BRIDGING MARL TO SARL: PARALLEL LATENT CONSENSUS FOR CENTRALIZED MULTI-AGENT CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Cooperative multi-agent reinforcement learning (MARL) is widely used to address large joint observation and action spaces by decomposing a centralized control problem into multiple interacting agents. However, such decomposition often introduces additional challenges, including non-stationarity, unstable training, weak coordination, and limited theoretical guarantees. In this paper, we propose the Consensus Multi-Agent Transformer (CMAT), a centralized framework that reformulates fully cooperative, fully observable MARL as a hierarchical single-agent, multi-action reinforcement learning problem. CMAT treats all agents as a unified entity and employs a Transformer encoder to process the large joint observation space. To handle the extensive joint action space, we introduce a hierarchical decision-making mechanism in which a Transformer decoder iteratively refines a high-level consensus vector, simulating the process by which agents reach agreement on their strategies in latent space. Conditioned on this consensus, all agents generate their actions simultaneously, enabling order-independent joint decision making and avoiding the sensitivity to action-generation order in conventional Multi-Agent Transformers (MAT). This factorization allows the joint policy to be optimized using single-agent PPO while preserving expressive coordination through the latent consensus. To evaluate the proposed method, we conduct experiments on benchmark tasks from StarCraft II, Multi-Agent MuJoCo, and Google Research Football. The results show that CMAT achieves superior performance over recent centralized solutions, sequential MARL methods, and conventional MARL baselines. The code for this paper is available at an anonymous repository: `https://anonymous.4open.science/r/CMAT-416A`.

1 INTRODUCTION

Cooperative Multi-Agent Reinforcement Learning (MARL) has become an important framework for solving complex real-world problems such as autonomous fleet coordination, traffic signal optimization, and robotic swarm control. In many of these settings, fully centralized control is feasible and sometimes even necessary; for example, ride-hailing order dispatch often requires global coordination to avoid redundant assignments. However, the joint observation and action spaces typically grow exponentially with the number of agents, giving rise to the Curse of Dimensionality (CoD) Hady et al. (2025). A common way to mitigate this issue is to decompose the original problem into multiple decentralized agents that learn collaboratively. Although this decomposition improves scalability, it also introduces substantial challenges, including non-stationarity, unstable training dynamics, and poor credit assignment, which can ultimately limit both empirical performance and theoretical guarantees Jin et al. (2025).

A major line of research addresses these difficulties through the Centralized Training Decentralized Execution (CTDE) paradigm. Representative methods such as QMIX Rashid et al. (2020), Mean Field MARL Yang et al. (2018), and COMA Foerster et al. (2018) use a centralized critic to exploit global state information or neighborhood interactions during training, thereby alleviating instability and credit assignment issues. Nevertheless, under decentralized execution, agents still act independently, which restricts cooperation because each agent can only infer the behavior of others from past training experience. Similar limitations also arise in many implicit consensus-based MARL approaches Oroojlooy & Hajinezhad (2023). More recently, explicit communication mechanisms have been explored Li et al. (2025), allowing agents to exchange information during both training and execution.

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.

While promising, these methods introduce additional design challenges, including how to choose communication partners and what information should be shared.

Motivated by these limitations, another line of research has shifted toward fully centralized solutions, namely Centralized Training Centralized Execution (CTCE). Among these approaches, the Multi-Agent Transformer (MAT) Wen et al. (2022) has emerged as a representative framework by formulating cooperative MARL as a sequential decision-making problem. Specifically, MAT employs a centralized Transformer Vaswani et al. (2017) encoder to capture relationships among all agents' observations and uses a decoder to generate actions autoregressively. However, the resulting policy can be highly sensitive to the action-generation order. Although recent studies have attempted to optimize this order jointly with action selection Hu et al. (2025); Takayama & Fujita (2025), doing so substantially increases the complexity of the problem by expanding the search space to $n!$, where n denotes the number of agents. In the same spirit, the recent Triple-BERT Zhao & Li (2026) explores an alternative direction by modeling joint action probabilities directly and simultaneously. However, this method relies on a structured policy space that may limit action expressiveness, making it primarily suitable for trip-vehicle assignment problems.

To address these limitations, we propose the Consensus Multi-Agent Transformer (CMAT), which recasts fully cooperative, fully observable MARL as a hierarchical single-agent, multi-action reinforcement learning problem. Building upon MAT, CMAT replaces sequential action generation with an iterative consensus-generation process in the decoder, which simulates how agents reach agreement on their strategies in latent space. Once the consensus vector is obtained, all agents generate their actions simultaneously while conditioning on this shared high-level strategy, allowing each agent to act with awareness of the strategies of the others. We model the joint action probability as the product of individual action probabilities conditioned on the consensus vector, which makes the overall framework amenable to optimization with single-agent Proximal Policy Optimization (PPO) Schulman et al. (2017). This reformulation removes the sensitivity to action-generation order by construction and enables the whole system to be trained end-to-end as a single agent, rather than as n interacting learners. We evaluate CMAT on a broad range of MARL benchmarks, including StarCraft II Whiteson et al. (2019), Multi-Agent MuJoCo de Witt et al. (2020), and Google Research Football Kurach et al. (2020). Experimental results show that CMAT consistently outperforms strong baselines under fully observable cooperative settings.

2 Preliminaries

2.1 Problem Formulation

We consider cooperative Markov games, represented by $< \mathcal{N}, \mathcal{O}, \mathcal{A}, \mathrm{R}, \mathrm{P}, \gamma >$ Littman (1994). Here, $\mathcal{N} = \{1, 2, \ldots, n\}$ denotes the set of agents, $\mathcal{O} = \{o^1, o^2, \ldots, o^n\}$ denotes the observation set of each agent, and $\mathcal{A} = \{a^1, a^2, \ldots, a^n\}$ denotes the action set of the agents. The joint reward function is defined as $\mathrm{R} : \mathcal{O} \times \mathcal{A} \to \mathbb{R}$, and the transition kernel is given by $\mathrm{P} : \mathcal{O} \times \mathcal{A} \to \Delta(\mathcal{O})$. The discount factor is denoted by $\gamma \in [0, 1)$. We further denote the joint policy by $\pi = \{\pi^1, \pi^2, \ldots, \pi^n\}$. At each time step t, all agents act simultaneously and receive a joint reward $\mathrm{R}(\mathcal{O}_t, \mathcal{A}_t)$, while the next observation $\mathcal{O}_{t+1}$ is generated according to $\mathrm{P}(\mathcal{O}_{t+1} \mid \mathcal{O}_t, \mathcal{A}_t)$. The objective is to maximize the long-term discounted cumulative reward, defined as $J = \sum_{t=0}^{\infty} \gamma^t \mathrm{R}(\mathcal{O}_t, \mathcal{A}_t)$, and the corresponding optimal policy is $\pi^* = \arg\max_\pi J$.

In this paper, we focus on a fully cooperative and fully observable setting, where the observations and policies of all agents are available to one another. This assumption is equivalent to a centralized controller with complete information Wen et al. (2022); Zhao & Li (2026). Although this excludes settings that require decentralized execution or that involve partial observability and non-shared rewards, the resulting regime is far from a corner case: it directly captures a broad class of real-world systems in which a single operator observes the global state and dispatches all actors under a common objective, such as ride-hailing order dispatch, traffic signal control, warehouse robot coordination, and emergency fleet dispatch. In these applications, the value of a method is determined precisely by how well it coordinates actors under global information, which is the regime we target. In addition, we consider a model-free setting, in which the transition function P is learned implicitly rather than modeled explicitly. Under this formulation, we define the value functions for single agent and agent set in Appendix B.

From Multi-Agent to Multi-Action Single-Agent. The setting above admits an equivalent single-agent description. Because all agents share the observation $\mathcal{O}$, share the reward R, and act simultaneously, the collection of n agents can be regarded as one meta-agent that, at each step, observes $\mathcal{O}$ and emits a single *composite (multi-)action* $\mathcal{A} = (a^1, \ldots, a^n)$. This viewpoint is not merely a formal convenience: it is the natural model for the fully cooperative and fully observable regime we target. With a shared objective and complete information there is no informational asymmetry or conflict of interest to preserve across agents, so the per-agent boundaries carry no decision-theoretic content and can be absorbed into a single controller without loss. The reformulation yields a single-agent Markov Decision Process (MDP) $\langle \mathcal{O}, \bar{\mathcal{A}}, \mathrm{R}, \mathrm{P}, \gamma \rangle$, where the composite action space is the product $\bar{\mathcal{A}} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^n$, and R, P, γ are inherited unchanged from the Markov game. A joint policy $\pi(\mathcal{A} \mid \mathcal{O})$ over the Markov game is exactly a single-agent policy over $\bar{\mathcal{A}}$, and the two problems share the same objective J. Casting the problem this way is what allows the entire system to be optimized with single-agent RL rather than with per-agent updates.

The remaining difficulty is that the composite action space grows as $\prod_i |\mathcal{A}^i|$, which is intractable to model as a flat categorical distribution. To keep the single-agent view while retaining tractability, we adopt a *hierarchical* MDP in which the meta-agent first selects a high-level action, the latent consensus $c \in \mathcal{C}$, and then selects the low-level composite action conditioned on c. This induces the factorization

$$\pi(\mathcal{A} \mid \mathcal{O}) = \pi^c(c \mid \mathcal{O}) \prod_{i=1}^{n} \pi^i(a^i \mid \mathcal{O}, c), \tag{1}$$

where the high-level policy π^c plays the role of a coordinator and the low-level factors π^i produce individual actions in parallel. Two properties make this factorization a reasonable model rather than a crude approximation. First, factoring a high-dimensional composite action into per-component conditionals given a shared context is standard practice in single-agent policies with factored or multi-discrete action spaces Schulman et al. (2017); Yu et al. (2022): the network hidden layers are shared and the conditional independence is imposed only at the *output layer*, so the dependence among the components is carried up to the final readout, and only the output distributions follow a product form given that shared representation. Second, in our design the shared context is precisely the latent consensus c, which is refined to encode a coordinated team strategy before any action is emitted, and every factor is conditioned on it. Under this hierarchical MDP, CMAT does not decompose the problem into n interacting learners; it instantiates a single multi-action policy whose internal structure captures inter-actor dependence through the shared latent c. We detail the corresponding architecture and optimization in the next section.

2.2 Multi-Agent Transformer and Its Variants

MAT Wen et al. (2022) formulates the cooperative Markov game as a sequential model based on the Transformer architecture Vaswani et al. (2017), with the goal of capturing action dependencies among agents. Specifically, it first employs an encoder to extract observation features for all agents, denoted by $\hat{O} = \{\hat{o}^1, \hat{o}^2, \ldots, \hat{o}^n\}$, and predicts the V-value $\mathrm{V}(\hat{o}^i)$ from each observation feature $\hat{o}^i$. It then uses a decoder to generate agent actions *sequentially*, where self-attention captures inter-agent relationships and cross-attention models the dependence between actions and agents' observations. For simplicity, we assume that the decision order proceeds from agent 1 to agent n, and we denote the parameters of the critic and actor by ϕ and θ, respectively. (*Note that ϕ and θ partially overlap.*) The loss functions can then be expressed as:

		Agent 1 A	Agent 1 B
Agent 2	A	1	-100
Agent 2	B	0	100

Figure 1: Payoff Matrix of Cooperative Game: The joint actions (A, A) and (B, B) are both NE, while (B, B) is the global optimum.

$$\begin{aligned} \mathrm{L}^{MAT}_{Critic}(\phi) &= \mathbf{E}_{i \in \mathcal{N}, t \in \mathcal{T}} \left[(\mathrm{R}(\mathcal{O}_t, \mathcal{A}_t) + \gamma \mathrm{V}_{\phi^-}(\hat{o}^i_{t+1}) - \mathrm{V}_\phi(\hat{o}^i_t))^2 \right] , \\ \mathrm{L}^{MAT}_{Actor}(\theta) &= \mathbf{E}_{i \in \mathcal{N}, t \in \mathcal{T}} \left[\min \left(r^i_t(\theta) \mathrm{A}(\mathcal{O}_t, \mathcal{A}_t), \mathrm{CLIP}(r^i_t(\theta), 1-\epsilon, 1+\epsilon) \mathrm{A}(\mathcal{O}_t, \mathcal{A}_t) \right) \right] , \end{aligned} \tag{2}$$

where ϕ^- are the parameters of the target critic network, $r^i_t(\theta)$ is defined as $\frac{\pi^i_\theta(a^i_t | \mathcal{O}_t, a^{1:i-1}_t)}{\pi^i_{\theta^-}(a^i_t | \mathcal{O}_t, a^{1:i-1}_t)}$, θ^- are the network parameters used for sample collection, and $\mathcal{T}$ is defined as $\{0, 1, 2, \ldots, \infty\}$. The advantage function is estimated using Generalized Advantage Estimation (GAE) Schulman et al. (2015) with the estimated V-value defined as $\mathrm{V}_\phi(\mathcal{O}_t) = \frac{1}{n} \sum_{i=1}^{n} \mathrm{V}_\phi(\hat{o}^i_t)$.

Although MAT Wen et al. (2022) has demonstrated promising performance when combined with Multi-Agent Advantage Decomposition Kuba et al. (2022); Zhong et al. (2024), which theoretically guarantees that this optimization scheme can improve the joint advantage, several studies Hu et al. (2025); Takayama & Fujita (2025) have observed that the decision order of agents can substantially affect performance. A primary reason for this phenomenon lies in incorrect credit assignment induced by biased value-function estimation. In particular, for leading agents, the value estimation often fails to adequately capture the influence of subsequent agents. Given agent i, the real advantage function should be (derived from Eq. 13 and Eq. 14):

$$\begin{aligned} \mathrm{A}^i(\mathcal{O}, a^{1:i-1}, a^i) &= \mathrm{Q}^{1:i}(\mathcal{O}, a^{1:i}) - \mathrm{Q}^{1:i-1}(\mathcal{O}, a^{1:i-1}) \,, \\ &= \mathbf{E}_{\hat{a}^{i:n}} \left[\mathrm{Q}(\mathcal{O}, [a^{1:i}, \hat{a}^{i+1:n}]) - \mathrm{Q}(\mathcal{O}, [a^{1:i-1}, \hat{a}^{i:n}]) \right] \,. \end{aligned} \tag{3}$$

However, in the loss function defined in Eq. 2, $\mathrm{A}^i(\mathcal{O}, a^{1:i-1}, a^i)$ is replaced by $\mathrm{A}(\mathcal{O}, \mathcal{A})$, thereby introducing the effects of subsequent agents' actions into the credit assignment of leading agents. Conversely, for following agents, another inconsistency arises between the joint V-value $\mathrm{V}(\mathcal{O})$, which depends only on joint observations, and the actors' behavior $\pi^i(\mathcal{O}, a^{1:i-1})$, which additionally conditions on the actions of preceding agents. Both issues can distort the optimization direction and ultimately hinder effective cooperation within the multi-agent system.

To illustrate a failure scenario involving MAT, consider the cooperative game given in Table **??**. This game consists of a single step where both agents have two possible actions. The values in the table represent the long-term team reward, with MAT following the decision order of Agent 1 leading to Agent 2, whereas in our formulation the two actions are selected simultaneously from a shared consensus. We assume that the initial V-value of all states is set to 0. If Agent 1 chooses action B and Agent 2 subsequently selects action A, this results in a substantial negative reward and advantage value of -100. As a consequence, the probabilities $\pi^1(a^1 = B|\mathcal{O})$ and $\pi^2(a^2 = A|\mathcal{O}, a^1 = B)$ will experience a significant decline. This situation can trigger an immediate reduction in action entropy, impeding the model's ability to explore the optimal action combination (B, B) due to insufficient exploration-exploitation mechanisms; specifically, the probability of Agent 1 exploring action B, $\pi^1(a^1 = B|\mathcal{O})$, decreases dramatically. Subsequently, when Agent 1 opts for action A with increased probability, the positive reward and advantage value will further elevate the action probability of $\pi^1(a^1 = A|\mathcal{O})$ while correspondingly reducing $\pi^1(a^1 = B|\mathcal{O})$, thereby exacerbating this dilemma.

Although Hu et al. (2025); Takayama & Fujita (2025) attempted to mitigate this issue by learning the decision order and agent actions simultaneously, such an approach substantially increases training complexity, as the search space over action orders grows to $n!$. More importantly, greater flexibility in ordering does not fundamentally resolve the optimization limitations of sequential decision making. *According to the theoretical analyses in Kuba et al. (2022); Liu et al. (2024a); Zhong et al. (2024), such sequential updating methods are guaranteed to converge only to a Nash Equilibrium (NE). For example, in the game in Table* **??***, conventional MAT-based methods may converge to a Pareto-suboptimal NE. This limitation motivates us to move beyond sequential MARL formulations and instead reformulate cooperative MARL as a single-agent, multi-action problem, which allows the whole system to be trained end-to-end with single-agent RL and removes the order-dependent bias by construction.* To the best of our knowledge, such a formulation has not yet been established in the existing literature, and it is the modeling and optimization structure it induces, rather than any tightening of the convergence rate, that we exploit in what follows.

From this perspective, a closely related line of research is to formulate cooperative decision making directly within a SARL framework. Recently, Zhao & Li (2026) proposed Triple-BERT, a SARL framework for ride-sharing tasks, which aims to directly learn the joint action probability using a BERT model Devlin et al. (2019). Specifically, it assumes that the joint action probability $\pi(\mathcal{A}|\mathcal{O})$ can be expressed as $\mathrm{z}(\prod_{i=1}^{n} \pi^i(a^i|\mathcal{O}))$, where $\mathrm{z}(\cdot)$ is an increasing mapping function. However, this assumption does not always hold in practice, and as a result, the method can still suffer from credit assignment issues similar to those in MAT. For instance, in the example shown in Fig. **??**, exploring the action (B, A) can reduce the probability of $\pi^1(a^1 = B|\mathcal{O})$, thereby making it less likely to explore the optimal action combination (B, B). Due to page limitations, a more detailed discussion of related work is provided in Appendix A.

3 METHODOLOGY

3.1 OVERVIEW

In this paper, we propose an order-independent MAT Wen et al. (2022) inspired by the consensus mechanism, named the CMAT. Unlike conventional MAT, our approach utilizes a decoder to iterate a consensus vector instead of specifying detailed actions, simulating the process by which all agents reach an agreement on their strategies in latent space. This instantiates the hierarchical MDP introduced in Eq. 1: we treat the whole system as a single multi-action agent, where the consensus c serves as a high-level action that summarizes the shared strategy and guides the low-level actions a^i, and the per-agent action probability $\pi^i(a^i|\mathcal{O}, c)$ is modeled independently conditioned on c. In this way, we treat all agents as a unified entity and directly apply a single-agent PPO approach Schulman et al. (2017) to optimize the joint action policy $\pi(\mathcal{A}|\mathcal{O})$ of Eq. 1.

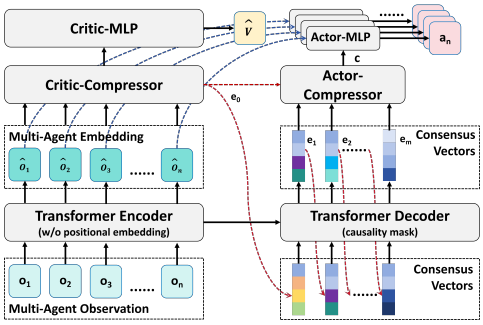


Figure 2: Network Architecture

As discussed in Eq. 1, the conditional independence of the per-agent factors is imposed at the output layer, so for a fixed observation the joint action distribution is a single product distribution over the heads. Inter-actor dependence reaches those heads through two channels: the encoder's bi-directional self-attention, which makes every observation embedding $\hat{o}^i$ a function of the full $\mathcal{O}$, and the consensus vector c, which is itself a function of $\mathcal{O}$ and is read by every head. Because $c = \mu_\theta(\mathcal{O})$ is indexed by the observation, the family of distributions CMAT expresses across observations is not confined to a single product distribution, and over a continuous consensus space it reaches the full joint policy class that MAT obtains through the chain rule (Appendix F.8). What CMAT does not reproduce is MAT's autoregressive coupling of the output distributions, which is where the order-dependence of MAT originates.

The overall network architecture is illustrated in Fig. 2 and is built on the Transformer backbone Vaswani et al. (2017). First, the encoder extracts features and relationships among the observations of all agents using bi-directional self-attention, resulting in an observation embedding sequence $\hat{\mathcal{O}}$. Next, a Critic-Compressor compresses this sequence into a single vector e^0, which is then used by the Critic-MLP to estimate the V-value $\mathrm{V}(\mathcal{O})$. This vector is also input to the Transformer decoder to perform consensus iteration for m times, producing a sequence of consensus vectors $\{e^1, e^2, \ldots, e^m\}$. Subsequently, an Actor-Compressor compresses the set $\mathcal{E} = \{e^0, e^1, e^2, \ldots, e^m\}$ into a single vector c, representing the final consensus and reflecting the potential strategy of all agents. Finally, the actor is generated by the Actor-MLP, which combines the consensus vector c with the observation embedding $\hat{o}^i$ of each agent. Below we will discuss the details of each component within this architecture.

The design choices above are not merely empirical conveniences, and Appendix E makes the reason precise. A single aggregation of the observation embeddings, whether the initial vector e^0 or any mean-pooling, factors through the first moment of the embedding set, so at a fixed width it cannot represent a consensus that depends on pairwise or higher-order structure among the agents, whereas the iterated cross-attention in the decoder reweights the tokens by data-dependent scores at every step (Proposition 1, Theorem 3). Weight-shared iteration is also the parameter-efficient route to such a consensus: reaching a given coordination accuracy by stacking an encoder costs a number of layers that grows with the stiffness of the coordination problem, while the decoder reaches the same accuracy with one reused block and additional inference compute (Theorem 4), which is the right trade at the parameter scales where deep RL remains trainable. These results are stated for the representation the network can express, and Section 4 shows that training reaches it.

3.2 NETWORK ARCHITECTURE

A. Encoder: In the proposed framework, we first utilize a Transformer encoder to extract features and relationships among agents. Specifically, since the input needs to be order-independent, we

eliminate the positional embedding that is commonly used in similar works Zhao & Li (2026); Wen et al. (2022). The extraction process can be expressed as:

$$\hat{\mathcal{O}} = \mathrm{Encoder}(\mathcal{O}) \in \mathbb{R}^{n,d} \ , \tag{4}$$

where d is the hidden dimension. Removing the positional embedding makes the encoder *permutation equivariant*: for any permutation ρ of the agent indices, $\mathrm{Encoder}(\rho \cdot \mathcal{O}) = \rho \cdot \mathrm{Encoder}(\mathcal{O})$, since self-attention without positional information treats its inputs as a set rather than a sequence. This property has been formally established for attention-based set models Zaheer et al. (2017); Lee et al. (2019). Consequently, permuting the order in which agent observations are fed to the encoder permutes the output embeddings identically but leaves their content unchanged; because the subsequent Critic-Compressor and Actor-Compressor aggregate these embeddings in a permutation-invariant manner (Eq. 5), the estimated V-value and the generated consensus are invariant to agent ordering, and each agent's action depends only on its own embedding $\hat{o}^i$ and the shared consensus c. CMAT therefore attains order independence by construction, without relying on any explicit ordering module.

Afterwards, we employ the Critic-Compressor to reduce $\hat{\mathcal{O}}$ into a single vector, referred to as the initial consensus vector e^0, represented by the following process:

$$\begin{aligned} x &\in \mathbb{R}^{N \times d_1} \ , \\ M &= \text{Softmax}(\text{MLP}(x), \text{dim=0}) \in \mathbb{R}_+^{N,h} \ , \\ z &= M^T \cdot x \in \mathbb{R}^{d_1,h} \ , \\ y &= \text{MLP}(\text{Flatten}(z)) \in \mathbb{R}^{d_2} \ , \end{aligned} \tag{5}$$

where x and y represent the input and output of the compressor, respectively, N is the length, d_1, h, and d_2 are hidden dimensions, and M and z are intermediate variables. This structure is commonly used in sequence compression Zhao et al. (2025); Chou et al. (2024), motivated by the following considerations: First, the intermediate feature z is constructed from the original sequence x using h different combinations. For the ith combination, it derives from a mixture of the original sequence, weighted by the ith row of the weight matrix M. Finally, the output hidden feature is produced by applying an MLP to compress the intermediate feature z. In the context of the Critic-Compressor, x and y correspond to $\hat{\mathcal{O}}$ and e^0, respectively, while dimensions d_1 and d_2 are both set to d.

Subsequently, we use the Critic-MLP to process the initial consensus vector e^0 to obtain the estimated V-value, expressed as:

$$\begin{aligned} \hat{\mathrm{V}}(\mathcal{O}) &= \text{Critic-MLP}(e^0) \ , \\ e^0 &= \text{Critic-Compressor}(\hat{\mathcal{O}}) \ . \end{aligned} \tag{6}$$

B. Decoder: For the decoder, we first auto-regressively iterate the consensus vector to achieve a converged strategy among agents, formulated as:

$$\{e^1, e^2, \ldots, e^m\} = \mathrm{Decoder}(e^0) \ . \tag{7}$$

Unlike MAT, the positional embedding of the decoder is retained because we want the model to be aware of the convergence process of the consensus.

Following this, we utilize the Actor-Compressor to compress $\mathcal{E}$ to obtain the final consensus vector, expressed as:

$$c = \text{Actor-Compressor}(\mathcal{E}) \ , \tag{8}$$

which follows the same procedural process as the Critic-Compressor defined in Eq. 5. *Here, we choose to utilize the combination of the entire set $\mathcal{E}$ instead of only the last generated vector e^m as the consensus. This decision helps to prevent any potential information loss that may occur during the iteration process.*

Finally, we combine the consensus vector c with the observation feature $\hat{o}^i$ and feed them to the Actor-MLP to generate the action for agent i, given by:

$$a^i = \text{Actor-MLP}([\hat{o}^i; c]) \ . \tag{9}$$

3.3 TRAINING PROCESS

A. Training: In our CMAT, we view all agents as a unified entity through a single-agent, multi-action perspective. The policy is optimized directly using a single-agent PPO approach:

$$\begin{aligned} \mathrm{L}_{Critic}^{CMAT}(\phi) =& \mathbf{E}_{t\in\mathcal{T}}\left[(\mathrm{R}(\mathcal{O}_t,\mathcal{A}_t)+\gamma \mathrm{V}_{\phi^-}(\mathcal{O}_{t+1})-\mathrm{V}_{\phi}(\mathcal{O}_t))^2\right] , \\ \mathrm{L}_{Actor}^{CMAT}(\theta) =& \mathbf{E}_{i\in\mathcal{N},t\in\mathcal{T}}\left[\min\left(\mathcal{R}_t(\theta)\mathrm{A}(\mathcal{O}_t,\mathcal{A}_t), \mathrm{CLIP}(\mathcal{R}_t(\theta),1-\epsilon,1+\epsilon)\mathrm{A}(\mathcal{O}_t,\mathcal{A}_t)\right)\right] , \end{aligned} \tag{10}$$

where the ratio $\mathcal{R}_t(\theta)$ is defined as (derived from Eq. 1):

$$\mathcal{R}_t(\theta)=\frac{\pi_\theta(\mathcal{A}_t|\mathcal{O}_t)}{\pi_{\theta^-}(\mathcal{A}_t|\mathcal{O}_t)}=\frac{\pi_\theta^c(c|\mathcal{O})\prod_{i=1}^n\pi^i(a^i|\mathcal{O},c)}{\pi_{\theta^-}^c(c|\mathcal{O})\prod_{i=1}^n\pi^i(a^i|\mathcal{O},c^-)}=\frac{\prod_{i=1}^n\pi_\theta^i(a^i|\mathcal{O},\mu_\theta(\mathcal{O}))}{\prod_{i=1}^n\pi_{\theta^-}^i(a^i|\mathcal{O},\mu_{\theta^-}(\mathcal{O}))} , \tag{11}$$

where $c=\mu_\theta(\mathcal{O})$ and $c^-=\mu_{\theta^-}(\mathcal{O})$ are consensus vectors generated by the current and old actors, respectively, and $\mu_\theta(\cdot)$ is the consensus generation network. Because μ_θ is deterministic, the high-level ratio reduces to the indicator $\mathbb{1}[c=\mu_\theta(\mathcal{O})]$, which is 1 on the trajectory actually collected by μ_θ and 0 elsewhere, so the high-level factor contributes a unit weight on support and the ratio is carried entirely by the action policies conditioned on the corresponding consensus. Similar to TD3 and DDPG, the consensus network is then updated through the gradient that flows back into $\mu_\theta(\mathcal{O})$ from the actor loss rather than through a high-level importance weight. We adopt a deterministic consensus for training stability and to keep the latent space free of importance weighting; a stochastic consensus policy, together with the associated cooperative Stackelberg (bilevel) formulation, is a natural generalization that we discuss as future work in Appendix F.

To illustrate how CMAT resolves the dilemma shown in Fig. **??**, consider again the joint action (B, A). Under our formulation, the affected probability is $\pi^1(a^1=B|\mathcal{O},c)$, which is conditioned on the specific consensus c and therefore tied only to the particular latent strategy that led to the suboptimal joint action. As a result, reducing this probability mainly indicates that the consensus c (i.e., $\mu(\mathcal{O})$) is suboptimal, rather than penalizing action B for Agent 1 in an unconditional manner. In contrast, the policy under the optimal consensus c^*, namely $\pi^1(a^1=B|\mathcal{O},c^*)$, remains unaffected. Meanwhile, the consensus generation module $\mu(\mathcal{O})$ is updated through gradient descent based on the overall advantage, gradually steering $\mu(\mathcal{O})$ toward c^* over time.

B. Fine-tuning: In the joint training phase above, the consensus generation policy $\mu_\theta(\mathcal{O})$ and the action policy $\pi_\theta(\cdot|\mathcal{O},c)$ are updated at the same time. This creates a moving-target problem: the action heads are chasing a consensus representation that is itself still changing, and conversely the consensus is optimized against action heads that have not yet adapted to it. Such coupled updates can interfere with each other and slow convergence, a difficulty familiar from bilevel and actor-critic style optimization where the inner and outer variables are trained jointly. To decouple the two, we introduce a fine-tuning phase that fixes one part of the model and refines the other, using the same loss as in Eq. 10. Concretely, we provide two complementary variants:

(i) Consensus Enhancement: We fine-tune the critic $\mathrm{V}_\phi(\mathcal{O})$ and the consensus generation policy $\mu_\theta(\mathcal{O})$ while keeping the action policy $\pi_\theta(\cdot|\mathcal{O},c)$ fixed, so that only the gradients of the Critic-MLP, decoder, and Actor-Compressor are active. This variant treats the current action heads as a fixed set of followers and searches for a better high-level coordination signal for them. It is the natural choice when the per-agent action heads are already competent but the team-level coordination encoded in c is suboptimal, i.e., when the bottleneck is coordination rather than individual control.

(ii) Action Policy Enhancement: We fine-tune the critic $\mathrm{V}_\phi(\mathcal{O})$ and the action policy $\pi_\theta(\cdot|\mathcal{O},c)$ while fixing the consensus generation policy $\mu_\theta(\mathcal{O})$, so that only the Critic-MLP and Actor-MLP receive gradients. This variant freezes the coordination signal and lets each agent better exploit an already-informative consensus. It is preferable when the consensus is already meaningful but the individual action heads underfit it, i.e., when the bottleneck is per-agent execution rather than coordination.

Because the two variants optimize the same objective while alternating which block is held fixed, they can be viewed as two directions of a single block-wise refinement of the joint policy from a single-agent perspective; the appropriate choice for a given task depends on which of the two bottlenecks dominates. Empirically we find that both yield similar final performance, indicating that CMAT is not sensitive to this choice in the benchmarks we consider. The whole training process is provided at Appendix C. A rigorous analysis of why the deployed deterministic architecture is

effective is given in Appendix E, and a higher-level discussion of a more general stochastic-consensus framework as future work is provided in Appendix F.

4 EXPERIMENT

4.1 EXPERIMENT SETUP

To validate the efficiency of our proposed method, we evaluate its performance in a series of benchmark MARL experiment scenarios, including:

- **StarCraft II Whiteson et al. (2019):** A challenging real-time strategy game environment that provides complex micromanagement tasks for testing multi-agent cooperation and coordination. Since many advanced MARL methods can achieve a 100% win rate in simpler environments Wen et al. (2022), we focus on the most difficult task scenarios for comparison in this paper, including "MMM2", "6h vs 8z", and "3s5z vs 3s6z".
- **Multi-Agent MuJoCo de Witt et al. (2020):** A set of continuous control robotic tasks adapted from MuJoCo, where multiple agents must coordinate to control a single or multiple articulated bodies, testing fine-grained cooperation. We select three challenging scenarios, each with a single agent controlling one body: "8×1-Agent Ant", "6×1-Agent HalfCheetah", and "6×1-Agent Walker2d".
- **Google Research Football Kurach et al. (2020):** A highly realistic football simulation platform that requires agents to master individual skills and long-horizon teamwork strategies in a dynamic, physics-based environment. The detailed tasks include "academy counterattack easy", "academy pass and shoot with keeper", and "academy 3 vs 1 with keeper".

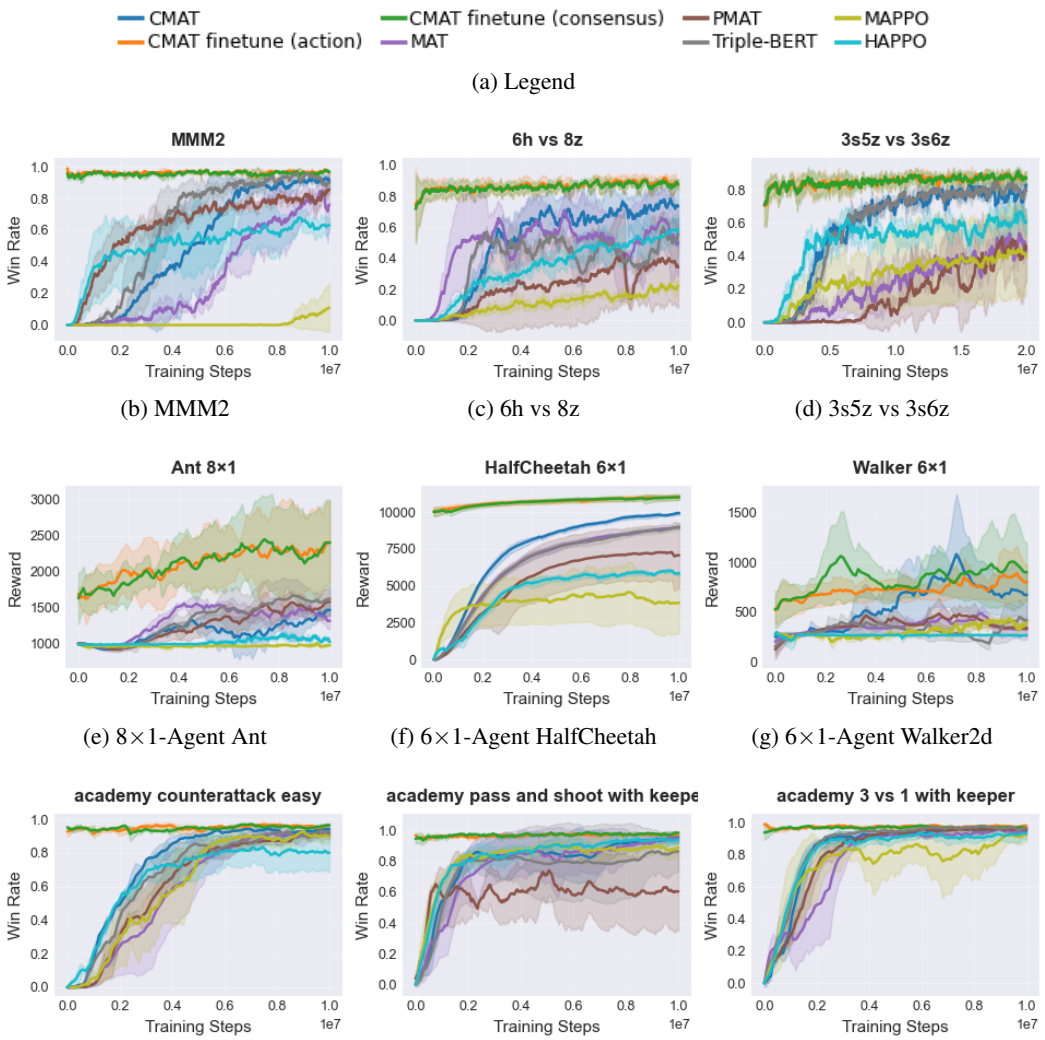
To illustrate the superior performance of our method, we compare it against several strong baselines, including MAT Wen et al. (2022), PMAT Hu et al. (2025), Triple-BERT Zhao & Li (2026), HAPPO Kuba et al. (2022); Zhong et al. (2024), and MAPPO Yu et al. (2022), using the same settings as reported in Wen et al. (2022). More details about the methods description and experiment configurations can be found at Appendix D.1.

4.2 EXPERIMENT RESULTS

The experimental results are shown in Fig. 3. Note that the two CMAT-finetune curves start with high performance because they are initialized from the well-trained CMAT. We observe that CMAT achieves superior performance in most scenarios, and its advantage becomes more evident after fine-tuning, with both CMAT-finetune variants achieving the best performance across all scenarios. The improvement is consistent across the three benchmark families, which stress different aspects of coordination: discrete micromanagement in SMAC, continuous high-dimensional control in MA-MuJoCo, and sparse-reward long-horizon cooperation in Google Research Football. This breadth indicates that the gains are not tied to a particular action type or reward structure, but follow from the consensus formulation itself. The margin over the baselines tends to widen on the harder, larger-agent scenarios, where coordinating actions is most difficult and the shared consensus provides the most benefit; on the easier scenarios the strong baselines are already close to saturation, so the gap is naturally smaller. Beyond mean return, CMAT typically exhibits comparable or lower variance across the five seeds than the MAT-family baselines, which we attribute to the cleaner single-agent optimization structure that avoids the sequential per-agent updates. Notably, the results of CMAT-finetune (action) clearly demonstrate the effectiveness of our consensus mechanism. If the consensus were invalid and independent of the states, it could be disregarded by the action head, in which case CMAT would degrade to Triple-BERT, where all agents take actions simultaneously without any consensus.

4.3 ABLATION STUDY AND SENSITIVITY ANALYSIS

We further conduct ablation studies and hyper-parameter sensitivity analyses to isolate the contribution of each design choice; the full setup, figures, and per-scenario curves are deferred to Appendix D.2. Two conclusions are validated. First, replacing the mixed consensus produced by our Actor-Compressor with the single last consensus vector e^m degrades performance in most scenarios, confirming that aggregating the whole consensus trajectory $\mathcal{E}$ preserves information that would



(a) Legend

(b) MMM2 (c) 6h vs 8z (d) 3s5z vs 3s6z

(e) 8×1-Agent Ant (f) 6×1-Agent HalfCheetah (g) 6×1-Agent Walker2d

(h) academy counterattack easy (i) academy pass and shoot with keeper (j) academy 3 vs 1 with keeper

Figure 3: Training Curves under 5 Random Seeds: The shadow parts represent the standard deviation.

otherwise be lost during auto-regressive decoding. Second, setting the number of decoder iterations to the agent count $m = n$ is a robust default: fewer iterations under-form the consensus, while more inject redundant information that complicates training.

5 Conclusion

In this paper, we proposed CMAT, a novel centralized method for fully observable cooperative MARL tasks. By using a Transformer decoder to iteratively generate a consensus representation, CMAT reformulates cooperative MARL as a hierarchical single-agent, multi-action problem, in which all agents act simultaneously while conditioning on a shared consensus and full information about one another. This formulation lets the whole system be optimized with single-agent PPO and alleviates several structural limitations of conventional MAT, including order dependency and actor-critic inconsistency, and it avoids the sequential updating scheme under which MAT-family methods are guaranteed to converge only to a Nash Equilibrium. Extensive experiments on a series of standard MARL benchmarks demonstrate that CMAT consistently outperforms strong baselines. Additional discussions are provided in Appendix G.

AI USE STATEMENT

In this work, we used generative AI tools for paper polishing, including grammar and wording checks, verification and support of mathematical derivations, and coding. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work adheres to the principles outlined in the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

The code for this paper is available at `https://anonymous.4open.science/r/CMAT-416A`.

REFERENCES

Daniel Adelman and Adam J Mersereau. Relaxations of weakly coupled stochastic dynamic programs. *Operations Research*, 56(3):712–727, 2008.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Robert J Aumann. Subjectivity and correlation in randomized strategies. *Journal of mathematical Economics*, 1(1):67–96, 1974.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 690–701, 2019.

Dimitri P Bertsekas. Neuro-dynamic programming. In *Encyclopedia of optimization*, pp. 1–6. Springer, 2025.

Johan Björck, Carla P Gomes, and Kilian Q Weinberger. Towards deeper deep reinforcement learning with spectral normalization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8242–8255, 2021.

René Carmona, François Delarue, et al. *Probabilistic theory of mean field games with applications I-II*, volume 3. Springer, 2018.

Chang Chen, Jaesik Yoon, Yi-Fu Wu, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. In *Deep RL Workshop NeurIPS 2021*, 2021a.

Dingyang Chen, Zhenyu Zhang, Xiaolong Kuang, Xinyang Shen, Ozalp Ozer, and Qi Zhang. Convergence rates of bayesian network policy gradient for cooperative multi-agent reinforcement learning. In *NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty*, 2024.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021b.

Yi-Hui Chou, I-Chun Chen, Joann Ching, Chin-Jui Chang, and Yi-Hsuan Yang. Midibert-piano: Large-scale pre-training for symbolic music classification tasks. *Journal of Creative Music Systems*, 8(1), 2024.

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*, pp. 1538–1546. PMLR, 2019.

Christian Schroeder de Witt, Bei Peng, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Deep multi-agent reinforcement learning for decentralized continuous cooperative control. *arXiv preprint arXiv:2003.06709*, 19, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Tanner Fiez, Benjamin Chasnov, and Lillian Ratliff. Implicit learning dynamics in stackelberg games: Equilibria characterization, convergence analysis, and empirical study. In *International conference on machine learning*, pp. 3133–3144. PMLR, 2020.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Ian Goodfellow. Deep learning, 2016.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pp. 399–406, 2010.

Mohamad A Hady, Siyi Hu, Mahardhika Pratama, Zehong Cao, and Ryszard Kowalczyk. Multi-agent reinforcement learning for resources allocation optimization: a survey. *Artificial Intelligence Review*, 58(11):354, 2025.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Jiang Hao and Pradeep Varakantham. Hierarchical value decomposition for effective on-demand ride-pooling. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 580–587, 2022.

Matthew J Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *AAAI fall symposia*, volume 45, pp. 141, 2015.

Markus Holzleitner, Lukas Gruber, José Arjona-Medina, Johannes Brandstetter, and Sepp Hochreiter. Convergence proof for actor-critic methods applied to ppo and rudder. In *Transactions on large-scale data-and knowledge-centered systems XLVIII: special issue in memory of univ. prof. dr. roland wagner*, pp. 105–130. Springer, 2021.

Kun Hu, Muning Wen, Xihuai Wang, Shao Zhang, Yiwei Shi, Minne Li, Minglong Li, and Ying Wen. Pmat: Optimizing action generation order in multi-agent reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 997–1005, 2025.

Shengchao Hu, Li Shen, Ya Zhang, Yixin Chen, and Dacheng Tao. On transforming reinforcement learning with transformers: The development trajectory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):8580–8599, 2024a.

Shengchao Hu, Li Shen, Ya Zhang, and Dacheng Tao. Learning multi-agent communication from graph modeling perspective, 2024b.

Siyi Hu, Fengda Zhu, Xiaojun Chang, and Xiaodan Liang. Updet: Universal multi-agent reinforcement learning via policy decoupling with transformers. *arXiv preprint arXiv:2101.08001*, 2021.

Nai-Chieh Huang, Ping-Chun Hsieh, Kuo-Hao Ho, and I-Chen Wu. Ppo-clip attains global optimality: Towards deeper understandings of clipping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12600–12607, 2024.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

Weiqiang Jin, Hongyang Du, Biao Zhao, Xingwu Tian, Bohang Shi, and Guang Yang. A comprehensive survey on multi-agent cooperative decision-making: Scenarios, approaches, challenges and perspectives. *arXiv preprint arXiv:2503.13415*, 2025.

Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *International Conference on Machine Learning*, pp. 5562–5571. PMLR, 2021.

JG Kuba, R Chen, M Wen, Y Wen, F Sun, J Wang, and Y Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *ICLR 2022-10th International Conference on Learning Representations*, pp. 1046. The International Conference on Learning Representations (ICLR), 2022.

Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zając, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 4501–4510, 2020.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Kuang-Huei Lee, Ofir Nachum, Mengjiao Sherry Yang, Lisa Lee, Daniel Freeman, Sergio Guadarrama, Ian Fischer, Winnie Xu, Eric Jang, Henryk Michalewski, et al. Multi-game decision transformers. *Advances in neural information processing systems*, 35:27921–27936, 2022.

Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. Shapley counterfactual credits for multi-agent reinforcement learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 934–942, 2021.

Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212, 2022.

Xinran Li and Jun Zhang. Context-aware communication for multi-agent reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pp. 1156–1164, 2024.

Xinran Li, Xiaolu Wang, Chenjia Bai, and Jun Zhang. Exponential topology-enabled scalable communication in multi-agent reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Runji Lin, Ye Li, Xidong Feng, Zhaowei Zhang, Xian Hong Wu Fung, Haifeng Zhang, Jun Wang, Yali Du, and Yaodong Yang. Contextual transformer for offline meta reinforcement learning. *arXiv preprint arXiv:2211.08016*, 2022.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

Fangchen Liu, Hao Liu, Aditya Grover, and Pieter Abbeel. Masked autoencoding for scalable and generalizable decision making. *Advances in Neural Information Processing Systems*, 35: 12608–12618, 2022.

Jiarong Liu, Yifan Zhong, Siyi Hu, Haobo Fu, QIANG FU, Xiaojun Chang, and Yaodong Yang. Maximum entropy heterogeneous-agent reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=tmqOhBC4a5.

Jie Liu, Yinmin Zhang, Chuming Li, Zhiyuan You, Zhanhui Zhou, Chao Yang, Yaodong Yang, Yu Liu, and Wanli Ouyang. MaskMA: Towards zero-shot multi-agent decision making with mask-based collaborative learning. *Transactions on Machine Learning Research*, 2024b. ISSN 2835-8856.

R Duncan Luce et al. *Individual choice behavior*, volume 4. Wiley New York, 1959.

Hangyu Mao, Zhengchao Zhang, Zhen Xiao, and Zhibo Gong. Modelling the dynamic joint policy of teammates with attention multi-agent ddpg. *arXiv preprint arXiv:1811.07029*, 2018.

Joe Marino, Yisong Yue, and Stephan Mandt. Iterative amortized inference. In *International Conference on Machine Learning*, pp. 3403–3412. PMLR, 2018.

Laëtitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 64–69. IEEE, 2007.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International conference on machine learning*, pp. 6820–6829. PMLR, 2020.

Vishnu Nanduri and Tapas K Das. A reinforcement learning algorithm for obtaining the nash equilibrium of multi-player matrix games. *IIE Transactions*, 41(2):158–167, 2009.

Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 16828–16847. PMLR, 2022.

Afshin Oroojlooy and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53(11):13677–13722, 2023.

Kei Ota, Devesh K Jha, and Asako Kanezaki. Training larger networks for deep reinforcement learning. *arXiv preprint arXiv:2102.07920*, 2021.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. Stabilizing transformers for reinforcement learning. In *International conference on machine learning*, pp. 7487–7498. PMLR, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 1975.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Machel Reid, Yutaro Yamada, and Shixiang Shane Gu. Can wikipedia help offline reinforcement learning? *arXiv preprint arXiv:2201.12122*, 2022.

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.

Satinder P Singh and Richard C Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, pp. 5887–5896. PMLR, 2019.

Jianyu Su, Stephen Adams, and Peter Beling. Value-decomposition multi-agent actor-critics. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11352–11360, 2021.

Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 2016.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087, 2018.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Shota Takayama and Katsuhide Fujita. Aoad-mat: Transformer-based multi-agent deep reinforcement learning model considering agents' order of action decisions. In *International Conference on Principles and Practice of Multi-Agent Systems*, pp. 303–310. Springer, 2025.

Paulina Varshavskaya, Leslie Pack Kaelbling, and Daniela Rus. Efficient distributed reinforcement learning through agreement. In *Distributed Autonomous Robotic Systems 8*, pp. 367–378. Springer, 2009.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Edward Wagstaff, Fabian Fuchs, Martin Engelcke, Ingmar Posner, and Michael A Osborne. On the limitations of representing functions on sets. In *International conference on machine learning*, pp. 6487–6494. PMLR, 2019.

Xiaofeng Wang and Tuomas Sandholm. Reinforcement learning to play an optimal nash equilibrium in team markov games. *Advances in neural information processing systems*, 15, 2002.

Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems*, 35:16509–16521, 2022.

S Whiteson, M Samvelyan, T Rashid, CS De Witt, G Farquhar, N Nardelli, TGJ Rudner, CM Hung, PHS Torr, and J Foerster. The starcraft multi-agent challenge. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pp. 2186–2188, 2019.

Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, 25(A):287–298, 1988.

Philipp Wu, Arjun Majumdar, Kevin Stone, Yixin Lin, Igor Mordatch, Pieter Abbeel, and Aravind Rajeswaran. Masked trajectory models for prediction, representation, and control. In *International Conference on Machine Learning*, pp. 37607–37623. PMLR, 2023.

Quan Xiao, Songtao Lu, and Tianyi Chen. A generalized alternating method for bilevel learning under the polyak-{\L} ojasiewicz condition. *arXiv preprint arXiv:2306.02422*, 2023.

Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In *international conference on machine learning*, pp. 24631–24645. PMLR, 2022.

Taku Yamagata, Ahmed Khalil, and Raul Santos-Rodriguez. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. In *International Conference on Machine Learning*, pp. 38989–39007. PMLR, 2023.

Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. In *International conference on machine learning*, pp. 5571–5580. PMLR, 2018.

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. *Advances in neural information processing systems*, 35:24611–24624, 2022.

Lei Yuan, Ziqian Zhang, Lihe Li, Cong Guan, and Yang Yu. A survey of progress on cooperative multi-agent reinforcement learning in open environment. *arXiv preprint arXiv:2312.01058*, 2023.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Zijian Zhao and Sen Li. Triple-bert: Do we really need marl for order dispatch on ride-sharing platforms? In *The Fourteenth International Conference on Learning Representations*, 2026.

Zijian Zhao, Fanyi Meng, Zhonghao Lyu, Hang Li, Xiaoyang Li, and Guangxu Zhu. Csi-bert2: A bert-inspired framework for efficient csi prediction and classification in wireless communication and sensing. *IEEE Transactions on Mobile Computing*, 2025.

Zijian Zhao, Dian Jin, Zijing Zhou, and Xiaoyu Zhang. Automatic stage lighting control: Is it a rule-driven process or generative task? In *The Fourteenth International Conference on Learning Representations*, 2026.

Qinqing Zheng, Amy Zhang, and Aditya Grover. Online decision transformer. In *international conference on machine learning*, pp. 27042–27059. PMLR, 2022.

Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67, 2024.

APPENDIX CONTENTS

A Related Work **17**

A.1 Cooperative Multi-Agent Reinforcement Learning 17

A.2 Transformer in Reinforcement Learning 18

B Value Function Definition **19**

C Algorithm **19**

D Experiment Supplement **19**

D.1 Experiment Configurations 19

D.2 Ablation Study and Sensitivity Analysis 21

E Why the CMAT Architecture Is Effective **22**

E.1 Setup: Consensus-Separable Team Value Functions 23

E.2 Collapse of the Exponential Joint Maximization 24

E.3 Coordination Dimension versus Suboptimality 24

E.4 Why Iterate: Consensus as Unrolled Coordination Optimization 25

E.5 Why Consensus Beats an Encoder-Only Controller at a Fixed Budget 26

E.6 Synthesis 28

F A More General Framework: Stochastic Consensus and Future Directions **28**

F.1 Preliminary Assumptions 28

F.2 Reformulation as a Cooperative Stackelberg Game 29

F.3 The Policy-Class Ladder and Coordination Rank 29

F.4 Theoretical Justification 1: The Leader's Problem as a Finite MDP 30

F.5 Theoretical Justification 2: Consensus as a Coordination Signal 30

F.6 Theoretical Justification 3: Alternating Optimization as Block Coordinate Ascent . 31

F.7 Policy Gradient and Credit Assignment under Deterministic Consensus 31

F.8 Comparison with MAT: Policy Space Expressiveness 32

F.9 Relation to Existing Convergence Results 32

F.10 Scope and Open Directions 33

G Discussions **33**

A Related Work

A.1 Cooperative Multi-Agent Reinforcement Learning

Cooperative MARL has found extensive applications across diverse domains, including power control, robotic fleet management, and ride-hailing systems Yuan et al. (2023). As highlighted in a comprehensive survey Oroojlooy & Hajinezhad (2023), existing methods can be taxonomically classified into five major categories: independent learning, centralized critic, value decomposition, consensus-based, and communication-based approaches.

Early research predominantly focused on independent learners, which represent the most straightforward adaptation of single-agent RL to multi-agent settings. By treating each agent as an independent entity and considering others as part of the environment, standard RL algorithms can be readily applied. However, this paradigm suffers from fundamental limitations: the environment becomes inherently non-stationary due to concurrently learning peers, and agents tend to converge to local optima by maximizing individual rewards while neglecting global cooperation. Although subsequent improvements such as Hysteretic Q-Learning Matignon et al. (2007) were proposed to mitigate these issues, independent learning still struggles in large-scale systems and sparse-reward scenarios.

To address these challenges, the Centralized Training with Decentralized Execution (CTDE) and Centralized Training with Centralized Execution (CTCE) paradigms have been widely adopted in centralized critic and value decomposition methods Oroojlooy & Hajinezhad (2023). In centralized critic approaches, actor-critic algorithms such as PPO, SAC, and DDPG are extended by replacing the critic with a centralized counterpart that observes global information during training, thereby stabilizing the learning process while preserving decentralized execution. To further enhance representational capacity, attention mechanisms inspired by the Transformer architecture Vaswani et al. (2017) have been incorporated to model inter-agent relationships Mao et al. (2018). Foerster et al. Foerster et al. (2018) proposed Counterfactual Multi-Agent (COMA) policy gradients to resolve the credit assignment problem through counterfactual baselines, an idea subsequently refined in works such as Li et al. (2021). Nevertheless, many of these methods encounter the Curse of Dimensionality (CoD) as the number of agents scales up.

Value decomposition methods, by contrast, focus on factorizing the global reward into individual credit assignments, enabling agents to optimize collective objectives rather than selfish returns. The seminal Value Decomposition (VD) network Sunehag et al. (2018) pioneered this direction but suffered from the “lazy agent" problem due to its simplistic additive Q-value factorization. Subsequent advances have significantly improved representational capacity through more expressive mixing architectures, including QMIX Rashid et al. (2020) which enforces monotonicity constraints, Value-Decomposition Actor-Critic (VDAC) Su et al. (2021), and QTRAN Son et al. (2019) which lifts monotonicity restrictions. However, most of these approaches require a mixing network to map individual Q-values to the global Q-value, and even with hyper-networks for parameter generation, they still grapple with scalability as agent populations grow Hao & Varakantham (2022).

Consensus and communication methods emerged later as a means to balance cooperation efficiency against the CoD challenge. In these paradigms, agents exchange information only with neighbors or selected peers rather than broadcasting globally. Consensus-based approaches leverage sparse communication to achieve policy alignment among agents, often with convergence guarantees under linear function approximation Varshavskaya et al. (2009). However, many such methods require multiple communication rounds, rendering them impractical for real-time applications like ride-sharing where low latency is paramount. Communication-based methods instead focus on designing efficient mechanisms for determining what information to share and with whom. CommNet Sukhbaatar et al. (2016) pioneered this direction by broadcasting each agent’s hidden features derived from local observations. Yet, similar to Mean Field MARL approaches Yang et al. (2018), CommNet considers only averaged influences, overlooking fine-grained inter-agent relationships. Subsequent attention-based methods were introduced to weigh the importance of information from different sources Das et al. (2019); Li & Zhang (2024). However, communication-based methods face nontrivial training difficulties, particularly in early stages when communicated messages carry limited meaningful information. Furthermore, they often involve inherent trade-offs between cooperation efficacy, communication overhead, and message content—issues intimately tied to CoD.

A.2 TRANSFORMER IN REINFORCEMENT LEARNING

Inspired by the success of the Transformer Vaswani et al. (2017) in Large Language Models (LLMs), particularly its scalability, strong generalization, and ability to capture long-range dependencies, the architecture has been widely adopted in fields such as computer vision Dosovitskiy et al. (2021), signal processing Zhao et al. (2026), and reinforcement learning Wen et al. (2022). According to Hu et al. (2024a), Transformer-based methods in RL can be broadly categorized into three main applications: (i) architecture enhancement, where Transformers serve as more powerful backbones to improve policy or model capacity; (ii) offline RL, where they learn from sequential trajectory data; and (iii) online RL, where they are integrated to enrich the learning paradigm.

In the context of architecture enhancement, most works leverage the Transformer's ability to model long-term temporal dependencies, particularly in Partially Observable Markov Decision Processes (POMDPs), where solutions previously relied on recurrent architectures such as DRQN Hausknecht & Stone (2015). For example, the Gated Transformer-XL (GTrXL) Parisotto et al. (2020) introduces a gating mechanism with pathway skip connections, achieving improved feature extraction from historical trajectories. Another line of research employs Transformers for environment modeling in model-based RL, capitalizing on their strong sequence prediction capabilities. TransDreamer Chen et al. (2021a), for instance, integrates Transformers into the Dreamer framework to construct a stochastic world model, outperforming conventional RNN-based counterparts Hafner et al. (2020).

For offline RL, the most prominent methods are the Trajectory Transformer (TT) Janner et al. (2021) and the Decision Transformer (DT) Chen et al. (2021b). TT models each feature of state, action, and reward as separate tokens and formulates behavior cloning as a next-token prediction task. In contrast, DT treats each state-action-reward triple as a single entity and frames the problem as a reward-to-go (RTG) guided sequence prediction task. These two works have profoundly influenced subsequent research. For instance, Online DT (ODT) Zheng et al. (2022) extends DT to an offline pre-training and online fine-tuning paradigm, addressing the distribution shift between offline trajectories and online interactions. Q-Learning DT (QDT) Yamagata et al. (2023) further refines dataset quality by relabeling RTGs.

Recently, the success of LLMs has prompted researchers to explore how their generalization properties, such as few-shot learning and rich representations, can benefit RL. Building on transfer learning theory Pan & Yang (2009), several studies directly leverage pre-trained Transformers to initialize RL policies Reid et al. (2022); Li et al. (2022). Others introduce masked prediction tasks during offline RL pre-training to enhance feature extraction Liu et al. (2022); Wu et al. (2023). Zhao & Li (2026) proposes a multi-agent pre-training method to improve single-agent RL under data scarcity. Inspired by the in-context learning capabilities of LLMs, Xu et al. (2022); Lin et al. (2022) investigate prompting-based methods that guide policies by conditioning on expert demonstrations. Moreover, Lee et al. (2022); Reed et al. (2022) train policies across multiple tasks in a multi-task setting, achieving generalization through direct exposure to diverse environments.

The aforementioned methods predominantly operate in offline or offline-to-online settings, as Transformers in single-agent RL are primarily used to model historical trajectories. In contrast, in MARL, Transformers are well-suited for capturing inter-agent relationships, leading to a surge of online MARL methods. The mat (MAT) Wen et al. (2022) pioneered this direction by employing an encoder to model agent interactions and a decoder for sequential decision-making. Subsequent works have built upon MAT in various ways. For example, Takayama & Fujita (2025); Hu et al. (2025) jointly optimize action selection and decision order, highlighting the significant impact of ordering on performance. CommFormer Hu et al. (2024b) integrates graph attention networks to enable communication-efficient decentralized execution in a centralized training paradigm. Other approaches, such as MaskMA Liu et al. (2024b) and UPDeT Hu et al. (2021), focus on learning general action representations with Transformers for agent interaction modeling. However, they still require task-specific architectural design, which limits their practical applicability.

B VALUE FUNCTION DEFINITION

The standard value functions are defined as follows:

$$\begin{aligned} \mathrm{V}_\pi(\mathcal{O}_t) &= \mathbf{E}_\pi \left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} \mathrm{R}(\mathcal{O}_\tau, \mathcal{A}_\tau) \mid \mathcal{O}_t \right] , \\ \mathrm{Q}_\pi(\mathcal{O}_t, \mathcal{A}_t) &= \mathbf{E}_\pi \left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} \mathrm{R}(\mathcal{O}_\tau, \mathcal{A}_\tau) \mid \mathcal{O}_t, \mathcal{A}_t \right] , \\ \mathrm{A}_\pi(\mathcal{O}_t, \mathcal{A}_t) &= \mathrm{Q}_\pi(\mathcal{O}_t, \mathcal{A}_t) - \mathrm{V}_\pi(\mathcal{O}_t) , \end{aligned} \tag{12}$$

where $\mathrm{V}_\pi(\cdot)$, $\mathrm{Q}_\pi(\cdot,\cdot)$, and $\mathrm{A}_\pi(\cdot,\cdot)$ denote the observation value function, the observation-action value function, and the advantage function, respectively.

We further define the Q-value function for a specific agent set ψ as Wen et al. (2022); Kuba et al. (2022); Zhong et al. (2024):

$$\mathrm{Q}_\pi^\psi(\mathcal{O}_t, a_t^\psi) = \mathbf{E}_{\hat{a}_t^{-\psi} \sim \pi} \left[\mathrm{Q}_\pi(\mathcal{O}_t, [a_t^\psi, \hat{a}_t^{-\psi}]) \right] , \tag{13}$$

where $-\psi$ denotes the complement of ψ. Based on this definition, the advantage function for ψ given the actions of Ψ is defined as

$$\mathrm{A}_\pi^\psi(\mathcal{O}_t, a_t^\Psi, a_t^\psi) = \mathrm{Q}_\pi^{\psi \cup \Psi}(\mathcal{O}_t, [a_t^\psi, a_t^\Psi]) - \mathrm{Q}_\pi^\psi(\mathcal{O}_t, a_t^\psi) , \tag{14}$$

where ψ and Ψ are disjoint sets. For notational simplicity, we omit the policy symbol π in the following sections whenever no ambiguity arises.

C ALGORITHM

The detailed training procedure of the proposed CMAT is presented in Algorithm 1. Furthermore, we illustrate the conceptual distinction between our method and conventional decentralized MARL approaches in Fig. 4, which highlights the key motivation behind our design.

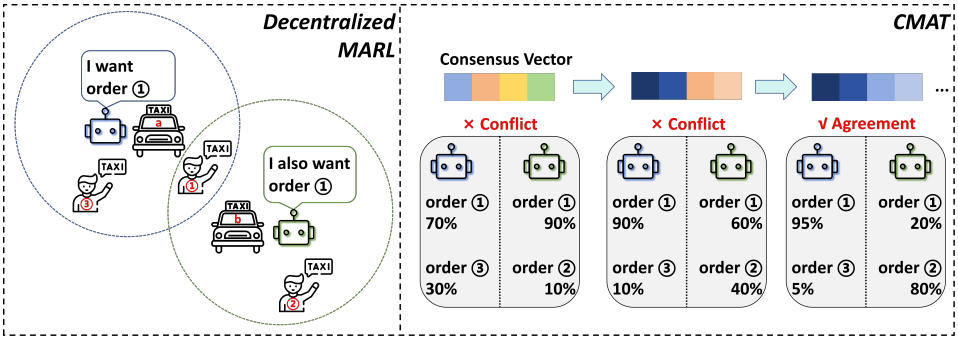


Figure 4: Comparison between CMAT and Conventional Decentralized MARL Methods.

D EXPERIMENT SUPPLEMENT

D.1 EXPERIMENT CONFIGURATIONS

The compared methods include:

- **MAT Wen et al. (2022):** The first centralized observation with a sequential decision-making MARL framework. More details are presented in Section 2.2.

Algorithm 1 Consensus Multi-Agent Transformer (CMAT)

Require: Number of agents n, consensus iterations m, PPO hyper-parameters ϵ, γ, GAE parameter λ, total training steps T_{total}
Ensure: Trained policy network θ and critic network ϕ

Initialize network parameters θ, ϕ, and target critic $\phi^- \leftarrow \phi$
Procedure ActionSelection$(\mathcal{O})$ ▷ Used for experience collection
  $\hat{\mathcal{O}} \leftarrow \text{Encoder}(\mathcal{O})$ ▷ order-independent
  $e^0 \leftarrow \text{CriticCompressor}(\hat{\mathcal{O}})$
  $\hat{V}(\mathcal{O}) \leftarrow \text{CriticMLP}(e^0)$
  $\mathcal{E} \leftarrow \{e^0\}$
for $k = 1$ **to** m **do**
    $e^k \leftarrow \text{Decoder}(e^{k-1}, k)$ ▷ auto-regressive with positional index k
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{e^k\}$
end for
  $c \leftarrow \text{ActorCompressor}(\mathcal{E})$
for $i = 1$ **to** n **do**
    $a^i \sim \pi^i_\theta(\cdot \mid \hat{o}^i, c)$ ▷ ActorMLP($[\hat{o}^i; c]$)
end for
  return $\mathcal{A} = \{a^1, \ldots, a^n\}$, $\hat{V}(\mathcal{O})$

A.Training
while training steps $< T_{\text{total}}$ **do**
  Collect trajectory $\tau = \{(\mathcal{O}_t, \mathcal{A}_t, R_t, \mathcal{O}_{t+1})\}$ by calling **ActionSelection**$(\mathcal{O}_t)$
  Compute advantages $\hat{A}_t$ using GAE with $\hat{V}_{\phi^-}$ and λ
  Compute critic loss:

$$\mathcal{L}_{\text{critic}}(\phi) \leftarrow \mathbb{E}_t \left[(R_t + \gamma \hat{V}_{\phi^-}(\mathcal{O}_{t+1}) - \hat{V}_\phi(\mathcal{O}_t))^2 \right]$$

  Compute importance ratio:

$$\mathcal{R}_t(\theta) \leftarrow \frac{\prod_{j=1}^n \pi^j_\theta(a^j_t \mid \hat{o}^j_t, c_\theta)}{\prod_{j=1}^n \pi^j_{\theta^-}(a^j_t \mid \hat{o}^j_t, c_{\theta^-})}$$

  Compute actor loss:

$$\mathcal{L}_{\text{actor}}(\theta) \leftarrow \mathbb{E}_t \left[\min\Big(\mathcal{R}_t(\theta)\hat{A}_t, \text{CLIP}(\mathcal{R}_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \Big) \right]$$

  Update θ and ϕ by minimizing $\mathcal{L}_{\text{actor}} + \mathcal{L}_{\text{critic}}$
  Soft-update target network: $\phi^- \leftarrow \tau\phi + (1-\tau)\phi^-$
end while

B. Fine-tuning ▷ Continue from current θ, ϕ
if Consensus Enhancement **then**
  Freeze all Actor-MLP layers
  Continue training by updating only Critic-MLP, Decoder, and Actor-Compressor
else if Action Policy Enhancement **then**
  Freeze Encoder, Decoder, Critic-Compressor, and Actor-Compressor
  Continue training by updating only Critic-MLP and all Actor-MLP layers
end if
return Trained policy network θ and critic network ϕ

- **PMAT Hu et al. (2025):** Based on MAT, PMAT introduces an additional module to determine agent action order, grounded in the theory of Plackett-Luce sampling Luce et al. (1959); Plackett (1975). AOAD-MAT Takayama & Fujita (2025), a synchronous work, employs a similar method to PMAT but decides action order followed by detailed action decision, while AOAD-MAT does both simultaneously. Given the high similarity between them and the fact that only PMAT provides official code, we select PMAT as the benchmark here.
- **Triple-BERT Zhao & Li (2026):** Triple-BERT is the first centralized SARL framework for the ride-sharing task, utilizing BERT for observation feature and relationship extraction and processing a large action space through an action decomposition mechanism.
- **HAPPO Kuba et al. (2022); Zhong et al. (2024):** HAPPO first illustrates and proves the efficiency of sequential optimization among agents in cooperative MARL scenarios, providing a strong foundation for MAT. Based on Zhong et al. (2024), HAPPO serves as the SOTA method in the family of Heterogeneous-Agent Reinforcement Learning (HARL) methods.
- **MAPPO Yu et al. (2022):** MAPPO is a popular and strong CTDE baseline in cooperative MARL, utilizing a centralized critic during training to provide direction for actors with global information, while actors operate independently during execution.

For all evaluated scenarios, the model configurations follow those established in MAT Wen et al. (2022), with the exception of setting the rollout threads to 8 due to hardware limitations. These configurations are also adopted for comparative methods unless their original papers or official repositories specify alternative setups. Our implementation is built upon the official MAT repository, available at `https://github.com/PKU-MARL/Multi-Agent-Transformer`.

To keep the comparison controlled, every method is retrained by us under one common configuration rather than transcribed from its original paper, so the reported numbers reflect a single set of random seeds and software versions. We therefore fix the environment versions listed below, and they apply uniformly to all comparative methods:

- **StarCraft II**: PySC2 version 4.0.0, SMAC version 1.0.0, with underlying StarCraft II game version Base55958.
- **Multi-Agent MuJoCo**: MuJoCo version 3.4.0, PettingZoo version 1.25.0. The evaluated robotic environments include HalfCheetah-v2, Ant-v2, and Walker2D-v2.
- **Google Research Football**: GFootball version 2.10.2.

All models were trained using the PyTorch framework Paszke et al. (2019) on a workstation running Windows 11, equipped with an Intel(R) Core(TM) i7-14700KF processor and an NVIDIA RTX 4080 graphics card. The GPU occupation is around 0.5-1.5 GB during the whole training process.

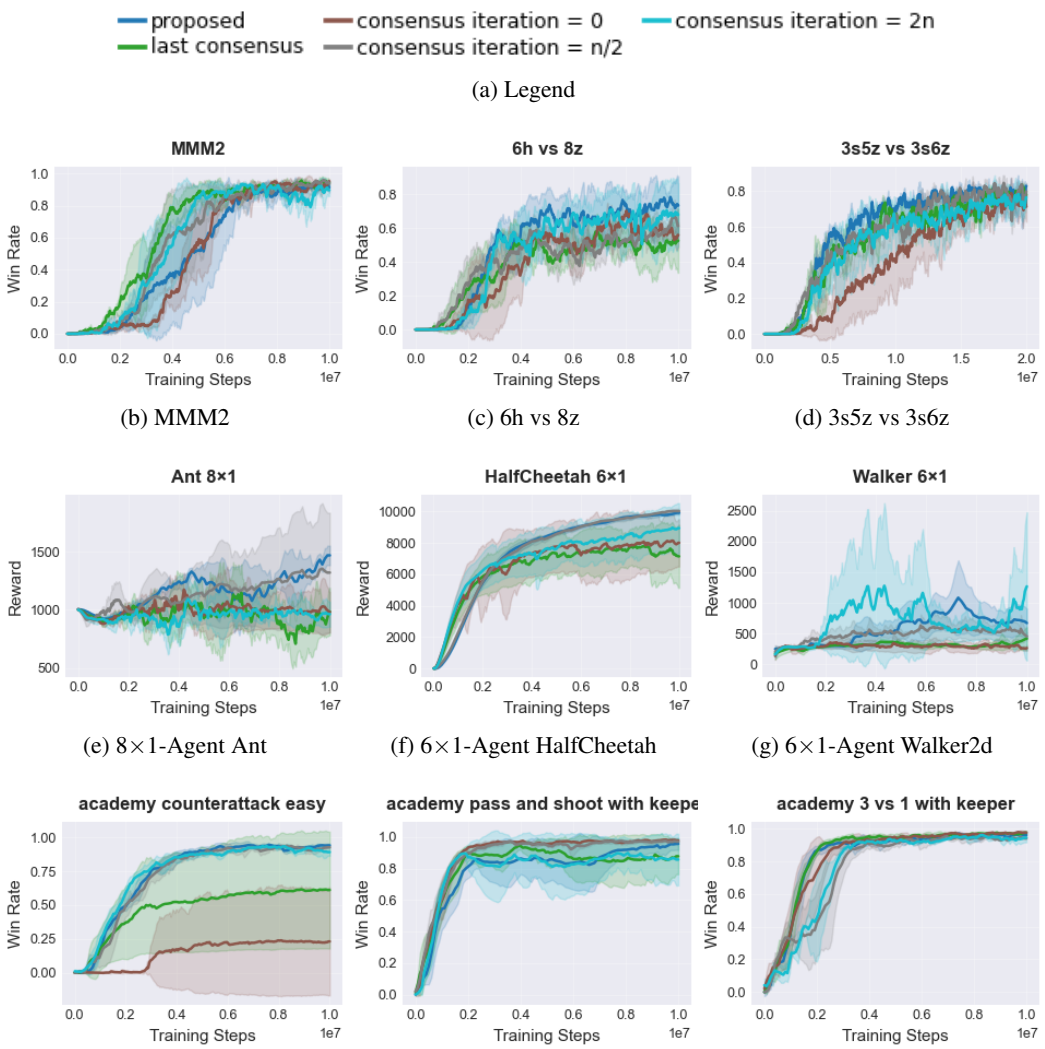
Rollout threads are set to 8 rather than the number recommended in MAT Wen et al. (2022), which the host machine cannot sustain. The setting is identical for every method and every seed, so it defines the common operating point at which the comparison is made.

D.2 Ablation Study and Sensitivity Analysis

To further illustrate the efficacy of each proposed module, we conduct a series of ablation studies and sensitivity analyses of hyper-parameters, including:

- **Consensus Mixture Versus Last Consensus:** We propose the decoder (actor) compressor to mix the middle-generated consensus $\mathcal{E}$ to avoid information loss during the auto-regressive decoding process. Here, we compare the performance of our mixture method against the direct use of the last generated consensus vector e^m.
- **Impact of Consensus Iteration Times:** In our method, there is only one newly introduced hyperparameter compared to CMAT, namely the decoder iteration times m. We default this to the number of agents n, with the intuition that in the worst-case scenario, CMAT can degrade to MAT. Here, we additionally compare the performance under $m = 0, \lfloor \frac{n}{2} \rfloor, 2n$.

The experimental results are shown in Fig. 5. We observe that when directly using the last consensus instead of mixing all consensus outputs via our Actor-Compressor, the model performance decreases in most scenarios, suggesting that the decoder auto-regression process may lose some useful information from previous generations. This confirms the value of the Actor-Compressor as a



(a) Legend

(b) MMM2 (c) 6h vs 8z (d) 3s5z vs 3s6z

(e) 8×1-Agent Ant (f) 6×1-Agent HalfCheetah (g) 6×1-Agent Walker2d

(h) academy counterattack easy (i) academy pass and shoot with keeper (j) academy 3 vs 1 with keeper

Figure 5: Ablation Study under 5 Random Seeds

mechanism for aggregating the full trajectory of consensus vectors rather than relying on the final one. Regarding the consensus iteration time, we find that selecting n (the number of agents) as the iteration count is a proper choice: too few iterations may be insufficient to generate a good consensus among agents, while too many iterations may introduce excessive noise and redundant information, thereby increasing the training difficulty of the Actor-Compressor. Intuitively, choosing the number of iterations equal to n aligns with the decoding process of MAT, allowing each agent to sufficiently adjust its action in response to the actions of others.

E WHY THE CMAT ARCHITECTURE IS EFFECTIVE

This appendix gives a rigorous, self-contained account of why the *deterministic*-consensus architecture that we actually deploy is effective. In contrast to the more general analysis of Appendix F, which is stated for a stochastic consensus policy and serves mainly to motivate future extensions, the results here make no probabilistic-coordination and no tabular-convergence assumptions. They are statements about the representational structure induced by conditioning every action head on a single deterministic consensus vector $c = \mu_\theta(\mathcal{O})$, and about the iterative refinement that produces

c. Throughout, we separate what is a property of the architecture itself from what reflects added capacity. The central thesis of this appendix is one of *representational efficiency*: iterative refinement with a single reused block can express the same coordination computation that an encoder-only controller can express only by stacking, and therefore by growing, its parameters. Every result here is accordingly a statement about what the network structure is able to represent at a given parameter budget, so the quantity each one characterizes is representational capacity, and trainability is established by our experiments.

Scope of the theory: representation, not optimization. We state the boundary of these results explicitly. Every statement in this appendix is about *representational capacity*: what the deterministic-consensus decoding structure can express, how well its best-case expressible policy approximates the optimum (Theorems 1–2), why iterative refinement is the right way to represent a self-consistent consensus (Theorem 3, Proposition 1), and why weight-shared iteration represents the same coordination computation as depth at a smaller parameter budget (Theorem 4). *Optimization*, whether a particular training procedure on our finite network finds the good representation the capacity results permit, is a separate question that our empirical study settles. The results are therefore structural justifications for the architecture, made precise, and they carry their predictive weight jointly with the experiments of Section 4.

We address the two questions that are specific to CMAT. The mitigation of the action-space Curse of Dimensionality follows from any attention-based centralized controller, and it appears below as the scaffold on which the two CMAT-specific arguments rest. The two questions are: (i) why the consensus must be produced by *iterative* refinement rather than a single aggregation such as mean-pooling; and (ii) why conditioning on a deterministic consensus is more effective than an encoder-only controller with independent action heads at a comparable parameter budget.

E.1 Setup: Consensus-Separable Team Value Functions

Recall the single-agent multi-action MDP of the Preliminaries, with team action $\mathcal{A} = (a^1, \dots, a^n) \in \bar{\mathcal{A}} = \prod_i \mathcal{A}^i$ and optimal team action-value $\mathrm{Q}^\star(\mathcal{O}, \mathcal{A})$. The action-space Curse of Dimensionality is the statement that the greedy step $\max_{\mathcal{A} \in \bar{\mathcal{A}}} \mathrm{Q}^\star(\mathcal{O}, \mathcal{A})$ ranges over $\prod_i |\mathcal{A}^i|$ joint actions, exponential in n. The CMAT decoder produces each action from its own observation embedding $\hat{o}^i$ and the shared consensus c through Eq. (9). The representational premise that makes this structure exact is that the team value is additively separable once conditioned on a low-dimensional coordination statistic.

Definition 1 (d-consensus-separability)**.** A team value function $\mathrm{Q}(\mathcal{O}, \mathcal{A})$ is *d-consensus-separable* if there exist a coordination map $f : \mathcal{O} \to \mathcal{C} \subseteq \mathbb{R}^d$ and local value functions q^i such that

$$\mathrm{Q}(\mathcal{O}, \mathcal{A}) = \sum_{i=1}^{n} q^i(\hat{o}^i, a^i, c), \qquad c = f(\mathcal{O}), \quad \forall\, \mathcal{O}, \mathcal{A}. \tag{15}$$

For a general Q, the best d-dimensional consensus-separable approximation error is

$$\varepsilon_d(\mathrm{Q}) = \inf_{f, \{q^i\}} \sup_{\mathcal{O}, \mathcal{A}} \Big| \mathrm{Q}(\mathcal{O}, \mathcal{A}) - \sum_{i=1}^{n} q^i(\hat{o}^i, a^i, f(\mathcal{O})) \Big|. \tag{16}$$

This is the learned, nonlinear, state-dependent analogue of the coupling signals used in classical operations-research decompositions of large-scale sequential decision problems: the Lagrangian (dual price) in weakly coupled MDPs and restless bandits (Whittle, 1988; Adelman & Mersereau, 2008), and the population mean field in mean-field control (Carmona et al., 2018). In those methods a shared signal, a price λ or a first moment, renders the n per-component problems conditionally independent. Here $c = f(\mathcal{O})$ plays the same decoupling role but is computed by attention on the true state and is not restricted to be linear or to a first moment. The quantity ε_d in Eq. (16) is a deterministic, distribution-free measure of how much coordination the problem intrinsically requires, and it is non-increasing in d.

E.2 Collapse of the Exponential Joint Maximization

Theorem 1 (Argmax collapse)**.** *If* Q *is* d*-consensus-separable (Def. 1), then for every* $\mathcal{O}$*,*

$$\max_{\mathcal{A}\in\bar{\mathcal{A}}} \mathrm{Q}(\mathcal{O},\mathcal{A}) = \sum_{i=1}^{n} \max_{a^i\in\mathcal{A}^i} q^i(\hat{o}^i, a^i, f(\mathcal{O})), \tag{17}$$

and $\mathcal{A}^\star(\mathcal{O}) = \big(\arg\max_{a^i} q^i(\hat{o}^i, a^i, f(\mathcal{O}))\big)_{i=1}^{n}$ *is a joint maximizer. Hence the greedy team action is obtained from one evaluation of the consensus* $c = f(\mathcal{O})$ *followed by* n *independent local maximizations, at cost* $O\big(\sum_i |\mathcal{A}^i|\big)$ *rather than* $O\big(\prod_i |\mathcal{A}^i|\big)$*.*

Proof. Fix $\mathcal{O}$ and set $c = f(\mathcal{O})$. The objective $\sum_i q^i(\hat{o}^i, a^i, c)$ is a sum whose i-th term depends on the decision variable a^i only, since the embeddings $\hat{o}^i$ and the shared c are determined once $\mathcal{O}$ is fixed. Maximization of a separable sum over a product domain decomposes coordinatewise, $\max_{\mathcal{A}} \sum_i q^i = \sum_i \max_{a^i} q^i$, attained at the tuple of coordinatewise maximizers. Each inner maximization enumerates $|\mathcal{A}^i|$ actions, giving the stated cost. □

Remark 1 (Exploiting structure, not defeating the curse). Theorem 1 does not eliminate the Curse of Dimensionality unconditionally. When $\varepsilon_d(\mathrm{Q}^\star) > 0$ the collapse is only approximate (Theorem 2). The precise claim is that CMAT *exploits* two structural properties, permutation symmetry across agents and a low-dimensional coupling channel, to trade an exponential joint maximization for a linear one, exactly the trade made by the classical decomposition methods above. CMAT's contribution is to *learn* the coupling statistic c rather than hand-design it.

E.3 Coordination Dimension versus Suboptimality

Theorem 2 (Suboptimality of consensus decoding, conditional on realizability)**.** (Realizability premise.) *Assume the optimal team value admits a* d*-consensus-separable approximation with error* $\varepsilon_d(\mathrm{Q}^\star)$*, and that the CMAT decoder actually realizes such a best approximation, that is, there exist a coordination map* f *and heads* q^i *within the decoding structure attaining the infimum in Eq.* (16)*.* (Conclusion.) *Let* $\mathrm{Q}^\star$ *be the optimal team action-value and let* $\widehat{\mathrm{Q}}(\mathcal{O},\mathcal{A}) = \sum_i q^i(\hat{o}^i, a^i, f(\mathcal{O}))$ *be this approximation, so that* $\|\widehat{\mathrm{Q}} - \mathrm{Q}^\star\|_\infty = \varepsilon_d(\mathrm{Q}^\star)$*. Let* $\hat{\pi}$ *be the deterministic policy that plays the greedy team action of* $\widehat{\mathrm{Q}}$*, that is, the collapse of Theorem 1. Then for every state,*

$$V^{\hat{\pi}}(\mathcal{O}) \;\geq\; V^\star(\mathcal{O}) - \frac{2\,\varepsilon_d(\mathrm{Q}^\star)}{1-\gamma}. \tag{18}$$

Proof. By Theorem 1, $\hat{\pi}$ selects $\arg\max_{\mathcal{A}} \widehat{\mathrm{Q}}(\mathcal{O},\mathcal{A})$, so it is greedy with respect to $\widehat{\mathrm{Q}}$. The classical bound on greedy policies of approximate action-value functions (Singh & Yee, 1994; Bertsekas, 2025) states that $\|\widehat{\mathrm{Q}} - \mathrm{Q}^\star\|_\infty \leq \varepsilon$ implies the greedy policy of $\widehat{\mathrm{Q}}$ is $\frac{2\varepsilon}{1-\gamma}$-suboptimal. Taking $\varepsilon = \varepsilon_d(\mathrm{Q}^\star)$ gives the claim. □

The greedy policy $\hat{\pi}$ is representable within the CMAT decoding structure, since each action head is a function of $(\hat{o}^i, c)$ alone. Because $\varepsilon_d(\mathrm{Q}^\star)$ is non-increasing in d and vanishes once d reaches the intrinsic coordination dimension of the problem, Theorem 2 both certifies the deterministic architecture (small ε_d implies near-optimal decoding) and gives quantitative guidance for choosing the consensus width d. It also pinpoints where CMAT should excel. In large-scale operational systems such as matching, queueing, fleet and capacity allocation, and congestion pricing, inter-agent coupling is transmitted mainly through low-dimensional aggregates such as aggregate supply-demand, prices, or congestion levels, so ε_d decays quickly in d and a modest consensus suffices even for large n. We stress that this is a conditional, "good representation implies good policy" result: it presumes the realizability premise, that the learned f and q^i reach the best d-consensus-separable fit with error $\varepsilon_d(\mathrm{Q}^\star)$. It is a statement about the representational capacity of the decoding structure, not a claim that PPO drives the realized approximation error down to $\varepsilon_d(\mathrm{Q}^\star)$; whether training actually attains this fit is an empirical question, addressed by our experiments rather than by this bound.

E.4 Why Iterate: Consensus as Unrolled Coordination Optimization

The consensus that supports Theorem 1 cannot be an arbitrary one-shot summary of $\mathcal{O}$. The local maximizers $\hat{a}^i(c) = \arg\max_{a^i} q^i(\hat{o}^i, a^i, c)$ depend on c, so for the decomposition to be self-consistent the coordination statistic must agree with the actions it induces. A valid consensus is therefore a fixed point

$$c^\star = \Phi\big(\mathcal{O}, (\hat{a}^i(c^\star))_{i=1}^n\big) =: \mathcal{T}_{\mathcal{O}}(c^\star), \tag{19}$$

where Φ re-aggregates the induced per-agent responses into the coordination statistic. This is the self-consistency that defines equilibrium aggregates in mean-field and weakly coupled formulations. A single feed-forward aggregation, whether the initial vector e^0 or any mean-pooling, computes $\mathcal{T}_{\mathcal{O}}(c_0)$ once from an uninformed c_0 and does not solve Eq. (19).

We therefore model the decoder as an unrolled optimizer. We adopt this view rather than a contraction argument because it does not rely on the contraction constant of attention, which is not automatically below one (Remark 2). We state the view as an explicit modeling assumption rather than a derived property, and the direction of the claim is fixed: the decoder reads as unrolling a coordination solver, not as a trained attention block provably performing exact gradient ascent on a specific potential.

Assumption 1 (Unrolled-optimization view of the decoder)**.** We model the refinement as follows. We posit a coordination potential $J(\cdot\,; \mathcal{O}) : \mathbb{R}^d \to \mathbb{R}$, bounded above and L-smooth in c, whose stationary points are the self-consistent consensus vectors of Eq. (19), and we treat one decoder refinement step as implementing a gradient (or proximal) ascent step $c_{k+1} = c_k + \eta_k \nabla_c J(c_k; \mathcal{O})$ with $\eta_k \le 1/L$. This is an interpretive assumption about the computation the block is designed and trained to carry out, not a proven identity between the softmax-attention map and $\nabla_c J$.

This is the standard modeling premise of the algorithm-unrolling, learning-to-optimize, and deep-equilibrium literatures (Gregor & LeCun, 2010; Marino et al., 2018; Bai et al., 2019), in which a recurrent, weight-shared block is designed by unrolling the iterations of a solver for an implicitly defined objective and is then trained end to end. Following that tradition, we invoke the assumption in the forward direction, to motivate why iterating helps and to predict the qualitative behaviour that follows (monotone-then-saturating improvement, Theorem 3), and we take the saturation reported in Appendix D.2 as the empirical confirmation that the modeled behaviour obtains.

Theorem 3 (Monotone improvement of iteration; one-shot is insufficient at a fixed width)**.** *Under Assumption 1, the decoder iterates satisfy*

$$J(c_{k+1}; \mathcal{O}) \;\ge\; J(c_k; \mathcal{O}) + \tfrac{\eta_k}{2} \|\nabla_c J(c_k; \mathcal{O})\|^2, \tag{20}$$

so $J(c_k; \mathcal{O})$ is non-decreasing in k and converges, and for constant step size $\min_{k \le m} \|\nabla_c J(c_k; \mathcal{O})\|^2 = O(1/m)$. In particular the one-shot consensus e^0 corresponds to $k = 1$ and is generally not stationary, incurring a strictly positive gap $J^\star - J(e^0; \mathcal{O}) > 0$ whenever $\nabla_c J(c_0; \mathcal{O}) \neq 0$; this gap is reduced monotonically by further iterations.

Proof. For an L-smooth J and a step $c_{k+1} = c_k + \eta_k \nabla_c J(c_k)$ with $\eta_k \le 1/L$, the descent lemma in ascent form gives $J(c_{k+1}) \ge J(c_k) + \eta_k(1 - \frac{L\eta_k}{2})\|\nabla_c J(c_k)\|^2 \ge J(c_k) + \frac{\eta_k}{2}\|\nabla_c J(c_k)\|^2$ (Nesterov et al., 2018). Hence $J(c_k; \mathcal{O})$ is non-decreasing and, being bounded above, converges. Summing the inequality over $k \le m$ and dividing by $\sum_k \eta_k$ bounds $\min_{k \le m} \|\nabla_c J(c_k)\|^2$ by $2(J^\star - J(c_0))/(\sum_k \eta_k) = O(1/m)$ for constant η_k. Since $e^0 = c_1$ is a single ascent step from an uninformed c_0, it is stationary only if $\nabla_c J(c_0) = 0$, which fails generically. □

This matches the behaviour observed empirically in Appendix D.2: replacing the iterated consensus by e^0 (no iteration) degrades performance, and increasing the number of iterations m yields diminishing returns that saturate, consistent with the non-decreasing and converging sequence $J(c_k; \mathcal{O})$.

Remark 2 (Contraction alternative). If $\mathcal{T}_{\mathcal{O}}$ is a contraction of modulus $\rho < 1$, then it has a unique fixed point $c^\star$ and $\|c_k - c^\star\| \le \rho^k \|c_0 - c^\star\|$, a geometric convergence that likewise predicts the empirical saturation. We do not rely on this route in the main statement because a standard softmax attention block is not guaranteed to be a contraction; its Lipschitz constant can exceed one (Kim et al., 2021). Enforcing contraction requires spectral or normalization constraints, or restricting attention to a neighborhood of $c^\star$ (local contraction). The unrolled-optimization view of Theorem 3 avoids this dependence.

Proposition 1 (Mean-pooling captures only a first-order moment)**.** *A single mean-pooling aggregation produces $c = \psi\big(\frac{1}{n}\sum_i \phi(\hat{o}^i)\big)$, a function of the empirical first moment of the embeddings, which is precisely the mean-field statistic and is exact only in the exchangeable, weak-coupling, large-n limit (Carmona et al., 2018). If the self-consistent consensus $c^\star$ of Eq.* (19) *depends on higher-order or pairwise structure of $\{\hat{o}^i\}$, then at any fixed embedding width mean-pooling cannot represent $c^\star$, whereas the data-dependent weighted aggregation of attention, iterated, can. Quantitatively, exact representation of permutation-invariant set functions by sum or mean pooling requires latent width at least the set cardinality (Wagstaff et al., 2019), a blow-up that iterated cross-attention avoids.*

Proof. Write the aggregation as $c = \psi(\mu)$ with $\mu = \frac{1}{n}\sum_i \phi(\hat{o}^i)$. The map factors through μ, so it is constant on every level set $\{\{\hat{o}^i\} : \frac{1}{n}\sum_i \phi(\hat{o}^i) = \mu\}$. Take two configurations $\{\hat{o}^i\}$ and $\{\tilde{o}^i\}$ with equal first moment, $\frac{1}{n}\sum_i \phi(\hat{o}^i) = \frac{1}{n}\sum_i \phi(\tilde{o}^i)$, but differing in some higher-order statistic; both are mapped to the same $c = \psi(\mu)$. If the self-consistent consensus $c^\star$ of Eq. (19) takes distinct values on these two configurations, then no function of μ can equal $c^\star$ on both, so mean-pooling cannot represent $c^\star$; this holds at every fixed width because the factorization through μ is width-independent. For the quantitative claim, Wagstaff et al. (2019) show that a sum- or mean-decomposition $\psi(\sum_i \phi(\hat{o}^i))$ represents all continuous permutation-invariant functions of an n-element set exactly only if the latent dimension of ϕ is at least n; below this width some set functions are unrepresentable. Iterated cross-attention is not of this fixed-moment form: each step reweights the tokens by data-dependent, non-uniform attention scores and composes these reweightings across steps, so it realizes functions that depend on pairwise and higher-order structure of $\{\hat{o}^i\}$ without a latent width that grows with n. □

E.5 Why Consensus Beats an Encoder-Only Controller at a Fixed Budget

We start from the comparison that any reader will make. Under full observability the encoder's self-attention already makes every embedding $\hat{o}^i$ a function of the entire $\mathcal{O}$, so with unbounded depth and width an encoder-only controller with independent heads is a universal approximator of the optimal deterministic map $\mathcal{A}^\star(\mathcal{O})$. The two architectures are therefore equivalent at infinite capacity, and the advantage of consensus is a fixed-budget phenomenon: it attains a given coordination accuracy with far fewer parameters, because it buys the required number of coordination rounds through weight-shared iteration rather than through depth.

We reconcile this with the previous subsection here, since the two results sit on the same axis: both are fixed-budget statements. When a single aggregation is insufficient (Proposition 1, Theorem 3), the quantifier is *at a fixed width*. With the latent width allowed to grow, a one-shot map represents the fixed point too, which is the infinite-capacity universality above. Put differently, a deep encoder-only controller does not avoid iteration: a stack of k layers performs k rounds of refinement, only with k independent sets of weights rather than one reused block. Multi-round coordination therefore happens in both architectures, and what differs is whether the rounds are paid for in parameters or in inference compute. The infinite encoder is the case of iteration realized by spending unbounded depth.

This distinction has practical, not merely asymptotic, weight in our setting. A recurring empirical finding in deep RL, and in MARL specifically, is that naively enlarging the network degrades rather than improves learning: value bootstrapping, non-stationary targets, and a moving data distribution make large models unstable and hard to train, and additional capacity often has to be actively stabilized before it helps at all (Ota et al., 2021; Björck et al., 2021; Nikishin et al., 2022). Under this regime, an architecture that reaches a given coordination accuracy with a *smaller* parameter count, by reusing one block across refinement steps rather than stacking depth, is not only parameter-efficient in the abstract but easier to optimize in practice, which is the regime our experiments operate in.

The result below rests on three premises, which we state explicitly. First, coordination is a fixed-point computation solved by repeated application of a refinement operator $\mathcal{T}_\mathcal{O}$, and it converges geometrically at some rate $\rho < 1$, so accuracy ϵ is reached after $K(\epsilon) = \Theta(\log(1/\epsilon)/\log(1/\rho))$ applications. Second, $\mathcal{T}_\mathcal{O}$ is nonlinear and state-dependent, so its k-fold composition cannot be collapsed into a single map computed at lower cost. Third, the point of comparison is an encoder that uses independent per-layer weights, which is exactly the standard Transformer encoder. Because this is a statement about representational capacity, we take the shared decoder block to have the capacity to represent one application of $\mathcal{T}_\mathcal{O}$, so that K iterations of the block represent the K-fold

composition $\mathcal{T}_{\mathcal{O}}^{(K)}$. This is a realizability premise of the same kind as elsewhere in this appendix, and it is well founded structurally: the block's input-output form is built to match one step of Eq. (19) (attend to the current c to form the induced responses, then re-aggregate them into the next c), and since $\mathcal{T}_{\mathcal{O}}$ is a *stationary* operator, applied identically at every step, a single reused block is its natural representation. We emphasize that this result does *not* invoke Assumption 1: it is stated directly in terms of the fixed-point operator $\mathcal{T}_{\mathcal{O}}$ of Eq. (19) and does not require the decoder to implement ascent on any potential J. The rate ρ is a modeling assumption on the task and operator, not a proven property of the softmax attention block, which need not be a contraction (Remark 2).

Theorem 4 (Parameter efficiency of weight-shared coordination)**.** *Under the three premises above, an encoder-only network with independent per-layer weights reaches coordination accuracy ϵ only at a cost of $\Theta(K(\epsilon))$ layers, and hence $\Theta(K(\epsilon))$ times the per-layer parameters, whereas CMAT's decoder reuses one block across refinement steps and reaches the same accuracy with a* fixed *parameter count and $K(\epsilon)$ times the inference compute. As the coordination problem stiffens ($\rho \to 1$, $K(\epsilon) \to \infty$), the encoder-only parameter count diverges while CMAT's stays constant.*

Proof. We separate the two quantities the statement involves: the number of refinement rounds $K(\epsilon)$, and the parameter cost of realizing those rounds.

Rounds. That $K(\epsilon) = \Theta(\log(1/\epsilon)/\log(1/\rho))$ is the definition of geometric convergence and is exhibited by any contractive instance. For concreteness take the linear operator $\mathcal{T}_{\mathcal{O}}(c) = Wc + b$ with $\|W\| = \rho < 1$ and fixed point $c^\star = (I - W)^{-1}b$; then $c_k - c^\star = W^k(c_0 - c^\star)$, so $\|c_k - c^\star\| = \Theta(\rho^k)$ for generic c_0, and $\rho^k \le \epsilon$ requires $k \ge \log(1/\epsilon)/\log(1/\rho) = K(\epsilon)$. This instance fixes the *rate*; it is not used to argue the parameter cost, since a linear W^k is itself a single matrix and would collapse to one layer.

Parameters. The parameter cost turns on non-compressibility, which is where the second premise enters. When $\mathcal{T}_{\mathcal{O}}$ is nonlinear and depends on $\mathcal{O}$ and on the current induced actions, its k-fold composition $\mathcal{T}_{\mathcal{O}}^{(k)}$ is a genuinely different nonlinear map for each k and cannot be pre-multiplied into a single lower-cost operator the way W^k can. An encoder-only network with independent per-layer weights can represent k rounds of such a composition only by instantiating k distinct nonlinear layers, one per round, so matching accuracy ϵ costs $\Theta(K(\epsilon))$ layers and therefore $\Theta(K(\epsilon))$ times the per-layer parameter count. A decoder that reuses a single block representing $\mathcal{T}_{\mathcal{O}}$ across steps (third premise) represents the same k-fold composition with one block of parameters applied k times, i.e. a fixed parameter count and $K(\epsilon)$ times the sequential inference compute. The reuse is legitimate precisely because $\mathcal{T}_{\mathcal{O}}$ is the same stationary map at every step, so representing $\mathcal{T}_{\mathcal{O}}^{(K)}$ requires representing only one operator, not K of them. Taking $\rho \to 1$ sends $K(\epsilon) \to \infty$, so the independent-weight parameter count diverges while the weight-shared count is unchanged. □

Were one instead to tie the encoder's layer weights, it would itself become a weight-shared iteration, which is the very ingredient we identify as useful; the comparison is therefore between depth-parameterized and iteration-parameterized coordination, not between CMAT and attention as such.

Remark 3 (No redundant replication). If every agent must condition on the same global summary $s(\mathcal{O})$, an encoder-only controller must carry $s(\mathcal{O})$ inside each of the n token streams, a width cost that scales with n, whereas consensus computes $s(\mathcal{O})$ once into c and broadcasts it. This is a representation-size argument in CMAT's favor, independent of Theorem 4.

Remark 4 (Consistency as an inductive bias). The fixed point of Eq. (19) requires all agents to decode from one self-consistent global state. Conditioning every head on a single c builds this consistency in by construction, whereas an encoder-only controller lets the n token streams evolve separately and stop interacting before the action heads (the point-wise head applied to $\hat{o}^i$ performs no further cross-agent mixing), so joint consistency must instead be learned. We state this as an optimization and inductive-bias advantage, not an impossibility: with enough capacity and data an encoder-only model can learn consistency, but the consensus structure does not have to.

Remark 5 (Design guidance for Transformer-based MARL). Theorem 4 is stated for CMAT but its content is architectural and transfers to Transformer-based cooperative MARL more broadly. It suggests three principles. First, when coordination is genuinely iterative, in the sense of requiring several rounds of mutual adjustment before a joint decision stabilizes (iterated best response, message passing, consensus formation), buying those rounds through a weight-shared recurrent block is more parameter-efficient than buying them through stacked depth, so a weight-tied refinement module

is preferable to simply deepening an encoder. Second, because the two costs are decoupled, the number of refinement steps becomes a free inference-time knob: a model can adapt its computation to the difficulty of the current state, spending more iterations on stiff coordination problems and fewer on easy ones, without changing its parameter count. Third, the stiffness ρ, or equivalently the coordination rank of Appendix F.3, is a diagnostic for when this helps at all: tasks with small $K(\epsilon)$ gain little from iteration, and the benefit of weight-shared refinement is largest precisely when coordination is rigid ($\rho \to 1$). These principles are conditional in the same way the theorem is: they presume coordination is an iterative fixed-point computation and that the refinement block represents a (at least locally) contractive operator; where a single aggregation already suffices, as in the weak-coupling, large-n regime of Proposition 1, iteration is not expected to pay off. As with the rest of this appendix, the guidance concerns representation capacity and parameter efficiency.

E.6 Synthesis

Points (i) and (ii) are the same mechanism seen from two sides. The consensus is a shared, weight-shared, iteratively refined solver for the coordination fixed point of Eq. (19). Iterating it is how the fixed point is reached (Theorem 3, Proposition 1), and reaching it by weight-shared iteration rather than by depth is why it dominates an encoder-only controller at a fixed budget (Theorem 4). The additive decomposition these arguments support is what collapses the exponential joint decision into n local ones with a controlled suboptimality (Theorems 1–2). None of these statements requires stochasticity in the consensus, tabular assumptions, or a convergence guarantee for the deep model.

F A More General Framework: Stochastic Consensus and Future Directions

The analysis in this appendix concerns a more general and more powerful version of CMAT than the one we deploy. It endows the consensus with its own stochastic policy $\pi^c(c \mid \mathcal{O})$ and casts training as a bilevel (cooperative Stackelberg) problem, which enlarges the representable policy class beyond what a deterministic consensus can reach (Appendix F.3). We present it as a future direction. The rigorous justification of the deterministic architecture we actually use is given separately in Appendix E; the present appendix instead explains, at a higher level, why the richer stochastic-consensus generalization is effective, why our deterministic method is a tractable simplification of it, and what additional mechanisms it would require to remain stable in practice.

This appendix provides a theoretical interpretation of CMAT's hierarchical design under simplified (tabular) conditions, building on established convergence results in tabular and linear settings, and the analysis clarifies how the latent consensus mechanism eliminates the order-dependent bias inherent in sequential MARL formulations such as MAT. It states the structural principles that motivate CMAT and develops the more general formulation that they point to. The most general analysis below is stated for a *stochastic* consensus policy $\pi^c(c \mid \mathcal{O})$ and a cooperative Stackelberg (bilevel) view, of which our implemented deterministic consensus $c = \mu_\theta(\mathcal{O})$ is a special case (Remark 6); the deep-network instantiation of these results is given in Appendix E. We develop the stochastic-consensus and full bilevel Stackelberg optimization as a promising extension, and note that realizing it in practice raises additional training-stability challenges (e.g., variance from sampling in the latent space and the need for variance-reduction techniques) that we leave for future work.

F.1 Preliminary Assumptions

For rigorous statements we adopt standard assumptions: finite observation space $\mathcal{O}$ and action spaces $\mathcal{A}^i$; a finite consensus set $\mathcal{C}$; softmax policy parameterization with full support; Robbins-Monro step sizes; and ergodicity of the induced Markov chain (Agarwal et al., 2021; Mei et al., 2020; Chen et al., 2024).

F.2 Reformulation as a Cooperative Stackelberg Game

CMAT models the joint policy as

$$\pi(\mathcal{A} \mid \mathcal{O}) = \int \pi^c(c \mid \mathcal{O}) \prod_{i=1}^{n} \pi^i\big(a^i \mid \mathcal{O}, c\big)\, dc, \tag{21}$$

where $\pi^c(c \mid \mathcal{O})$ is the consensus policy (stochastic in general), and each π^i is the action policy for agent i conditioned on $\mathcal{O}$ and c. The objective is the discounted cumulative reward $J(\pi^c, \pi^{1:n})$.

Remark 6 (Stochastic vs. Deterministic Consensus). The integral formulation in Eq. (21) provides the most general theoretical framework. In our implementation, we employ a deterministic consensus generator $c = \mu_\theta(\mathcal{O})$, which corresponds to $\pi^c(c \mid \mathcal{O}) = \delta(c - \mu_\theta(\mathcal{O}))$. Under this specialization, the joint policy reduces to

$$\pi(\mathcal{A} \mid \mathcal{O}) = \prod_{i=1}^{n} \pi^i(a^i \mid \mathcal{O}, \mu_\theta(\mathcal{O})), \tag{22}$$

which matches Eq. 1 in the main paper. While this restricts the mixture to a single component for any given $\mathcal{O}$, the expressiveness is recovered by allowing μ_θ to vary across different observations and by the high dimensionality of the consensus vector $c \in \mathbb{R}^d$. In practice, this deterministic choice simplifies the ratio computation (Eq.10 in the main paper) and avoids the need for importance weighting over the latent space.

This hierarchical structure can be viewed as a cooperative Stackelberg game: the leader (consensus policy π^c) commits to a latent strategy c, anticipating that the followers (action policies $\pi^{1:n}$) will best respond. Because all agents share the same reward, the game is fully cooperative. Stackelberg equilibria are known to be more Pareto-efficient than Nash equilibria in such settings:

Proposition 2 (Pareto Improvement over Nash Equilibrium)**.** *In a cooperative Markov game with identical rewards, any Stackelberg equilibrium* $(\pi^{c*}, \pi^{1:n*})$ *achieves a payoff at least as high as any Nash equilibrium payoff* J_{NE} *of the simultaneous-move game:*

$$J(\pi^{c*}, \pi^{1:n*}) \geq J_{NE}. \tag{23}$$

The inequality is strict when the Nash equilibrium is Pareto-suboptimal and the leader's commitment power is sufficient to steer the followers away from it.

Proof. In a cooperative (identical-interest) game, all agents share the objective J. A Stackelberg equilibrium $(\pi^{c*}, \pi^{1:n*})$ satisfies $\pi^{c*} \in \arg\max_{\pi^c} J(\pi^c, \mathrm{BR}(\pi^c))$, where $\mathrm{BR}(\pi^c)$ denotes the followers' best response.

Any Nash equilibrium $(\bar{\pi}^{1:n})$ of the simultaneous-move game can be recovered in the Stackelberg framework as follows: let π^c assign all probability to an arbitrary fixed consensus $\bar{c}$. Since $\bar{c}$ is uninformative, the followers' decision problem reduces to the original simultaneous-move game. By the definition of Nash equilibrium, $(\bar{\pi}^{1:n})$ is a mutual best response, so the payoff under this trivial consensus is exactly J_{NE}.

Since the Stackelberg leader optimizes over all possible π^c, including this trivial choice, the Stackelberg payoff must be at least the Nash payoff: $J(\pi^{c*}, \pi^{1:n*}) \geq J_{\mathrm{NE}}$. Strict inequality occurs when the Nash equilibrium is Pareto-suboptimal and the leader can commit to an informative consensus that induces a better response (Nanduri & Das, 2009; Fiez et al., 2020). □

F.3 The Policy-Class Ladder and Coordination Rank

The reason a *stochastic* consensus is strictly more powerful than the deterministic one we deploy can be stated exactly. Fix a state $\mathcal{O}$ and view a joint action distribution as a nonnegative tensor $P \in \mathbb{R}_{\geq 0}^{|\mathcal{A}^1| \times \cdots \times |\mathcal{A}^n|}$ with $\sum P = 1$. Define three policy classes,

$$\begin{aligned} \Pi_{\text{ind}} &= \Big\{ \textstyle\prod_i \pi^i(a^i \mid \mathcal{O}) \Big\}, \\ \Pi_{\text{con}}(d) &= \Big\{ \textstyle\sum_{c=1}^{d} \pi^c(c \mid \mathcal{O}) \prod_i \pi^i(a^i \mid \mathcal{O}, c) \Big\}, \\ \Pi_{\text{auto}} &= \Big\{ \textstyle\prod_i \pi(a^i \mid a^{<i}, \mathcal{O}) \Big\}, \end{aligned} \tag{24}$$

where $\Pi_{\text{con}}(d)$ uses a consensus taking d discrete values. By the chain rule Π_{auto} is the full simplex over joint actions (the class MAT represents).

Proposition 3 (Coordination-rank ladder)**.** *For a fixed $\mathcal{O}$, a joint distribution P belongs to $\Pi_{con}(d)$ if and only if its nonnegative tensor rank satisfies $\operatorname{rank}_+(P) \le d$. Consequently*

$$\Pi_{ind} = \Pi_{con}(1) \subsetneq \Pi_{con}(2) \subsetneq \cdots \subsetneq \Pi_{con}(d) \subsetneq \cdots \subseteq \Pi_{auto}, \tag{25}$$

with equality $\Pi_{con}(d) = \Pi_{auto}$ once $d \ge \prod_i |\mathcal{A}^i|$.

Proof. View a joint distribution over $\mathcal{A} = (a^1, \dots, a^n)$ as an order-n nonnegative tensor $P \in \mathbb{R}_{\ge 0}^{|\mathcal{A}^1| \times \cdots \times |\mathcal{A}^n|}$ with entries summing to one. An element of $\Pi_{\text{con}}(d)$ has the form $P = \sum_{c=1}^{d} \pi^c(c \mid \mathcal{O}) \prod_i \pi^i(\cdot \mid \mathcal{O}, c)$. Each summand is a nonnegative scalar times an outer product $\bigotimes_i \pi^i(\cdot \mid \mathcal{O}, c)$ of probability vectors, i.e. a nonnegative rank-one tensor, and the coefficients $\pi^c(\cdot \mid \mathcal{O})$ are nonnegative and sum to one, so P is a convex combination of d nonnegative rank-one tensors. By the definition of nonnegative tensor rank rank_+ as the least number of nonnegative rank-one terms summing to P, membership $P \in \Pi_{\text{con}}(d)$ is equivalent to $\operatorname{rank}_+(P) \le d$ (any such decomposition can be renormalized so that the coefficients form a distribution and each factor is a probability vector, since all terms are nonnegative). This proves the characterization.

The chain follows. Π_{ind} is the set of single product distributions, which is exactly $\operatorname{rank}_+(P) = 1$, so $\Pi_{\text{ind}} = \Pi_{\text{con}}(1)$. The inclusions $\Pi_{\text{con}}(d) \subseteq \Pi_{\text{con}}(d+1)$ are immediate (append a zero-weight term). They are strict below saturation because nonnegative rank is a strict hierarchy: for each d smaller than the maximal attainable nonnegative rank there exist nonnegative tensors of rank exactly $d+1$, which lie in $\Pi_{\text{con}}(d+1) \setminus \Pi_{\text{con}}(d)$. Finally, any P is a mixture of at most $\prod_i |\mathcal{A}^i|$ degenerate product distributions, one placing all mass on each joint action, so $\operatorname{rank}_+(P) \le \prod_i |\mathcal{A}^i|$ and $\Pi_{\text{con}}(d) = \Pi_{\text{auto}}$ once $d \ge \prod_i |\mathcal{A}^i|$, since Π_{auto} is the full simplex over joint actions. □

This motivates a problem-intrinsic scalar, the *coordination rank* $\operatorname{crank}_\varepsilon(\pi^\star) = \min\{d : \exists \pi \in \Pi_{\text{con}}(d),\ \|\pi - \pi^\star\| \le \varepsilon\}$, which measures how much coordination a task genuinely requires and can be estimated empirically from the effective rank of the learned consensus. Marginalizing the shared latent c correlates the a^i, so c acts as a learned correlation device, the analogue of the shared signal in a correlated equilibrium (Aumann, 1974), which is entirely absent in Π_{ind}. We stress the boundary with Appendix E: the deterministic model we deploy is, for a fixed $\mathcal{O}$, the rank-one end $\Pi_{\text{con}}(1)$ of this ladder (Remark 7), and its effectiveness rests on the value-function decomposition proved there, not on this ladder. The ladder characterizes the strictly larger class unlocked only by a stochastic consensus, which is why we treat it as future work.

F.4 Theoretical Justification 1: The Leader's Problem as a Finite MDP

When the action policies $\pi^{1:n}$ are fixed, the leader faces a finite MDP with state space $\mathcal{O}$, action space $\mathcal{C}$, transition

$$P(\mathcal{O}' \mid \mathcal{O}, c) = \mathbb{E}_{\pi^{1:n}}\big[P_{\text{env}}(\mathcal{O}' \mid \mathcal{O}, \mathcal{A}) \mid c\big], \tag{26}$$

and reward $r(\mathcal{O}, c) = \mathbb{E}_{\pi^{1:n}}[R(\mathcal{O}, \mathcal{A}) \mid c]$. For this finite MDP, standard RL algorithms such as policy iteration or Q-learning are guaranteed to converge to the optimal consensus policy π^{c*} (Sutton et al., 1998). Moreover, since the leader's problem is a finite MDP, restricting π^c to deterministic policies incurs no loss of optimality in the tabular setting (Puterman, 2014), which justifies the deterministic implementation in Remark 6.

F.5 Theoretical Justification 2: Consensus as a Coordination Signal

With the consensus policy π^c fixed, the followers' joint policy factorizes as in Eq. (21). Conditioned on a specific c, the action policies are conditionally independent.

Crucially, the policy gradient for agent i takes the form:

$$\nabla_{\theta_i} J = \mathbb{E}\Big[\nabla_{\theta_i} \log \pi^i_{\theta_i}(a^i \mid \mathcal{O}, c) \cdot A(\mathcal{O}, \mathcal{A})\Big], \tag{27}$$

where the expectation is over trajectories generated by the current joint policy. Although the environment dynamics still couple the agents through the joint action $\mathcal{A}$, the parameter updates for

different agents are conditionally independent given c, in the sense that the gradient for θ_i does not directly involve the policy parameters θ_j ($j \neq i$) except through their effect on the trajectory distribution (captured by the advantage estimate).

Under tabular softmax parameterization, multi-agent policy gradient in identical-interest games converges to a Nash equilibrium (Chen et al., 2024; Zhong et al., 2024). Furthermore, in identical-interest games, any Nash equilibrium maximizes the common payoff when the best response is unique (Wang & Sandholm, 2002). When multiple best responses exist, the consensus c serves as a coordination device that selects among them: by conditioning on c, agents can break symmetry and coordinate on the Pareto-optimal equilibrium. This coordination role is precisely what is missing in simultaneous-move formulations, where agents must independently select among multiple equilibria. Therefore, for a fixed consensus c, the followers' optimization converges to the global optimum of J conditional on c.

F.6 Theoretical Justification 3: Alternating Optimization as Block Coordinate Ascent

The fine-tuning phase of CMAT (Consensus Enhancement and Action Policy Enhancement) performs alternating updates. This can be interpreted as block coordinate ascent on J:

$$\pi^c_{k+1} \leftarrow \arg\max_{\pi^c} J(\pi^c, \pi^{1:n}_k), \tag{28}$$

$$\pi^{1:n}_{k+1} \leftarrow \arg\max_{\pi^{1:n}} J(\pi^c_{k+1}, \pi^{1:n}). \tag{29}$$

Each block update guarantees monotonic improvement under tabular assumptions when solved exactly. In practice, each block is optimized approximately via a finite number of PPO steps. Under the Polyak-Łojasiewicz (PL) condition, which holds for softmax policies in tabular settings (Mei et al., 2020), approximate alternating optimization still converges at a rate of $O(1/K)$ to a stationary point of the bilevel objective (Xiao et al., 2023). The sequence $J(\pi^c_k, \pi^{1:n}_k)$ converges to a Stackelberg equilibrium.

F.7 Policy Gradient and Credit Assignment under Deterministic Consensus

In our practical implementation, the consensus policy π^c is replaced by a deterministic mapping $c = \mu_\theta(\mathcal{O})$ as in Eq. (22). The joint policy gradient can then be derived via the reparameterization trick:

$$\nabla_\theta \log \pi_\theta(\mathcal{A} \mid \mathcal{O}) = \sum_{i=1}^{n} \nabla_\theta \log \pi^i_\theta(a^i \mid \mathcal{O}, \mu_\theta(\mathcal{O})). \tag{30}$$

This gradient flows through both the action policies π^i_θ and the consensus generator μ_θ, allowing end-to-end optimization via standard policy gradient methods such as PPO.

To be more precise, the gradient decomposes as:

$$\begin{aligned} &\nabla_\theta \log \pi_\theta(\mathcal{A} \mid \mathcal{O}) = \\ &\sum_{i=1}^{n} \Big(\underbrace{\nabla_\theta \log \pi^i_\theta(a^i \mid \mathcal{O}, c)\big|_{c=\mu_\theta(\mathcal{O}),\, \nabla_\theta c=0}}_{\text{direct effect}} \\ &+ \underbrace{\frac{\partial \log \pi^i_\theta}{\partial c} \cdot \frac{\partial \mu_\theta}{\partial \theta}}_{\text{consensus effect}} \Big). \end{aligned} \tag{31}$$

Here, the "direct effect" treats the consensus c as a fixed input (i.e., stopping the gradient through μ_θ), while the "consensus effect" captures the gradient flowing through the consensus generator. The direct effect for agent i depends only on agent i's observation and action (o^i, a^i, c). The consensus effect aggregates information from all agents but flows through the shared consensus vector, avoiding the cascading sequential dependencies of MAT. This structural decoupling means that the gradient signal for each agent's action contributes additively to the overall update, in contrast to MAT where the policy of agent i explicitly conditions on the actions of preceding agents $a^{1:i-1}$, creating entangled gradients that complicate credit assignment.

F.8 Comparison with MAT: Policy Space Expressiveness

MAT's autoregressive factorization $\pi(\mathcal{A} \mid \mathcal{O}) = \prod_{i=1}^{n} \pi^{\sigma(i)}(a^{\sigma(i)} \mid \mathcal{O}, a^{\sigma(1:i-1)})$ can represent any joint policy (by the chain rule of probability). CMAT's factorization in Eq. (21) is a mixture of product distributions. The following proposition clarifies their relative expressiveness:

Proposition 4 (Expressiveness of CMAT). *Let Π_{MAT} and Π_{CMAT} be the sets of joint policies representable by MAT and CMAT, respectively.*

(i) If $|\mathcal{C}| \geq \prod_{i=1}^{n} |\mathcal{A}^i|$, then $\Pi_{CMAT} \supseteq \Pi_{MAT}$.

(ii) For any finite $|\mathcal{C}| < \prod_{i=1}^{n} |\mathcal{A}^i|$, there exist policies in Π_{MAT} that cannot be represented in Π_{CMAT}.

(iii) With a continuous consensus space $\mathcal{C} = \mathbb{R}^d$ and sufficient network capacity, CMAT can approximate any joint policy arbitrarily well.

Proof. (i) Enumerate the joint actions as $\mathcal{A} = \{A_1, \ldots, A_M\}$ with $M = \prod_i |\mathcal{A}^i|$. For each $A_m = (a_m^1, \ldots, a_m^n)$ let the consensus state c_m induce the deterministic factors $\pi^i(\cdot \mid \mathcal{O}, c_m) = \delta_{a_m^i}$, so that $\prod_i \pi^i(\cdot \mid \mathcal{O}, c_m) = \delta_{A_m}$ is the point mass on A_m. Any target joint distribution P can then be written as $P = \sum_{m=1}^{M} P(A_m)\, \delta_{A_m}$, which is realized by setting $\pi^c(c_m \mid \mathcal{O}) = P(A_m)$. Hence with $|\mathcal{C}| \geq M$ consensus states every joint distribution, including every element of Π_{MAT}, is representable, giving $\Pi_{\text{CMAT}} \supseteq \Pi_{\text{MAT}}$.

(ii) Fix $|\mathcal{C}| = K < M$. The representable set is the image of the map sending $(\pi^c, \{\pi^i(\cdot \mid \mathcal{O}, c)\})$ to $\sum_{c=1}^{K} \pi^c(c) \prod_i \pi^i(\cdot \mid \mathcal{O}, c)$. Its parameters number $(K-1)$ for π^c and $\sum_i (|\mathcal{A}^i| - 1)$ per consensus state for the K product factors, so the image is the image of a smooth map from a domain of dimension at most $(K-1) + K \sum_i (|\mathcal{A}^i| - 1) = K(\sum_i |\mathcal{A}^i| - n) + (K-1)$. By Sard's theorem the image of a smooth map has measure zero in any target manifold of strictly larger dimension. The full simplex over joint actions has dimension $M-1$, and for $K < M$ the parameter-count above is strictly smaller than $M-1$ for action spaces of interest, so the representable set has measure zero in the simplex. Therefore some joint distributions, in particular full-rank ones requiring $\mathrm{rank}_+ = M$, cannot be represented; this is the $d = K$ case of Proposition 3.

(iii) Let $\mathcal{C} = \mathbb{R}^d$. Any target joint distribution is a finite mixture of the M point masses as in (i). Instantiate M well-separated latent codes $c_1, \ldots, c_M \in \mathbb{R}^d$; a network of sufficient capacity can approximate the piecewise-constant assignment $c_m \mapsto \delta_{a_m^i}$ for each head π^i and the weighting $c_m \mapsto P(A_m)$ for π^c to arbitrary accuracy by universal approximation (Goodfellow, 2016), so the induced joint distribution approximates P arbitrarily well. (A continuous π^c approximates the discrete mixing weights to any tolerance.) □

In practice, CMAT employs a high-dimensional continuous consensus vector ($d = 128$ or 256), ensuring sufficient expressive capacity to match or exceed MAT's representational power while avoiding the order-dependence bias.

Remark 7 (Scope of Proposition 4 under a deterministic consensus). Parts (i) and (iii) of Proposition 4 concern the general mixture policy of Eq. (21), in which the consensus is drawn from a distribution $\pi^c(c \mid \mathcal{O})$. Our implemented model instead uses a deterministic consensus $c = \mu_\theta(\mathcal{O})$ (Remark 6), so for a *fixed* observation $\mathcal{O}$ the joint policy is a single product distribution $\prod_i \pi^i(a^i \mid \mathcal{O}, c)$ rather than a mixture. Consequently, conditioned on a fixed $\mathcal{O}$, the deterministic model cannot represent within-state correlations among agents' actions that MAT's autoregressive factorization can capture; the expressive gains of parts (i) and (iii) are realized only by the stochastic-consensus variant. In the deterministic case, coordination is instead carried by allowing $c = \mu_\theta(\mathcal{O})$ to vary across observations and by the shared representation feeding all action heads. We view closing this gap through a stochastic consensus policy as the main theoretical extension left for future work.

F.9 Relation to Existing Convergence Results

The above insights are supported by recent theoretical advances:

Table 1: Structural comparison between MAT and CMAT. Entries marked "(tabular)" refer to the idealized analysis of this appendix, not to guarantees for the deep, deterministic-consensus implementation.

Property	MAT	CMAT
Action generation	Sequential (order-dependent)	Simultaneous (order-independent)
Convergence target (tabular)	Nash Equilibrium	Stackelberg Equilibrium $\geq$ NE (stochastic-consensus variant)
Credit assignment	Entangled (cascading)	Decoupled (given c) + consensus gradient
Policy expressiveness	Universal (chain rule)	Universal with stochastic/continuous c; single product given fixed $\mathcal{O}$ if deterministic
Convergence complexity (tabular)	$O(n \cdot \lvert\mathcal{A}^{\max}\rvert)$ (per step)	$O(\lvert\mathcal{C}\rvert + n\lvert\mathcal{A}^{\max}\rvert)$ (additive scaling)

- Tabular softmax policy gradient converges globally at a sublinear rate (Agarwal et al., 2021; Mei et al., 2020).
- Multi-agent policy gradient converges to Nash equilibria in cooperative Markov games under tabular assumptions (Chen et al., 2024; Zhong et al., 2024).
- PPO variants have been shown to converge in both tabular and linear approximation settings (Holzleitner et al., 2021; Huang et al., 2024).
- Stackelberg equilibria in Markov games can be learned via bilevel reinforcement learning (Nanduri & Das, 2009; Fiez et al., 2020).

Remark 8 (Convergence Rate Intuition). Under tabular softmax parameterization, the leader's MDP has $|\mathcal{O}|$ states and $|\mathcal{C}|$ actions, yielding a convergence rate of $\tilde{O}(|\mathcal{O}||\mathcal{C}|/\epsilon^2)$ for ϵ-optimality (Agarwal et al., 2021). Each follower's problem, with fixed c, has effective action space $|\mathcal{A}^i|$, converging at $\tilde{O}(|\mathcal{O}||\mathcal{A}^i|/\epsilon^2)$. The total complexity of one alternating cycle is thus

$$\tilde{O}\Big(|\mathcal{O}|\big(|\mathcal{C}| + n|\mathcal{A}^{\max}|\big)/\epsilon^2\Big), \tag{32}$$

which scales additively in the number of agents n, in contrast to the $\prod_i |\mathcal{A}^i|$ joint action space of a naive centralized approach.

F.10 Scope and Open Directions

The formal results in this appendix are stated for the tabular and linear-quadratic instances in which the coordination quantities are well defined, and they carry over to the deep network by the fixed-budget reading of Section F.3: they characterize what the architecture can represent at a given parameter budget, and the experiments determine how much of it training reaches. Extending the convergence statements to neural function approximation is a target for further work rather than a claim made here. Two directions in the same spirit are within reach. The first is stochastic consensus: a deterministic $\mu_\theta(\mathcal{O})$ maps each observation to one coordination signal, and letting the consensus be sampled (for instance through a variational formulation) would widen the space of team strategies the latent can carry and enrich exploration over it, at the cost of a variance-reduction scheme to keep training stable. The second is the bilevel scheme implied by the Stackelberg formulation of Appendix F, in which the consensus policy is the leader and the action policies are the followers. The alternating updates we use can be read as one practical instantiation of it, and a full treatment requires controlling the relative training speeds of the two levels, which is where the difficulty lies. The structural advantages established here (order independence, a policy space that can match autoregressive models, decoupled credit assignment, and the alternating scheme itself) are what the empirical results in Section 4 reflect, in the same way that tabular analyses have guided deep algorithms elsewhere (Schulman et al., 2017; Yu et al., 2022).

G Discussions

In this section, we discuss the scope of this work and outline promising directions for future research. First, although this paper addresses the fully cooperative setting with global observation, a scenario relevant to real-world smart city applications such as ride-hailing, traffic signal control, and power system management, our experiments are conducted on standard MARL game testbeds, which isolate the coordination structure we study from the engineering constraints of a deployed system. Extending the evaluation to practical, large-scale tasks is the natural next step. Second, as noted in Hu et al. (2024b), while the centralized paradigm leveraged in our approach achieves strong performance by

exploiting global information, it also brings communication overhead and exposure to single-point failures. Communication-efficient mechanisms are a valuable direction for broadening the robustness and scalability of the method. Finally, given the remarkable generalization capabilities demonstrated by Transformers in large language models, future research could explore the potential of our approach in few-shot learning and other transfer learning scenarios, thereby extending its applicability to tasks with limited data.