X_{M(s)} = \sum_{i \in M(s)} \mathbf{1}[a_i = a^\bullet]$ is given by $\mathrm{Var}(X_{M(s)} \mid s) = \sum_{i \in M(s)} p_i(1 - p_i)$. To solve the problem with 100% success, this variance must be exactly zero, which is only possible if every probability p_i is 0 or 1, taking a deterministic binary action $b_i \in \{0, 1\}$. Exact

success therefore requires a deterministic ID-to-action mapping, which is optimal for a fixed roster but roster-bound: it need not transfer when agents drop out or the team is partitioned, a setting we formalize as our fourth condition: **(4) Membership-blind inputs:** A finite collection of reachable states $s_1, \ldots, s_L$ satisfies $H_i(s_u) = H_i(s_v)$ whenever $i \in M(s_u) \cap M(s_v)$. Unlike Condition 2, this compares the same agent across configurations, allowing different agents to have distinct inputs.

Theorem 4.3. *Under Condition 4 and conditionally independent execution, if the system of equations $\sum_{i \in M(s_u)} b_i = K$ for $u = 1, \ldots, L$ has no valid integer solution $\bar{b} \in \{0,1\}^N$, no decentralized policy can guarantee the exact target at every s_u, regardless of identifiers or memory capacity.*

The formal proofs for theorems, including the exact derivations of the expected reward bounds and convergence dynamics, are deferred to Appendix Section B and Section C.

4.2 The MAC Diagnostic Suite

To empirically evaluate the limitations proven in Section 4.1, we introduce the MAC diagnostic suite of three tasks, visualized in Figure D.1 and fully specified in Section D. In **Aerial Ordering**, a heterogeneous drone team navigates to retrieve distributed targets, each demanding a strict agent count: sensor drones observe target requirements, and worker drones execute the coordinated retrieval. In **Charging Rendezvous**, agents navigate to a central charging station and must coordinate under intermittent communication so that all recharge before battery depletion without ever exceeding its port capacity. In **Team Partition**, the multinomial instantiation of our theoretical framework, agents must simultaneously satisfy exact, zone-specific quotas, leaving any surplus agents idle.

4.3 Semi-Decentralized Multi-Agent Transformer

To address the MAC problem, we propose the Semi-Decentralized Multi-Agent Transformer (SDec-MAT). We build on MAT (Wen et al., 2022), whose transformer natively handles variable input sizes and thus the changing neighborhoods produced by communication loss, but replace its assumption of fully centralized execution with the live communication graph.

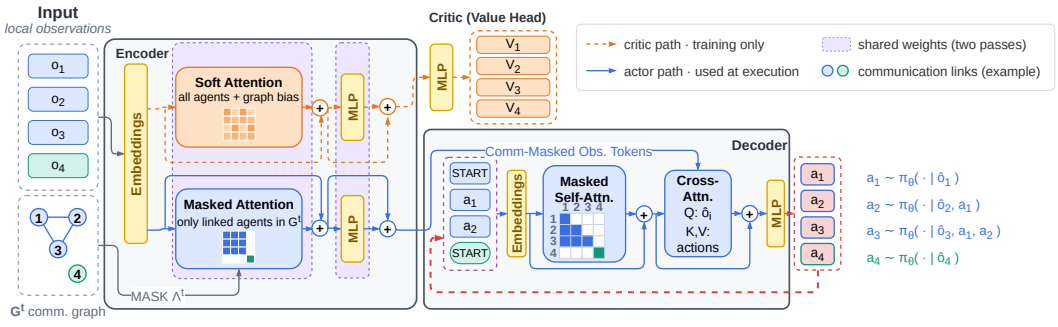

Figure 1: **SDec-MAT architecture** at one decision epoch t. Shared encoder weights support a communication-masked actor (blue) and a global, graph-biased critic (orange). The decoder conditions each action on accessible predecessors; an isolated agent receives its own START token.

Communication graph and decoding order. At decision epoch t, agent i can receive information from the agents in its one-hop neighborhood $\mathcal{N}^t(i)$ of the communication graph G^t, with $i \in \mathcal{N}^t(i)$; we write Λ^t for the adjacency matrix of G^t with self-loops, $\Lambda^t_{ij} = \mathbf{1}[j \in \mathcal{N}^t(i)]$, which is symmetric in all our experiments. Agents are decoded in index order, and agent i's accessible predecessors are $\mathcal{P}^t(i) = \{j \in \mathcal{N}^t(i) : j < i\}$; its position in the decoding chain thus changes with its neighborhood, within and across episodes, while the relative order of two connected agents is fixed. The order is never permuted, in training or execution, so agents need not agree on it at run time; it only decides which of two connected agents acts first and is never an input to the network, which receives no agent identifiers or positional encodings.

Encoder. SDec-MAT runs two passes of a shared encoder (Figure 1). The actor pass, used at execution, embeds each observation o^t_i as z_i and applies masked self-attention, $\alpha_{ij} \propto \exp(e_{ij})\, \Lambda^t_{ij}$

with $e_{ij} = \langle W_q z_i, W_k z_j \rangle / \sqrt{d}$ for learned projections W_q, W_k and embedding width d, so that the encoding $\hat{o}_i^t$ aggregates observations only from $\mathcal{N}^t(i)$; a single encoder block keeps this aggregation within one hop. The critic pass, used only in training, adds to each z_i an embedding of agent i's view of the global state and attends over all agents, replacing the mask with a learned additive bias $\eta\, \Lambda_{ij}^t$ on the scores (η initialized to zero); a value head maps each output to V_i^t.

Decoder. The decoder conditions agent i's action only on $\hat{o}_i^t$ and the realized actions of $\mathcal{P}^t(i)$. Masking attention alone is insufficient, because MAT's right-shifted action tokens would pass a_{i-1} to agent i even when the two are not connected; SDec-MAT therefore builds each agent's action tokens from its own neighborhood, starting from a dedicated START token when no neighbor precedes it, and restricts self-attention to these tokens. We use a single decoder block, whose cross-attention query and residual are $\hat{o}_i^t$, so that no information reaches agent i from beyond one hop. The masks and token construction add no learnable parameters.

Joint policy. The resulting joint policy follows the live communication graph:

$$\bar{\pi}_\theta(\bar{a}^t \mid \bar{o}^t, \Lambda^t) = \prod_{i \in I} \pi_\theta\Big(a_i^t \mid \bar{o}^t_{\mathcal{N}^t(i)}, \bar{a}^t_{\mathcal{P}^t(i)}\Big). \tag{1}$$

When G^t is complete, Equation (1) reduces to MAT's chain-rule factorization; otherwise, it deliberately restricts each factor to the observations and actions agent i can receive. In both cases, unlike Condition 3, it is not an independence assumption: each factor conditions on the realized actions of accessible predecessors, so agents with identical observations can act differently without identifier inputs, and Theorem 4.1 no longer applies. In particular, agent i can compute the number of accessible predecessors that took the coordinating action, $x_i^t = \sum_{j \in \mathcal{P}^t(i)} \mathbf{1}[a_j^t = a^\bullet]$, a quantity that depends on their actions rather than on agent identities, and acting exactly when $x_i^t < K$ attains the target (Proposition C.1) whenever i's preceding neighbors are exactly its active predecessors, for example when the active sub-team is fully connected and disconnected from other agents. Otherwise, x_i^t can miscount, either because an active predecessor acts unseen or because a neighbor outside the sub-team acts. SDec-MAT thus supplies the information that exact counting requires to the extent that the live communication graph makes it available.

Training and execution. All parameters θ are trained jointly with PPO (Schulman et al., 2017). With the probability ratio $\rho_i^t = \pi_\theta(a_i^t \mid \cdot)/\pi_{\theta_{\text{old}}}(a_i^t \mid \cdot)$ of agent i's factor in Equation (1), we minimize

$$\mathcal{L}(\theta) = -\,\mathbb{E}_{i,t}\big[\min\big(\rho_i^t\, \mathrm{Adv}_i^t,\ \mathrm{clip}(\rho_i^t, 1-\epsilon, 1+\epsilon)\, \mathrm{Adv}_i^t\big) + \beta_H\, \mathcal{H}_i^t\big] + \beta_V\, \mathcal{L}_V, \tag{2}$$

where ϵ is the clipping range, Adv_i^t is the generalized advantage estimate (Schulman et al., 2015) of the shared reward computed from agent i's value V_i^t, $\mathcal{H}_i^t$ is the entropy of agent i's factor, $\mathcal{L}_V$ is the clipped Huber value loss with value normalization of MAPPO (Yu et al., 2022), and β_H, β_V are weighting coefficients. Since both passes share encoder weights, the value term of Equation (2) shapes the actor's representation, while the actor's inputs remain local. At each decision epoch, agents first exchange observations with their neighbors, and each agent i computes $\hat{o}_i^t$ locally. Decoding then proceeds in rounds: agent i samples its action once it has received the actions of all agents in $\mathcal{P}^t(i)$, and broadcasts it to its neighbors. The number of action rounds therefore equals the number of agents on the longest index-increasing path in G^t: one for an isolated agent, and at most N, the number of sequential decoding steps MAT always requires. G^t is fixed within a decision epoch, and messages on existing links are assumed delivered. In training and evaluation, this protocol is computed by batched forward passes with per-agent masks. Routing global information only to the critic, and to the actor only what this protocol delivers, yields a CTSDE paradigm.

5 Experiments

We evaluate SDec-MAT against CTDE baselines (MAPPO, QMIX (Rashid et al., 2020), CommNet (Sukhbaatar et al., 2016), TarMAC (Das et al., 2019), InforMARL) and MAT-Dec, the decentralized variant of MAT, which shares SDec-MAT's architecture, including its communication-masked encoder and critic, but conditions each agent's action only on its own encoding, thus differing from

SDec-MAT only in predecessor-action conditioning. We progressively assess permutation invariance, zero-shot generalization, and out-of-distribution scalability. First, Pickup Consensus, a zero-navigation environment, validates the symmetry-breaking collapse proven in Theorems 4.1 and 4.2 and isolates the roles of static agent IDs and the autoregressive decoder. Second, the full MAC environments add spatial navigation and intermittent communication; there, all baselines receive static IDs to test generalization under agent dropouts, network partitions, and unseen team sizes. Finally, we verify on standard MARL environments that SDec-MAT's topology-aware design incurs no cost in the fully connected regime. Full settings and hyperparameters are in Appendix Section G.

5.1 Theoretical Purity Test: Pickup Consensus

As an empirical control for the structural limitations established in Theorems 4.1 and 4.2, we create the task Pickup Consensus that strips spatial navigation and environmental noise from Aerial Ordering, leaving a pure, zero-navigation exact-cardinality selection task. Five identical agents have full communication and a complete view of the task, so no information is ever withheld; their observations are bit-identical, and they must decide exactly which $K \in \{2, 3, 4\}$ of them commit to the PICKUP action while the rest wait, with an episode succeeding only when the count is met exactly.

Figure 2 reports the baselines' training success rates with and without static agent IDs. In the ID-free setting of Figure 2(a), standard CTDE baselines fail to reach the optimum, as Theorem 4.1 predicts: for $N = 5$, the highest success rate a shared policy can attain, reached when each agent picks up with probability K/N, is 0.346, 0.346 and 0.410 for $K = 2, 3, 4$, averaging ≈ 0.37. The policy-gradient baselines converge to a sub-optimal stochastic behavior at this ceiling, consistent with case (ii) of Theorem 4.2, while QMIX, acting greedily over its Q-values, collapses to a deterministic no-attempt behavior with zero success. Conversely, SDec-MAT attains perfect success without identifier inputs, conditioning instead on its predecessors' actions. In Figure 2(b), we inject unique agent IDs into the baselines to break Condition 2 of the MAC problem. This lets them solve the task, but only through a fixed mapping from identifiers to actions tied to the training roster. To test this, Table 1 evaluates each method zero-shot under a 1-agent drop, a 2-agent drop, and a random split, where a 10-agent team divides in half to pick up two targets simultaneously. Under these perturbations, baseline success rates drop sharply, whereas SDec-MAT maintains perfect success across all scenarios. This degradation is structural rather than merely out-of-distribution: when agents cannot identify which teammate dropped, Theorem 4.3 implies no identifier-based decentralized policy can guarantee the exact count for $K \in \{2, 3\}$, even if trained on these configurations.

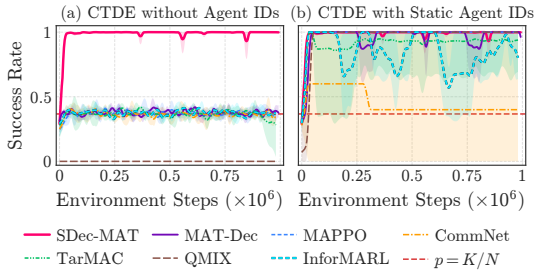

Figure 2: Performance on the Pickup Consensus task. Success rates of CTDE methods compared to SDec-MAT ($N = 5, K \in \{2, 3, 4\}$); the dashed red line marks a shared policy with $p = K/N$.

5.2 Coordination Under Physical Complexity

We next evaluate on the full MAC suite, equipping the baselines with unique static agent IDs to give them a structural foothold to break symmetry and a fair chance to learn, and report evaluation success rates throughout training in Figure 3.

Even with IDs, baselines remain highly inconsistent. InforMARL solves Aerial Ordering and Charging Rendezvous but fails on Team Partition, where a random six of the nine agents are active each episode: the active sub-team $M(s_u)$ changes while each agent's identifier does not, so by Theorem 4.3 no fixed identity-to-action mapping can meet the quotas for every such sub-team. CommNet

Table 1: Zero-shot success rates (mean ± std) of converged policies on Pickup Consensus. Baseline IDs are one-hot over the five training agents, so in Split each ID is reused by two of the ten agents.

Method	**No Drop**	**Drop 1**	**Drop 2**	**Split** (2×5)
MAPPO (w/ ID)	1.000 ± 0.000	0.696 ± 0.003	0.597 ± 0.003	0.360 ± 0.006
QMIX (w/ ID)	1.000 ± 0.000	0.701 ± 0.003	0.602 ± 0.005	0.367 ± 0.007
TarMAC (w/ ID)	1.000 ± 0.000	0.674 ± 0.055	0.549 ± 0.173	0.353 ± 0.015
CommNet (w/ ID)	1.000 ± 0.000	0.550 ± 0.213	0.752 ± 0.053	0.208 ± 0.003
InforMARL (w/ ID)	1.000 ± 0.000	0.697 ± 0.003	0.597 ± 0.004	0.364 ± 0.002
MAT-Dec (w/ ID)	1.000 ± 0.000	0.701 ± 0.002	0.598 ± 0.001	0.363 ± 0.001
SDec-MAT	1.000 ± 0.000	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$

Table 2: Zero-shot concurrent multi-target generalization in Aerial Ordering with T targets and N workers ($T \times N$). Success rate is the fraction of episodes in which all T targets are completed simultaneously; InforMARL receives static agent IDs.

		2×10		3×15	
Comm.	**Split**	**SDec-MAT**	**InforMARL**	**SDec-MAT**	**InforMARL**
Global	Random	$\mathbf{0.946 \pm 0.015}$	0.408 ± 0.027	$\mathbf{0.882 \pm 0.028}$	0.172 ± 0.030
Isolated	Random	$\mathbf{0.968 \pm 0.018}$	0.425 ± 0.038	$\mathbf{0.926 \pm 0.020}$	0.195 ± 0.022
Isolated	Exact	$\mathbf{1.000 \pm 0.000}$	0.691 ± 0.135	$\mathbf{1.000 \pm 0.000}$	0.496 ± 0.143

and TarMAC, whose aggregated messages blur which agents are connected, fail in all three, consistent with the same obstruction. MAT-Dec and QMIX make partial progress in isolated environments, while MAPPO flatlines near zero. SDec-MAT converges to high success in all three.

We evaluate zero-shot multi-target generalization by deploying Aerial Ordering policies trained on a single target (5 workers) into shared environments with 2 targets (10 workers) and 3 targets (15 workers). InforMARL, despite high success on the base task, degrades severely in every multi-target configuration (Table 2). For SDec-MAT, we identify two sources of degradation. First is *communication noise*: under global communication, agents navigating to different targets can enter the same radius, mixing observations and interfering with each other's autoregressive decode chains. Second is *distributional shift*: random target assignments create sub-team sizes differing from the fixed training size. When we filter communication between agents aiming at different targets (Isolated Comm) and keep sub-team sizes identical to training (Exact Split), SDec-MAT's success rate recovers to 1.000 in both 2×10 and 3×15 configurations.

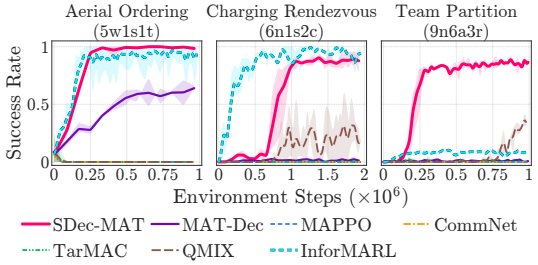

Figure 3: Success rates on the MAC suite; all baselines receive static agent IDs. Aerial Ordering: five workers, one sensor, one target. Charging Rendezvous: six agents, one two-port station. Team Partition: nine agents, six of which split into three regions each episode.

5.3 Joint-Action Coordination in Continuous Navigation

The MAC theorems establish an information-dependent obstruction: under symmetric or membership-blind inputs, decentralized policies cannot represent exact counting. We now ask whether predecessor-action conditioning still helps when no such obstruction exists. We therefore

extend the MPE-based graph navigation task of InforMARL (Nayak et al., 2023; Lowe et al., 2017) into *Joint-Action Navigation*: six agents navigate among three targets, and a target is captured only when *at least three* or *exactly three* agents in range select CAPTURE simultaneously; each failed in-range attempt, a *false capture*, costs -1 or -5. The at-least rule lets in-range agents fire without risk of overshooting, whereas the exact rule demands selective participation whenever more than three gather, and the higher penalty makes failure costlier. An independent solution exists: no method receives IDs, but spatial observations distinguish agents and capture groups are mutually visible, so a common distance ordering can pick three capturers. Penalty-game dynamics nonetheless suggest a hypothesis: a failed attempt penalizes an independent learner for its teammates' exploration, whereas an agent conditioned on its predecessors' realized choices is penalized mainly for its own decision. Within each condition all methods share settings, including optimistic advantage estimation (Zhao et al., 2024) at -5; details, full results and settings are in Appendix Section E.

Table 3: **Joint-Action Navigation.** Success rate (%), false captures per episode (FCap) and training budget (Steps) per condition; Values are mean $\pm$ std over 3 seeds of training-rollout statistics. Full results, including captures, are in Table 5.

	SDec-MAT		InforMARL		MAT-Dec		MAPPO		
Condition	Succ.	FCap	Succ.	FCap	Succ.	FCap	Succ.	FCap	Steps
At least 3, -1	$88.0_{\pm 2.0}$	$0.49_{\pm 0.05}$	$93.0_{\pm 1.0}$	$2.74_{\pm 0.47}$	$93.0_{\pm 0.0}$	$0.67_{\pm 0.41}$	$87.0_{\pm 1.0}$	$4.38_{\pm 0.20}$	2M
At least 3, -5	$88.0_{\pm 1.0}$	$0.79_{\pm 0.21}$	$88.0_{\pm 3.0}$	$1.79_{\pm 0.14}$	$88.0_{\pm 1.0}$	$1.14_{\pm 0.21}$	$0.0_{\pm 0.0}$	$3.55_{\pm 1.62}$	2M
Exactly 3, -1	$85.0_{\pm 8.0}$	$1.80_{\pm 0.89}$	$88.0_{\pm 1.0}$	$5.86_{\pm 0.33}$	$0.0_{\pm 0.0}$	$2.61_{\pm 0.18}$	$27.0_{\pm 39.0}$	$4.96_{\pm 2.19}$	4M
Exactly 3, -5	$93.0_{\pm 1.0}$	$1.07_{\pm 0.08}$	$0.0_{\pm 0.0}$	$1.72_{\pm 0.13}$	$0.0_{\pm 0.0}$	$1.49_{\pm 0.01}$	$0.0_{\pm 0.0}$	$2.18_{\pm 0.21}$	4M

Completion conceals coordination efficiency. Success tracks what each method conditions on Table 3. MAPPO, which does not aggregate observations, succeeds reliably only under the at-least rule with the mild penalty; InforMARL, which aggregates neighbors' observations, withstands either the exact rule or the high penalty, but not both, and MAT-Dec only the high penalty; SDec-MAT succeeds in all four conditions. In every condition it makes the fewest false captures of any method with nonzero success, even where InforMARL succeeds more often: 0.49 versus 2.74 under the at-least rule at -1, and 1.80 versus 5.86 under the exact rule at -1. Success alone thus hides how reliably the count is reached; the exact rule with the high penalty separates methods most: InforMARL and MAT-Dec keep attempting captures with 1.5 to 1.7 failures per episode yet complete at most 0.11 targets, whereas SDec-MAT completes 2.91 and succeeds in 93.0%. Since both baselines succeed under the at-least rule at the same penalty, their failure is specific to exact selection; since MAT-Dec differs from SDec-MAT only in lacking predecessor-action conditioning, the gain is attributable to that conditioning, which therefore matters even when an independent solution exists.

Standard benchmark. We also evaluate SDec-MAT on four SMAC maps (Samvelyan et al., 2019) with distance-based communication graphs. SMAC does not require exact counting, and SDec-MAT remains competitive with MAT-Dec, matching it on 3s5z and 5m_vs_6m and trailing it slightly on MMM and 8m_vs_9m (Appendix Section F).

6 CONCLUSION

We introduced the Multi-Agent Counting diagnostic suite and proved that parameter-shared decentralized policies cannot reliably reach an exact count under symmetric observations. Under membership-blind inputs, no conditionally independent policy, even with identifiers, can guarantee it when the quota constraints are infeasible. SDec-MAT restricts autoregressive decoding to the live communication graph. It attains high success across the MAC suite, generalizes zero-shot to dropouts, partitions and larger multi-target teams far better than identifier-based baselines, and makes the fewest false captures among methods with nonzero success in Joint-Action Navigation. Regarding its limitations, SDec-MAT decodes sequentially within each sub-team, so latency grows with sub-team size, and it assumes the communication graph stays fixed while a joint action is generated. It also obeys every available link, even between agents pursuing unrelated subtasks. Yet simply filtering links between agents bound to different targets already improves its success in Table 2, suggesting that learning which links to use is a promising direction.

AI USE STATEMENT

In this work, we used generative AI tools to refine the formalization of our theoretical framework, to formulate and refine mathematical claims, to fill in and strengthen steps of proofs and assist in their writing, to propose and refine research hypotheses, to design and provide feedback on the research methodology and experiments, and to implement methods (code generation and coding assistance). The core ideas, the overall framework, and the initial theorem statements and proof approaches originated with the authors. We have not used generative AI tools to generate synthetic datasets, clean or reformat data, perform qualitative or thematic data analysis, interpret results, or draft sections of the paper; translation is not applicable to this work. Additionally, we used generative AI tools to polish the writing of the paper for readability, to identify and summarize relevant literature, to format references, and for brainstorming. We have reviewed all AI-assisted work: all theorem statements and proofs were verified by the authors, LLM-generated code was tested for correctness, and LLM-suggested references and summaries were checked against the original sources before citation. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses simulated environments only and involves no human subjects, personal data, or externally sourced datasets; the MAC suite and Joint-Action Navigation are synthetic tasks we constructed, and SMAC is a standard public benchmark. The method addresses coordination in cooperative multi-agent systems under intermittent communication; like other multi-agent control methods, it could in principle be applied to autonomous vehicle or drone teams in both civilian and defense settings. We see no ethical concerns specific to this work beyond those common to multi-agent reinforcement learning research, and we have followed the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

Full proofs of Theorems 4.1 to 4.3 and Proposition C.1, with all assumptions stated, are given in Sections B and C. The complete mechanics of the MAC diagnostic suite, including state and action spaces, communication topologies, and reward functions, are specified in Section D; Joint-Action Navigation is specified in Section E, with its experimental protocol, per-condition settings, and seed sets in Section E.3; the SMAC setup is in Section F. Hyperparameters for all methods are listed in Section G. The architecture is fully described in Section 4.3, including the mask and token construction. Source code for SDec-MAT, all baselines, and all environments is provided in an anonymous repository at `https://github.com/SDec-MAT-ICLR-2027/SDec-MAT-ICLR-2027` as supplementary material and will be released publicly upon acceptance.

AUTHOR CONTRIBUTIONS

Omitted for double-blind review.

ACKNOWLEDGMENTS

Omitted for double-blind review.

REFERENCES

Mahdi Al-Husseini, Mykel J. Kochenderfer, and Kyle H. Wray. A semi-decentralized approach to multiagent control. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2026. doi: 10.65109/ifsk7108.

Dana Angluin. Local and global properties in networks of processors. In *ACM Symposium on Theory of Computing*, pp. 82–93, 1980.

Shushman Choudhury, Jayesh K Gupta, Peter Morales, and Mykel J Kochenderfer. Scalable online planning for multi-agent mdps. *Journal of Artificial Intelligence Research*, 73:821–846, 2022.

Filippos Christianos, Georgios Papoudakis, Muhammad A Rahman, and Stefano V Albrecht. Scaling multi-agent reinforcement learning with selective parameter sharing. In *International Conference on Machine Learning (ICML)*, pp. 1989–1998. PMLR, 2021.

Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI Conference on Artificial Intelligence (AAAI)*, 1998.

Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning (ICML)*, pp. 1538–1546, 2019.

Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Jakob Foerster, Francis Song, Edward Hughes, Neil Burch, Iain Dunning, Shimon Whiteson, Matthew Botvinick, and Michael Bowling. Bayesian action decoder for deep multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 1942–1951, 2019.

Longyue Fu, Jiao Wang, and Hongchen Luo. Multi-agent reinforcement learning for cooperative search under aperiodically intermittent communication. *Expert Systems with Applications*, 280: 127526, 2025.

Claudia V Goldman and Shlomo Zilberstein. Optimizing information exchange in cooperative multiagent systems. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 137–144, 2003.

Carlos Guestrin, Michail Lagoudakis, and Ronald Parr. Coordinated reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 227–234, 2002.

Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 66–83, 2017.

Shengchao Hu, Li Shen, Ya Zhang, and Dacheng Tao. Learning multi-agent communication from graph modeling perspective. In *International Conference on Learning Representations (ICLR)*, 2024.

Siyi Hu, Fengda Zhu, Xiaojun Chang, and Xiaodan Liang. UPDeT: Universal multi-agent RL via policy decoupling with transformers. In *International Conference on Learning Representations (ICLR)*, 2021.

Alon Itai and Michael Rodeh. Symmetry breaking in distributed networks. *Information and Computation*, 88(1):60–87, 1990. ISSN 0890-5401. doi: https://doi.org/10.1016/0890-5401(90)90004-2. URL `https://www.sciencedirect.com/science/article/pii/0890540190900042`.

Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Jiechuan Jiang and Zongqing Lu. The emergence of individuality. In *International Conference on Machine Learning (ICML)*, pp. 4992–5001, 2021.

Kemal Devrim Kafadar, Eren Özaltun, Mahmud Efnan Şanlı, Feyza Orak, Emirhan Gazi, Kubilay Kağan Kömürcü, and Nazım Kemal Üre. Value-aware prediction for robust multi-agent coordination under communication loss. *arXiv e-prints*, pp. arXiv–2607, 2026.

Woojun Kim, Myungsik Cho, and Youngchul Sung. Message-dropout: An efficient training method for multi-agent deep reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 6079–6086, 2019.

Mykel J. Kochenderfer, Tim A. Wheeler, and Kyle H. Wray. *Algorithms for Decision Making*. MIT Press, 2022. URL https://mitpress.mit.edu/9780262047012/algorithms-for-decision-making/.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=EcGGFkNTxdJ.

Chuming Li, Jie Liu, Yinmin Zhang, Yuhong Wei, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. Ace: Cooperative multi-agent q-learning with bidirectional action-dependency. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 8536–8544, 2023.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Siddharth Nayak, Kenneth Choi, Wenqi Ding, Sydney Dolan, Karthik Gopalakrishnan, and Hamsa Balakrishnan. Scalable multi-agent reinforcement learning through intelligent information aggregation. In *International Conference on Machine Learning (ICML)*, pp. 25817–25833, 2023.

Ashutosh Nayyar, Aditya Mahajan, and Demosthenis Teneketzis. Optimal control strategies in delayed sharing information structures. *IEEE Transactions on Automatic Control*, 56(7):1606–1620, 2011.

Yaru Niu, Rohan R Paleja, and Matthew C Gombolay. Multi-agent graph-attention communication and teaming. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 964–973, 2021.

Frans A Oliehoek and Christopher Amato. *A concise introduction to decentralized POMDPs*. Springer, 2016.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

Philipp Robbel, Frans Oliehoek, and Mykel Kochenderfer. Exploiting anonymity in approximate linear programming: Scaling to large multiagent MDPs. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016. doi: 10.1609/aaai.v30i1.10133.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft multi-agent challenge. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 2186–2188, 2019.

Pedro P Santos, Diogo S Carvalho, Miguel Vasco, Alberto Sardinha, Pedro A Santos, Ana Paiva, and Francisco S Melo. Centralized training with hybrid execution in multi-agent reinforcement learning via predictive observation imputation. *Artificial Intelligence*, pp. 104404, 2025.

Christian Schroeder de Witt, Jakob Foerster, Gregory Farquhar, Philip Torr, Wendelin Boehmer, and Shimon Whiteson. Multi-agent common knowledge reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 2085–2087, 2018.

Justin K Terry, Nathaniel Grammel, Sanghyun Son, Benjamin Black, and Aakriti Agrawal. Revisiting parameter sharing in multi-agent deep reinforcement learning. *arXiv preprint arXiv:2005.13625*, 2020.

Rose E. Wang, Michael Everett, and Jonathan P. How. R-MADDPG for partially observable environments and limited communication, 2019. URL `https://openreview.net/forum?id=Ske_NJK2s4`.

Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: multi-agent reinforcement learning with emergent roles. In *International Conference on Machine Learning (ICML)*, 2020.

Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 16509–16521, 2022.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 24611–24624, 2022.

Wenshuai Zhao, Yi Zhao, Zhiyuan Li, Juho Kannala, and Joni Pajarinen. Optimistic multi-agent policy gradient. In *International Conference on Machine Learning (ICML)*, 2024.

A SDec-POMDP Observation Selector

Al-Husseini et al. (2026) propagate information into agent memories and a blackboard through three selector functions, for memories, actions and observations. We use only the observation selector. Given the joint realization $\bar{\tau}$ and the joint observation $\bar{o}$, it records for each agent $i \in I$ and for the blackboard

$$o_i^{\mathrm{sel}} = \begin{cases} \{o_j : j \in I,\ \tau_j = 0\} & \text{if } \tau_i = 0, \\ \{o_i\} & \text{if } \tau_i > 0, \end{cases} \qquad o_{\mathrm{c}}^{\mathrm{sel}} = \{o_j : j \in I,\ \tau_j = 0\}. \tag{3}$$

A communicating agent therefore records the observations of every communicating agent, and a non-communicating agent records only its own. Since the predicate $\tau_j = 0$ ranges over all of I, every communicating agent writes to a single blackboard, although Al-Husseini et al. (2026) note that semi-decentralization admits several distinct communicating sets. Section 4.3 instead restricts each agent i to its one-hop neighborhood $\mathcal{N}^t(i)$, which equals the communicating set containing i for every $i \in I$ exactly when the connected components of G^t are cliques. In that case each component is one communicating set, and when G^t is determined by the state, as with proximity, the event $\tau_i' = 0$ is determined by the next state s' for every $i \in I$, so F assigns it probability zero or one given s'.

B Extended Theoretical Analysis and Proofs of Multi-Agent Counting Problem

To formally establish the limitations of standard decentralized execution, we consider a Dec-POMDP with N interchangeable agents. We isolate a critical coordination state s where each agent i must select an action from a binary set $a_i \in \{a^\bullet, a^\circ\}$, with $a^\bullet$ representing the distinguished coordinating action. The total number of agents executing this action defines the global coordination count, $X = \sum_{i \in I} \mathbf{1}[a_i = a^\bullet] \in \{0, \ldots, N\}$.

We define the fundamental constraints of standard Centralized Training with Decentralized Execution (CTDE) frameworks through three structural conditions:

- **Condition 1 (Count-only reward with an interior peak):** The shared environmental reward $r(X)$ depends exclusively on the global count and is uniquely maximized at a specific, randomly initialized interior target K ($1 < K < N$), such that $r(K) > r(k)$ for all $k \neq K$.
- **Condition 2 (Symmetric observation and communication):** At the coordination state s, the environment provides identical local observations $o_1 = \cdots = o_N \equiv o$. Furthermore, under any permutation-invariant broadcast communication protocol, identical observations cause agents to emit identical messages, resulting in perfectly identical received message profiles and symmetric observations o across all agents.
- **Condition 3 (Symmetric shared policy):** To enable sample efficiency and zero-shot generalization, agents share a single set of policy parameters θ, forcing the joint policy to factorize independently as $\bar{\pi}_\theta(\bar{a} \mid o) = \prod_{i \in I} \pi_\theta(a_i \mid o)$.

Because the shared policy processes identical inputs under these conditions, every agent independently emits the exact same distribution over its action space. This reduces the multi-agent decision-making process to a common probability of choosing the coordinating action, defined as $p := \pi_\theta(a^\bullet \mid o) \in [0, 1]$. Consequently, the global count X behaves as a sum of N independent and identically distributed trials, forming a Binomial distribution: $X \sim \mathrm{Binomial}(N, p)$.

The expected global reward, parameterized by the shared probability p, is the reward averaged against this Binomial law:

$$\mathbb{E}[r \mid p] = \sum_{k=0}^{N} r(k) \binom{N}{k} p^k (1-p)^{N-k}. \tag{4}$$

B.1 The Maximal Probability of Exact Coordination

Before establishing the reward bounds, we define the theoretical ceiling for reliability: the highest attainable probability that exactly K agents coordinate. As a function of p, this probability is $f(p) = \binom{N}{K} p^K (1-p)^{N-K}$. Maximizing the log-likelihood $\ell(p) = \ln f(p)$ yields a unique critical point at $p = K/N$. Thus, the probability of hitting the exact target count is strictly bounded by:

$$\max_{p \in [0,1]} \mathbb{P}(X = K) = \binom{N}{K} \left(\frac{K}{N}\right)^K \left(1 - \frac{K}{N}\right)^{N-K}. \tag{5}$$

Crucially, this reliability bound decays as the team size increases. Fixing a target ratio $\kappa = K/N \in (0, 1)$ and taking the limit as $N \to \infty$, the local de Moivre-Laplace theorem shows that the peak Binomial mass at the mean satisfies:

$$\max_p \mathbb{P}(X = \kappa N) = \mathbb{P}(X = \kappa N \mid p = \kappa) \sim \frac{1}{\sqrt{2\pi\kappa(1-\kappa)N}}. \tag{6}$$

Consequently, the best attainable probability of exact coordination is $\Theta(N^{-1/2})$ and vanishes as $N \to \infty$.

B.2 Proof of Theorem 4.1

Theorem 4.1. *Under Conditions 1–3, every shared policy incurs a strictly positive optimality gap:* $\sup_{p \in [0,1]} \mathbb{E}[r \mid p] < r(K)$. *In particular, a team constrained by such a shared policy can never perfectly coordinate, yielding its maximum expected return strictly below* $r(K)$.

Proof. For any $p \in (0, 1)$, let $P_k := \mathbb{P}(X = k) = \binom{N}{k} p^k (1-p)^{N-k} > 0$. Since $\sum_{k=0}^{N} P_k = 1$, we can decompose the expected reward around the optimum:

$$\mathbb{E}[r \mid p] = \sum_{k=0}^{N} r(k) P_k = r(K) - \sum_{k \neq K} (r(K) - r(k)) P_k. \tag{7}$$

By Condition 1, $r(K) - r(k) > 0$ for every $k \neq K$. Therefore, the weighted sum of penalties is strictly positive, implying $\mathbb{E}[r \mid p] < r(K)$ for all $p \in (0, 1)$. Furthermore, deterministic policies

where $p \in \{0, 1\}$ can only produce counts of $X = 0$ or $X = N$, which Condition 1 also eliminates since $1 < K < N$. Thus, $\mathbb{E}[r \mid p] < r(K)$ holds for all $p \in [0, 1]$. Because $p \mapsto \mathbb{E}[r \mid p]$ is a continuous polynomial on the compact set $[0, 1]$, it attains its supremum, which remains strictly below $r(K)$ by this pointwise bound.

B.3 Optimal Shared Policies and Proof of Theorem 4.2

While Theorem 4.1 bounds the performance of every representable policy, Theorem 4.2 characterizes the best one: the maximizers $p^\dagger$ of the shared objective $\mathcal{J}(p) = \mathbb{E}[r \mid p]$, which policy-gradient methods ascend. It makes no claim about which stationary point a particular learner reaches.

Theorem 4.2. *Under Conditions 1–3, every maximizer $p^\dagger$ of $\mathbb{E}[r \mid p]$ over $p \in [0, 1]$ is a suboptimal shared policy of one of two types:* ***(i) A deterministic policy*** *($p^\dagger \in \{0, 1\}$), if at least one boundary case yields a return strictly greater than the expectation over all $p \in (0, 1)$ due to severe miscoordination penalties (e.g., $r(0) > \mathbb{E}[r \mid p]$);* ***(ii) A stochastic policy*** *($p^\dagger \in (0, 1)$), if no such dominating boundary exists, with $p^\dagger = K/N$ as the unique maximizer under uniform miscoordination penalties ($r(k) = c$ for all $k \neq K$).*

Proof. By Theorem 4.1, $\sup_p \mathbb{E}[r \mid p] < r(K)$, so every maximizer $p^\dagger$ is suboptimal. Because $\mathcal{J}(p) = \mathbb{E}[r \mid p]$ is a polynomial, it is continuous on the compact set $[0, 1]$ and attains its maximum.

For case (i), suppose $r(0) > \mathbb{E}[r \mid p]$ for all $p \in (0, 1)$; the case of $p = 1$, where $\mathbb{E}[r \mid 1] = r(N)$, is symmetric. Since $\mathbb{E}[r \mid 0] = r(0)$, no interior point is a maximizer, so $p^\dagger \in \{0, 1\}$.

For case (ii), if the domination condition of (i) is not met, then for each boundary point some interior point attains at least its return, so $\mathcal{J}$ attains its maximum at some $p^\dagger \in (0, 1)$. Assuming uniform penalties $r(k) = c$ for all $k \neq K$, and letting $P_K = \mathbb{P}(X = K)$, we have:

$$\mathbb{E}[r \mid p] = \sum_{k=0}^{N} r(k) P_k = c \sum_{k \neq K} P_k + r(K) P_K = c + (r(K) - c) P_K. \tag{8}$$

Since $r(K) > c$, this expectation is strictly increasing with respect to P_K. By Equation (5), P_K is uniquely maximized at $p = K/N$. Hence, $p^\dagger = K/N$ is the unique maximizer.

C Proofs for Coordination with Identifiable Agents

We use the notation of Section 4.1: at a coordination state s, $H_i(s)$ is agent i's complete policy input, including its identifier and any memory, $M(s)$ is the active sub-team, and $X_{M(s)} = \sum_{i \in M(s)} \mathbf{1}[a_i = a^\bullet]$. Conditionally independent execution means $\mathbb{P}(\bar{a} \mid s) = \prod_{i \in I} \pi_i(a_i \mid H_i(s))$. The main text uses a shared policy, $\pi_i = \pi_\theta$, but every argument below also holds for distinct policies π_i, so separate parameters do not remove the obstruction either. The input maps are fixed throughout.

C.1 Proof of Theorem 4.3

Theorem 4.3. *Under Condition 4 and conditionally independent execution, if the system of equations $\sum_{i \in M(s_u)} b_i = K$ for $u = 1, \ldots, L$ has no valid integer solution $\bar{b} \in \{0, 1\}^N$, no decentralized policy can guarantee the exact target, regardless of identifiers or memory capacity.*

Proof. Suppose a conditionally independent policy attains the exact target almost surely at every listed state $s_1, \ldots, s_L$. Write $M_u = M(s_u)$ and $p_i(s_u) = \pi_i(a^\bullet \mid H_i(s_u))$. Since $X_{M_u} = K$ almost surely, conditional independence gives

$$0 = \mathrm{Var}(X_{M_u} \mid s_u) = \sum_{i \in M_u} p_i(s_u)\big(1 - p_i(s_u)\big). \tag{9}$$

Every summand is nonnegative, so $p_i(s_u) \in \{0, 1\}$ for every active agent: exact attainment forces deterministic decisions on the active agents' inputs.

By Condition 4, $H_i(s_u)$ is identical across all listed states in which agent i is active, so its coordinating-action probability takes a single binary value across those states; denote it by b_i. For agents never active, choose b_i arbitrarily. Exact attainment then requires

$$\sum_{i \in M_u} b_i = K, \qquad u = 1, \ldots, L, \qquad \bar{b} \in \{0,1\}^N, \tag{10}$$

which contradicts the assumed infeasibility of the system.

The argument allows different decisions for different agents. Identifiers therefore do not remove the obstruction: each agent must make a consistent decision across states its complete input cannot distinguish. Independent randomization may improve imperfect success rates, but cannot achieve almost-sure exactness on an infeasible collection.

Example. Let any one of three agents drop out with $K = 1$, and suppose the remaining agents cannot tell which teammate left. The active sub-teams $\{1,2\}$, $\{1,3\}$ and $\{2,3\}$ require $b_1 + b_2 = b_1 + b_3 = b_2 + b_3 = 1$; summing gives $2(b_1 + b_2 + b_3) = 3$, which has no integer solution.

C.2 Sequential Counting

Proposition C.1 (Sequential counting)**.** *Suppose agents act in a fixed order, the active sub-team $M(s)$ and target $K \leq |M(s)|$ remain fixed while the joint action is generated, and each active agent knows its own membership, the target K, and the number of active predecessors that selected $a^\bullet$. Then the rule that selects $a^\bullet$ exactly when the agent is active and this number is below K attains $X_{M(s)} = K$ deterministically.*

Proof. Let ξ_n denote the selected count after n active agents have acted. The rule gives

$$\xi_0 = 0, \qquad \xi_{n+1} = \xi_n + \mathbf{1}[\xi_n < K], \tag{11}$$

and induction yields $\xi_n = \min(n, K)$. After all active agents have acted,

$$X_{M(s)} = \xi_{|M(s)|} = \min(|M(s)|, K) = K.$$

The rule is not conditionally independent, since each decision depends on realized predecessor actions. Together with Theorem 4.3, it separates sequential from decentralized execution whenever sequential execution supplies the required information. The proposition does not establish that a particular decoder recovers these quantities, that training converges to the rule, or that the predecessor count is available when some active predecessors lie outside an agent's communication neighborhood, the case discussed in Section 4.3.

D Full Mechanics of the MAC Diagnostic Suite

In Section 5.2, we evaluate standard baseline methods on the MAC diagnostic suite. The environments force exact, quota-based coordination but isolate distinct axes of physical complexity. The conceptual mechanics of each environment, along with their exact state-action spaces, communication topologies, and reward formulations, are detailed below. Visual representations are provided in Figure D.1, and a summary of key hyperparameters is presented in Table 4.

D.1 Aerial Ordering

Overview & Dynamics: A sensor agent and $N - 1$ workers must navigate to a target with a hidden lifting requirement K, which workers discover only upon entering the sensor's communication radius. Exactly K workers must then pick up the target simultaneously. In its multi-target variant, the team must additionally decompose into independent sub-tasks while ignoring interfering communication from agents bound to other targets. Incidental physical structure (e.g., staggered arrivals or wind drift) can in principle leak asymmetric information into local histories, giving parameter-shared policies a potential foothold for symmetry breaking.



(a) Aerial Ordering (b) Charging Rendezvous (c) Team Partition

Figure D.1: Visualizations of the MAC diagnostic suite environments.

Table 4: Key environment configurations and dimensions for the MAC suite.

Parameter	**Aerial Ordering**	**Charging Rendezvous**	**Team Partition**
Total Agents (N)	6 (Heterogeneous)	6 (Uniform)	9 (Uniform)
Target/Capacity	$K \sim \mathcal{U}\{2,3,4\}$	$C = 2$	$\sum_{q=1}^{Q} K_q = 6, Q = 3$
Grid Dimensions	10×10	10×10	9×6
Horizon	100	100	20
Comm. Radius (r_{com})	0 (Sensor cell)	1 (Manhattan)	2 (Manhattan)
State dim	24	46	22
Obs. dim	5	5	14
Actions ($\lvert A_i \rvert$)	3	4	5
Stochasticity	Disp. $p_{\text{disp}} = 0.1$	None	None

State and Action Space: The global state s includes the normalized spatial coordinates of all agents and the target, the hidden quota K, network connectivity metrics, and task completion status. Local observations o_i contain proximity indicators, perceived K (only if connected), and connected worker counts. The discrete action space is $A_i = \{\text{MOVE}, \text{PICKUP}, \text{WAIT}\}$. The PICKUP action is heavily masked and strictly reserved for unloaded workers physically sharing the target cell with the sensor.

Communication Topology: The network forms a dynamic star topology centered on the sensor agent. A worker i connects to the sensor if their Manhattan distance is $\text{dist}(i, \text{sensor}) \leq r_{\text{com}} = 0$. The adjacency matrix Λ^t forms a localized clique containing the sensor and connected workers, while disconnected workers act as isolated singletons. Workers not attempting a pickup are subject to a stochastic spatial displacement with probability $p_{\text{disp}} = 0.1$, dynamically severing communication links.

Reward Function: The joint reward encourages minimal steps and penalizes exact-coordination failures:

$$R = \begin{cases} +20, & \text{if exactly } K \text{ workers successfully lift target} \\ -20, & \text{if premature, overshoot, or undershoot lift attempted} \\ -0.1 - 0.2 \cdot \min(\frac{\delta}{2(g-1)}, 1), & \text{standard step penalty based on mean distance } \delta \end{cases} \tag{12}$$

D.2 Charging Rendezvous

Overview & Dynamics: N agents must share a central station with exactly C ports. Up to C requests can dock, but any excess triggers a brownout that rejects all agents. Agents are dispatched from varying distances yet initialized with batteries proportional to their trip length, meaning they arrive with identical residual charge. This removes battery level as a sorting key, forcing a symmetric tie that must be broken collectively rather than by prioritizing the neediest agent.

State and Action Space: The state space monitors global agent coordinates, individual residual batteries, and station port utilization. Agents spawn in two staggered cohorts but are initialized with proportional residual charge, explicitly removing energy levels as a symmetry-breaking mechanism. Actions are $A_i = \{\text{CONNECT}, \text{WAIT}, \text{DISCONNECT}, \text{MOVE}\}$.

Communication Topology: Agents establish connections strictly at the charging hub ($r_{\text{com}} = 1$). The active communication graph consists of a single clique comprising unsatisfied agents within the hub's vicinity.

Reward Function: The critical challenge is the capacity constraint $C = 2$. If the number of simultaneous CONNECT requests N_{req} exceeds C, a capacity violation occurs.

$$R = \begin{cases} +100, & \text{if all agents reach satisfied state} \\ -100, & \text{if any agent's battery depletes to } \leq 0 \\ -5 \cdot (N_{\text{req}} - C), & \text{if } N_{\text{req}} > C \text{ (capacity violation)} \\ -0.5 \cdot C_{\text{free}}, & \text{if ports are idle while agents are waiting} \\ +10, & \text{per agent successfully satisfied} \end{cases} \tag{13}$$

D.3 Team Partition

Overview & Dynamics: N agents scattered in two arrival waves around a central staging cell must regroup and split across Q color-coded zones so that the realized counts match a public quota $\mathbf{K}$ exactly. Partial matches earn per-zone credit, and only an exact match pays the completion bonus. Departures are inert until a quorum of agents stands on the cell, and the gate they open must be used within a single round and before a deadline that only a team in which everyone rushes can meet. This ensures the opportunity is a narrow, environment-timed window rather than a state the team can settle into and deliberate from. Because the quota sums to fewer than N, the surplus agents are genuinely spare. At the decision point, every present agent observes the exact same thing (the public quota and its own arrival bit, with no identifier). Consequently, a memoryless parameter-shared policy is provably capped at the multinomial ceiling $\max_{\mathbf{p}} \mathbb{P}(\mathbf{X} = \mathbf{K})$.

State and Action Space: The environment comprises 9 agents and $Q = 3$ zones with quotas summing to exactly $\sum_q K_q = 6$, leaving 3 agents redundant. Local observations o_i contain the public quota vector and an arrival bit, with no agent identifiers. Actions are $A_i = \{\text{STAY}, \text{GO}_1, \text{GO}_2, \text{GO}_3, \text{MOVE}\}$.

Communication Topology: A localized radio disk exists around the central staging cell ($r_{\text{com}} = 2$). The adjacency matrix connects agents strictly when a quorum of at least two agents occupies the staging zone, isolating distant agents.

Reward Function: Let X_q denote the realized agent count at zone q. The reward strongly incentivizes exact vector matching for the public quota $\mathbf{K}$:

$$R = 10 \cdot \sum_{q=1}^{Q} \mathbf{1}[X_q = K_q] + R_{\text{bonus}} \cdot \mathbf{1}[\mathbf{X} = \mathbf{K}] - 20 \cdot \mathbf{1}[\mathbf{X} \neq \mathbf{K}] \tag{14}$$

where R_{bonus} is the completion scalar. Missing the tight commit window results in a catastrophic truncation penalty of -25.

E Joint-Action Navigation Details

E.1 Joint-Action Navigation

Overview & Dynamics: $N = 6$ agents share a 2×2 continuous arena with three shared targets (Figure E.1). A target is captured only when agents within its capture radius ($r_{\text{cap}} = 0.3$) select CAPTURE on the same step, either at least $K = 3$ of them (at-least-K) or exactly K (exact-K, where extra firers destroy the capture). Agents are double integrators, and CAPTURE applies no movement force, so firing has a real cost. Under at-least-K, all six agents can pile onto one target; under exact-K, the group must split into three firers and three holders, which a single shared intent

cannot achieve. Captured targets are deactivated, and the episode ends when all targets are captured or after 50 steps.

State and Action Space: The global state contains agent positions and velocities, target positions and active flags, and the communication graph (critic only). Each agent observes its own position and velocity and the relative positions of agents and targets within $r_{\text{com}} = 0.6$, with no agent identifier. Actions are $A_i = \{\text{NoOp}, \text{Left}, \text{Right}, \text{Down}, \text{Up}, \text{Capture}\}$.

Communication Topology: Agents within r_{com} of each other are connected, using the same radius as sensing. The graph is recomputed every step and strictly one-hop, ranging from six singletons to a full clique. Since $r_{\text{com}} \geq 2r_{\text{cap}}$, agents at the same target always see each other, so an independent solution is informationally feasible; this task tests learning rather than the MAC impossibility.

Reward Function: The shared reward is broadcast identically to all agents, so it carries no per-agent credit.

$$R = \begin{cases} +10, & \text{per successful capture} \\ c_{\text{fail}}, & \text{if fewer than } K \text{ agents fire (undershoot)} \\ c_{\text{fail}}, & \text{if more than } K \text{ agents fire (overkill, exact-}K\text{ only)} \\ \Phi(s') - \Phi(s), & \text{potential-based shaping} \end{cases} \tag{15}$$

The potential Φ rewards approaching the nearest active target and assembling around it, while the capture itself stays unshaped. No step or collision cost is applied.

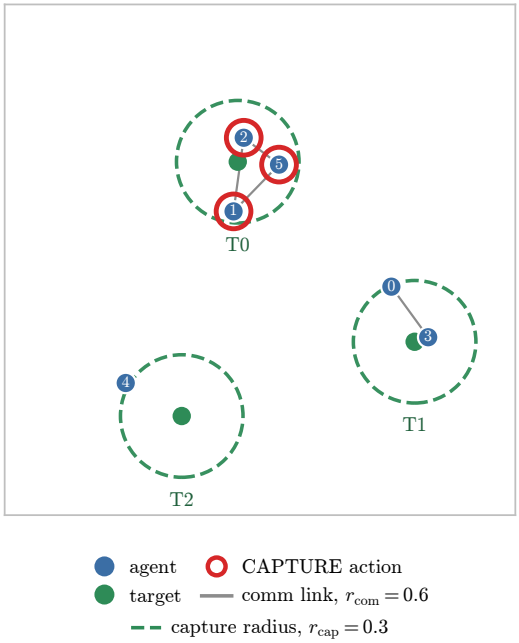

Figure E.1: Joint-Action Navigation. Six agents navigate among three targets; a target is captured only when at least three, or exactly three, agents within its capture radius select CAPTURE simultaneously.

E.2 Full Results

Table 5 reports all metrics. Under the at-least rule, SDec-MAT and MAT-Dec capture nearly the same number of targets, so the difference between them lies in false captures rather than completion, most clearly at penalty -5. Under exactly-three with penalty -1, SDec-MAT and InforMARL substantially outperform MAT-Dec and MAPPO, and SDec-MAT matches InforMARL's captures with less than a third of its false captures, although InforMARL retains the highest success rate. Under exactly-three with penalty -5, InforMARL and MAT-Dec complete about 0.1 targets per episode.

InforMARL's failure in the exact, high-penalty condition occurs despite its success on Aerial Ordering and Charging Rendezvous (Section 5.2) and despite mutual visibility within capture groups. The distinction is therefore between information sufficient to define a coordination rule and learning to use that information: MAC establishes a representational limitation under specified assumptions, whereas navigation exposes an empirical learning gap where an independent solution remains available. Because the results are training-time measurements and budgets differ across conditions, differences between rows cannot be attributed to the capture rule or penalty alone, and the comparisons support a benefit under the tested training procedure rather than a general convergence guarantee.

Table 5: **Joint-Action Navigation, full results.** Success: percentage of episodes completing all targets. Captures: mean targets completed per episode (maximum 3). False captures: mean unsuccessful in-range capture attempts per episode.

Task	Metric	SDec-MAT	InforMARL	MAT-Dec	MAPPO	Steps
At least 3 Penalty -1	Success (%)	88.0$\pm$2.0	93.0$\pm$1.0	93.0$\pm$0.0	87.0$\pm$1.0	2M
	Captures	2.86$\pm$0.02	2.88$\pm$0.00	2.92$\pm$0.01	2.78$\pm$0.04	
	False captures	0.49$\pm$0.05	2.74$\pm$0.47	0.67$\pm$0.41	4.38$\pm$0.20	
At least 3 Penalty -5	Success (%)	88.0$\pm$1.0	88.0$\pm$3.0	88.0$\pm$1.0	0.0$\pm$0.0	2M
	Captures	2.87$\pm$0.02	2.87$\pm$0.03	2.86$\pm$0.02	0.72$\pm$0.43	
	False captures	0.79$\pm$0.21	1.79$\pm$0.14	1.14$\pm$0.21	3.55$\pm$1.62	
Exactly 3 Penalty -1	Success (%)	85.0$\pm$8.0	88.0$\pm$1.0	0.0$\pm$0.0	27.0$\pm$39.0	4M
	Captures	2.81$\pm$0.10	2.83$\pm$0.03	0.12$\pm$0.01	1.00$\pm$1.23	
	False captures	1.80$\pm$0.89	5.86$\pm$0.33	2.61$\pm$0.18	4.96$\pm$2.19	
Exactly 3 Penalty -5	Success (%)	93.0$\pm$1.0	0.0$\pm$0.0	0.0$\pm$0.0	0.0$\pm$0.0	4M
	Captures	2.91$\pm$0.03	0.11$\pm$0.01	0.09$\pm$0.01	0.13$\pm$0.04	
	False captures	1.07$\pm$0.08	1.72$\pm$0.13	1.49$\pm$0.01	2.18$\pm$0.21	

E.3 Joint-Action Navigation: Training Protocol

Shared settings. SDec-MAT and MAT-Dec share the same critic and differ only in the actor; MAPPO and InforMARL use their standard centralized critics, and all critics access the global state. The environment, reward, PPO hyperparameters (Table 9), exploration, off-policy weighting, optimistic advantages, step budget, rollout workers and three seeds are identical across all four methods; we extended InforMARL's training loop so that both code bases share this configuration. Reported values are training-rollout statistics, with exploration active for all methods.

Exploration. Since exact joint actions are rarely reached by independent per-agent exploration, all methods use the same group exploration: with probability p_{explore}, a random action is drawn for a communication group, and each member adopts it independently with probability p_{adopt}, otherwise following its own policy. This exploration is random, affects only the behavior policy, and, like the off-policy weighting of exploratory transitions used for stability, is identical for every method. It does not solve the coordination problem: under exactly-three at -5, every baseline fails with the same exploration.

Condition-dependent settings. Two settings depend on the task and apply to all methods (Table 6): a lower adoption probability p_{adopt} under the exact rule, which requires partial rather than whole-group participation, and optimistic advantages (Zhao et al., 2024) at the -5 penalty, where exploration otherwise yields almost uniformly negative advantages.

F SMAC Results

Setup. We evaluate on the SMAC maps MMM, 3s5z, 5m_vs_6m and 8m_vs_9m (Samvelyan et al., 2019). SMAC does not require exact counting, so this experiment tests whether predecessor-action conditioning hurts on a standard cooperative benchmark. We compare SDec-MAT with MAT-Dec, which differs from it only in predecessor-action conditioning (Section 5).

Table 6: Joint-Action Navigation settings, identical for all four methods. Shared across conditions: $N = 6$ agents, 3 targets, $r_{\text{cap}} = 0.3$, $r_{\text{com}} = 0.6$, horizon 50, 32 rollout workers, 3 seeds; PPO hyperparameters are in Table 9.

Condition	p_{explore}	p_{adopt}	Optimism	Steps
At least 3, -1	0.2	1.0	–	2M
At least 3, -5	0.2	1.0	✓	2M
Exactly 3, -1	0.2	0.5	–	4M
Exactly 3, -5	0.2	0.5	✓	4M

Communication graph. Agents i and j are connected at epoch t when their Euclidean distance is at most $r_{\text{com}} = 3$, shorter than the SMAC unit sight range of 9. G^t is recomputed at every epoch, so, as in the MAC suite, it changes as units move, and the joint action is decoded only within connected sub-teams. Dead agents are removed from G^t.

Inputs. Neither method receives agent identifiers. Both use the standard SMAC local observations, and the critic additionally receives the standard SMAC global state.

Training and evaluation. Both methods share a single configuration on all maps, adapted from the SMAC settings of Wen et al. (2022): learning rate 5×10^{-4}, 15 PPO epochs, clip 0.05, one minibatch, entropy coefficient 0.01 and 20 environments $\times$ 100 steps per update, without the group exploration, off-policy weighting or optimistic advantages used in Joint-Action Navigation. Both are trained for 10×10^6 environment steps with 3 seeds. Every 25 training iterations, we run 32 evaluation episodes with greedy actions and record the win rate. Figure F.1 reports the mean $\pm$ std over seeds.

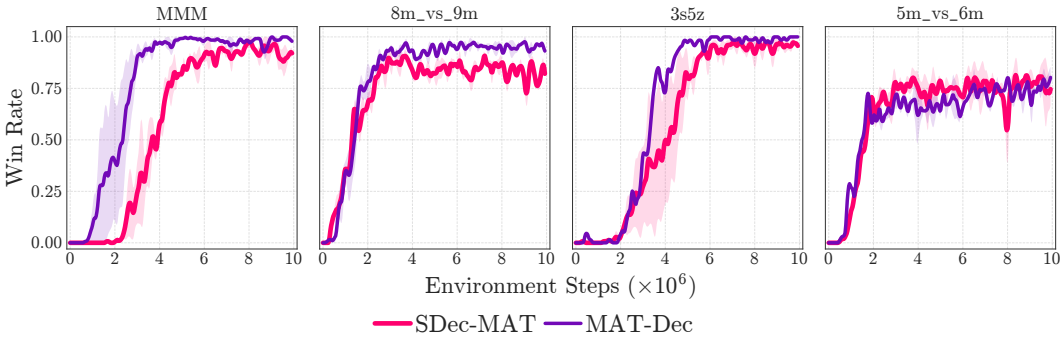

Figure F.1: Evaluation win rate of SDec-MAT and MAT-Dec on four SMAC maps with distance-based communication graphs ($r_{\text{com}} = 3$; mean $\pm$ std over 3 seeds).

Results. SDec-MAT reaches final win rates comparable to MAT-Dec on 3s5z and 5m_vs_6m, and slightly lower ones on MMM and 8m_vs_9m. MAT-Dec also learns faster on MMM and 3s5z. SMAC's heterogeneous units observe different positions, health and types, so independent agents can take different roles without predecessor actions, and predecessor-action conditioning offers no structural advantage here; its extra input may instead slow learning. SDec-MAT therefore remains competitive on a standard benchmark while providing the exact-coordination gains of Section 5.

G Experimental Details and Hyperparameters

In this section, we provide the hyperparameters used to train SDec-MAT and the baselines. For the MAC suite, Table 7 details the on-policy algorithms and Table 8 QMIX. Joint-Action Navigation uses a separate configuration shared by all four methods and listed in Table 9, with task settings in Table 6.

Table 7: Hyperparameters for On-Policy Algorithms.

Hyperparameter	**MAPPO / CommNet / TarMAC**	**InforMARL**	**MAT-Dec / SDec-MAT**
Learning Rate	3×10^{-4}	3×10^{-4}	3×10^{-4}
Discount Factor (γ)	0.99	0.99	0.99
PPO Epochs	4	4	4
PPO Clip (ϵ)	0.2	0.2	0.2
Entropy Coefficient (β_H)	0.1	0.1	0.1
Advantage Estimate	5-step returns	GAE ($\lambda = 0.95$)	GAE ($\lambda = 0.95$)
Max Gradient Norm	0.5	0.5	10.0
Batch Size	10 episodes	10 envs $\times$ 10 steps	32 envs $\times$ 10 steps
Actor Architecture	GRU (128 hidden)	GNN (2 layers, 3 heads)	Transformer (1 block, 1 head, 64 dim)
Critic Architecture	Centralized $V(s)$	Global Graph Aggregation	Transformer Encoder (graph-biased)

Table 8: Hyperparameters for QMIX.

Hyperparameter	**Value**
Learning Rate	3×10^{-4}
Replay Buffer Capacity	5,000 episodes
Batch Size	32 episodes
Target Network Update	Hard update every 200 episodes
Exploration (ϵ-greedy)	$1.0 \rightarrow 0.05$ over 50k steps
Agent Architecture	GRU (128 hidden)
Mixer Architecture	Hypernetwork (2 layers, 64 hidden, 32 embed)

Table 9: Hyperparameters for Joint-Action Navigation, identical across all conditions and seeds.

Hyperparameter	**MAPPO**	**InforMARL**	**MAT-Dec / SDec-MAT**
Learning Rate (actor and critic)	5×10^{-4}	5×10^{-4}	5×10^{-4}
Optimizer	Adam ($\epsilon_{\text{Adam}} = 10^{-5}$)	Adam ($\epsilon_{\text{Adam}} = 10^{-5}$)	Adam ($\epsilon_{\text{Adam}} = 10^{-5}$)
Discount Factor (γ)	0.99	0.99	0.99
PPO Epochs	5	5	5
Minibatches	1	1	1
PPO Clip (ϵ)	0.2	0.2	0.2
Entropy Coefficient (β_H)	0.01	0.01	0.01
Value Loss Coefficient (β_V)	1	1	1
Advantage Estimate	GAE ($\lambda = 0.95$)	GAE ($\lambda = 0.95$)	GAE ($\lambda = 0.95$)
Value Loss	Clipped Huber (threshold 10)	Clipped Huber (threshold 10)	Clipped Huber (threshold 10)
Value Normalization	Yes	Yes	Yes
Max Gradient Norm	10.0	10.0	10.0
Batch Size	32 envs $\times$ 50 steps	32 envs $\times$ 50 steps	32 envs $\times$ 50 steps
Actor Architecture	MLP (2 layers, 64 hidden)	GNN (2 layers, 3 heads)	Transformer (1 block, 1 head, 64 dim)
Critic Architecture	Centralized $V(s)$	Global Graph Aggregation	Transformer Encoder (graph-biased)