SDM: SYNERGISTIC DEPENDENCY MAXIMIZATION FOR COOPERATIVE MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Cooperative multi-agent reinforcement learning requires agents to develop complementary behaviors whose task value emerges through interaction with their teammates. A fundamental challenge is to quantify task-relevant synergy arising from these interactions and translate it into an effective signal for policy learning. To address this challenge, we introduce Synergistic Dependency Maximization (SDM), which promotes task-relevant interaction coupling by encouraging an agent's action to become more informative about the multi-step team return when its teammates' actions are taken into account. Using Partial Information Decomposition (PID), we formalize synergistic dependency through synergy beyond redundancy, yielding a principled task-relevant coordination criterion for multi-agent interactions. To estimate this dependency reliably, SDM adopts a conservative estimation strategy and uses the resulting signal to guide policy optimization while retaining the original task objective for value estimation. Experiments across MPE, MAMuJoCo, SMAC, and SMACv2 demonstrate consistent improvements over strong multi-agent policy optimization baselines and existing explicit coordination objectives. These results establish synergistic dependency as a principled objective for promoting cooperative behavior.

1 INTRODUCTION

Cooperative multi-agent reinforcement learning (MARL) provides a framework for multiple agents to learn coordinated policies that maximize a shared team return through interaction with an environment (Yuan et al., 2023; Huh & Mohapatra, 2023). It has been applied to a wide range of domains, including autonomous driving (Schmidt et al., 2022), UAV swarms (Batra et al., 2022), intelligent transportation (Chu et al., 2019), and multi-robot control (Nachum et al., 2019). In many practical settings, agents must execute decentralized policies using only locally available information under partial observability and communication constraints (Foerster et al., 2016; Peng et al., 2021). During training, continual updates to teammates' policies make the effective environment non-stationary from each agent's perspective (Foerster et al., 2017; De Witt et al., 2020). Together, these challenges make it difficult for independently learning agents to acquire robust and coordinated behaviors.

To alleviate these difficulties, centralized training with decentralized execution (CTDE) has become a widely adopted paradigm in cooperative MARL, allowing agents to exploit global information during training while retaining decentralized policies for execution (Sunehag et al., 2017; Foerster et al., 2018; Yu et al., 2022). Under the CTDE paradigm, substantial progress has been achieved by both off-policy methods, such as MADDPG (Lowe et al., 2017) and QMIX (Rashid et al., 2020), and on-policy methods, such as MAPPO (Yu et al., 2022). More recently, HATRPO and HAPPO introduce sequential policy updates to enable coordinated optimization of heterogeneous policies (Kuba et al., 2021; Zhong et al., 2024). Despite these advances, most CTDE methods do not explicitly optimize inter-agent coordination, instead relying on centralized training and shared-return optimization for coordinated behaviors to emerge.

Recent studies have pursued more explicit coordination through auxiliary objectives that directly capture inter-agent interactions, such as promoting statistical dependence among agents' actions via social influence (Jaques et al., 2019; Kim et al., 2020; Hou et al., 2025b) or quantifying how individ-

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.

ual actions influence subsequent state transitions (Wang et al., 2019; Liu et al., 2023; Li et al., 2024). Task-aware extensions further distinguish superior and inferior trajectories in PMIC (Li et al., 2022), or combine multi-step causal influence on teammates' future states with advantage-based gating in MAGIC (Yu et al., 2026). However, existing influence-based objectives mainly characterize either action–action or action–state influence, without explicitly capturing the task relevance and interaction coupling that underlie synergistic cooperation. In particular, they do not directly determine whether an agent's contribution to the team objective emerges or is strengthened through interaction with its teammates. Consequently, these influence measures provide limited guidance for explicitly promoting task-relevant synergistic behaviors.

To address this gap, we propose *Synergistic Dependency Maximization* (SDM), a principled framework for explicitly promoting synergistic coordination. Rather than rewarding inter-agent influence itself, SDM asks whether an agent's action becomes more informative about the multi-step team return when its teammates' actions are known than when it is considered individually. We term this interaction-induced increase in task-relevant dependency *synergistic dependency*, which captures the extent to which an agent's contribution is strengthened through interaction with its teammates. A key challenge, however, is that estimation errors can induce spurious synergy. To address this, SDM adopts a conservative estimation strategy (Oord et al., 2018; Cheng et al., 2020). As a result, strong synergy is identified only when there is sufficient evidence of additional task-relevant information arising from interaction. SDM then incorporates this signal into the policy advantage to guide policy optimization toward cooperative behaviors, while leaving the original task objective for value estimation unchanged. Our contributions are summarized as follows:

- We formulate synergistic dependency to capture task-relevant coordination strengthened by teammate context.
- We propose SDM with a conservative estimation principle to obtain a robust coordination signal for promoting synergistic behaviors.
- We show that SDM can be readily integrated with representative on-policy CTDE methods, including MAPPO and HAPPO, as a plug-in coordination mechanism.
- Extensive experiments demonstrate consistent performance gains, with ablation and mechanism analyses further supporting the effectiveness of SDM.

2 Background and Related Work

2.1 Cooperative MARL under CTDE

Problem formulation. We consider fully cooperative multi-agent reinforcement learning problems modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) Oliehoek et al. (2016), defined by $M = \langle \mathcal{N}, \mathcal{S}, \boldsymbol{\mathcal{A}}, \mathcal{P}, \mathcal{R}, \boldsymbol{\mathcal{O}}, \gamma \rangle$. Here, $\mathcal{N} = \{1, \dots, n\}$ is the set of agents, $\mathcal{S}$ is the global state space, and $\boldsymbol{\mathcal{A}} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^n$ is the joint action space. $\gamma \in [0, 1)$ denotes the discount factor. At time step t, each agent i receives a local observation $o_t^i \in \mathcal{O}^i$, and selects an action $a_t^i \in \mathcal{A}^i$. The joint action $\boldsymbol{a}_t = (a_t^1, \dots, a_t^n)$ induces a transition $\mathcal{P}(s_{t+1} \mid s_t, \boldsymbol{a}_t)$, and all agents receive a shared reward $r_t = \mathcal{R}(s_t, \boldsymbol{a}_t)$. The objective is to learn a joint policy $\boldsymbol{\pi} = \{\pi^i\}_{i=1}^n$ that maximizes the expected discounted team return $J(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]$.

CTDE. Under centralized training with decentralized execution (CTDE) (Lowe et al., 2017), agents leverage global information during training while acting only on local observations at execution. Off-policy value-decomposition methods, such as VDN (Sunehag et al., 2017), QMIX (Rashid et al., 2020), QTRAN (Son et al., 2019; 2020), and QPLEX (Wang et al., 2020a), learn coordinated behaviors through centralized value learning, but can suffer from stale-data issues under multi-agent non-stationarity due to temporal-difference learning and replay (Foerster et al., 2017). In contrast, on-policy methods optimize policies using data from the current joint policy. MAPPO (Yu et al., 2022) combines PPO with centralized value estimation, while HATRPO and HAPPO (Kuba et al., 2021; Zhong et al., 2024) support sequential optimization of heterogeneous policies. Despite their effectiveness, these methods mainly rely on shared-return optimization and do not explicitly promote inter-agent synergy. We therefore instantiate SDM on MAPPO and HAPPO to provide an explicit synergy signal during policy learning.

Table 1: Comparison of explicit coordination signals, where A_t^i and A_t^{-i} denote the current individual and teammate actions, S_t and S_{t+1} the current and subsequent states, and Y_t the multi-step team return. "Partial" indicates that the signal reflects inter-agent interaction but does not explicitly determine whether an agent's task-relevant contribution is strengthened by teammate context.

Signal	Representative measure	Task relevance	Interaction coupling
Action–Action	$I(A_{t+1}^{-i}; A_t^i \mid S_t, A_t^{-i})$	No	Partial
Action–State	$I(S_{t+1}; A_t^i \mid S_t, A_t^{-i})$	No	Partial
Synergistic Dependency ($\beta = 1$)	$I(Y_t; A_t^i \mid S_t, A_t^{-i}) - I(Y_t; A_t^i \mid S_t)$	Yes	Yes

2.2 Influence-Based Coordination

Existing influence-based coordination methods mainly characterize either action–action or action–state dependencies, as summarized in Table 1. Action–action approaches promote coordination by encouraging statistical dependence among agents' actions through social influence or mutual-information objectives (Jaques et al., 2019; Kim et al., 2020; Hou et al., 2025b). Action–state approaches instead measure how an agent's action influences subsequent state transitions (Wang et al., 2019; Liu et al., 2023; Li et al., 2024). Both capture inter-agent interaction to some extent, but neither explicitly links the measured dependency to the shared team objective or separates interaction-induced contributions from individual effects. We therefore characterize genuine synergy through two properties, *task relevance* and *interaction coupling*, which motivate the synergistic dependency studied in this work. A more comprehensive discussion of related work is provided in Appendix A.

2.3 Information-Theoretic Dependency Estimation

Mutual information (MI) provides a principled measure of statistical dependence and has been widely used in representation learning and decision making (Cover et al., 1991; Belghazi et al., 2018; Poole et al., 2019). For random variables X and Y, $I(X;Y) = D_{\mathrm{KL}}(p(x,y)\|p(x)p(y))$, while conditional mutual information (CMI), $I(X;Y \mid Z) = \mathbb{E}_{p(z)}\left[D_{\mathrm{KL}}\big(p(x,y \mid z)\|p(x \mid z)p(y \mid z)\big)\right]$, measures statistical dependence under a given context. Both have been used in MARL to characterize dependencies among agents and task-related variables (Li et al., 2021; Jo et al., 2024).

Estimating MI from finite samples is challenging, particularly in high-dimensional spaces (Paninski, 2003), motivating variational and contrastive estimators (Nguyen et al., 2010; Belghazi et al., 2018; Poole et al., 2019; Guo et al., 2022). InfoNCE (Oord et al., 2018) provides a tractable contrastive lower bound, whereas CLUB (Cheng et al., 2020) provides a variational upper bound based on contrastive conditional likelihoods. Such lower- and upper-bound estimators make information-theoretic objectives tractable for maximizing, preserving, or constraining statistical dependencies in high-dimensional learning problems. In the next section, we leverage these complementary estimators to construct a conservative estimate of synergistic dependency.

3 Synergistic Dependency Maximization

In this section, we present Synergistic Dependency Maximization (SDM), which explicitly quantifies and promotes task-relevant dependencies arising from inter-agent interactions. As illustrated in Figure 1, SDM extracts a synergistic dependency signal from multi-step team returns and uses it to guide actor optimization without altering environment-based value estimation. We first define synergistic dependency, then develop a conservative estimator for reliable estimation, and finally incorporate the resulting signal into on-policy optimization. See Appendix C.

3.1 Synergistic Dependency

Effective cooperation requires agents to complement one another toward a shared task objective. Yet, what distinguishes genuine synergy from an agent's individual influence or behavioral correlation? We argue that truly synergistic behavior should exhibit two essential properties: *task relevance*, meaning that an agent's contribution is directly related to the team task, and *interaction coupling*, meaning that this contribution emerges or becomes stronger through interaction with its teammates.

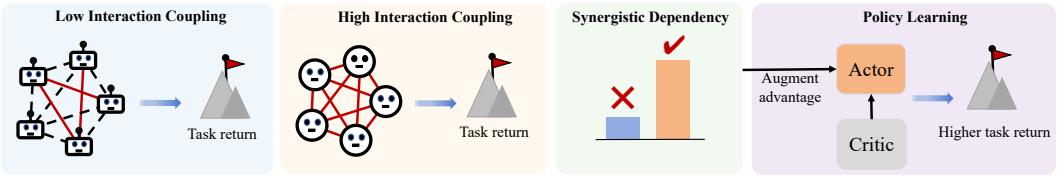

Figure 1: Overview of Synergistic Dependency Maximization (SDM). SDM captures the additional task-relevant dependency revealed by teammates' actions and uses it to guide actor optimization, while value estimation remains grounded solely in environment returns.

To capture task relevance, we consider the H-step team return $Y_t = \sum_{k=0}^{H-1} \gamma^k r_{t+k}$, which summarizes the consequence of the current joint decision over a future horizon. We then characterize interaction coupling through partial information decomposition (PID) (Williams & Beer, 2010; Bertschinger et al., 2014), which decomposes the task-relevant information about Y_t across the action of agent i and the joint actions of its teammates.

For clarity, we first omit the conditioning on the current state. Given two information sources, A_t^i and A_t^{-i}, partial information decomposition (PID) decomposes their joint information about Y_t as

$$I(Y_t; A_t^i, A_t^{-i}) = R(Y_t; A_t^i, A_t^{-i}) + U_i(Y_t; A_t^i \mid A_t^{-i}) + U_{-i}(Y_t; A_t^{-i} \mid A_t^i) + \mathrm{Syn}(Y_t; A_t^i, A_t^{-i}), \quad (1)$$

For brevity, we write the redundant, unique, and synergistic components as $R_t^{i,-i}$, U_t^i, U_t^{-i}, and $\mathrm{Syn}_t^{i,-i}$, respectively. Here, $R_t^{i,-i}$ denotes the information redundantly available from both agent i and its teammates; U_t^i and U_t^{-i} denote the information uniquely available from agent i and its teammates, respectively; and $\mathrm{Syn}_t^{i,-i}$ denotes the information that is available only when their actions are considered jointly. Under this decomposition, the information about Y_t available from agent i alone and from its teammates alone can be expressed as $I(Y_t; A_t^i) = R_t^{i,-i} + U_t^i$ and $I(Y_t; A_t^{-i}) = R_t^{i,-i} + U_t^{-i}$, respectively. Substituting these relations into Eq. 1, the synergistic information can be expressed as

$$\mathrm{Syn}_t^{i,-i} = I(Y_t; A_t^i, A_t^{-i}) - I(Y_t; A_t^i) - I(Y_t; A_t^{-i}) + R_t^{i,-i}. \quad (2)$$

The synergistic component $\mathrm{Syn}_t^{i,-i}$ captures task-relevant information that emerges only when agent i and its teammates are considered jointly. However, the two sources may also contain substantial redundant information. Therefore, $\mathrm{Syn}_t^{i,-i} - R_t^{i,-i}$ captures the net complementary information arising from their joint interaction, indicating whether synergistic information outweighs redundancy. Such complementarity is particularly desirable in cooperative decision making, where differentiated yet coordinated behaviors can improve collective performance (Wang et al., 2020b;c; Goel et al., 2025). This motivates $\mathrm{Syn}_t^{i,-i} - R_t^{i,-i}$ as a principled measure of synergistic dependency that can guide agents toward learning more cooperative policies. Using the chain rule $I(Y_t; A_t^i, A_t^{-i}) = I(Y_t; A_t^{-i}) + I(Y_t; A_t^i \mid A_t^{-i})$, Eq. 2 can be rewritten as

$$\mathrm{Syn}_t^{i,-i} - R_t^{i,-i} = I(Y_t; A_t^i \mid A_t^{-i}) - I(Y_t; A_t^i). \quad (3)$$

Under the PID interpretation, this difference captures synergy beyond redundancy and is closely related to co-information, providing a principled task-relevant coordination criterion for cooperative MARL. Returning to the cooperative MARL setting, we condition all information quantities on the current state S_t to isolate task-relevant dependencies beyond those explained by the current environment context. Accordingly, we define the synergistic dependency (SD) of agent i as

$$\mathcal{D}_t^i = \underbrace{I(Y_t; A_t^i \mid S_t, A_t^{-i})}_{\text{interaction-conditioned dependency}} - \beta \underbrace{I(Y_t; A_t^i \mid S_t)}_{\text{individual dependency}}, \quad (4)$$

where $\beta \geq 0$ controls the strength of the individual-dependency correction. When $\beta = 1$, $\mathcal{D}_t^i$ recovers the PID-motivated synergy-minus-redundancy criterion, while other values provide a practical relaxation of this criterion. Both terms are defined with respect to the team return Y_t, ensuring *task relevance*. Importantly, the first term alone does not isolate synergy, since it may also reflect task-relevant information attributable to the agent individually. Subtracting the individual dependency therefore isolates the additional task-relevant dependency revealed by teammate context. Accordingly, $\mathcal{D}_t^i$ characterizes *interaction coupling*, indicating whether teammate context strengthens the informativeness of agent i's action about the team return beyond its individual dependency.

3.2 Conservative Dependency Estimation

Reliable estimation of synergistic dependency from finite samples is challenging. In particular, overestimating the interaction-conditioned dependency or underestimating the individual dependency can spuriously inflate the resulting synergy signal.

We therefore adopt an asymmetric conservative estimation principle: the interaction-conditioned dependency is estimated using a lower-bound objective, whereas the individual dependency is estimated using an upper-bound objective. Let

$$I_{\text{int}}^{i,\text{LB}} \le I(Y_t; A_t^i \mid S_t, A_t^{-i}), \qquad I_{\text{ind}}^{i,\text{UB}} \ge I(Y_t; A_t^i \mid S_t),$$

under the corresponding variational assumptions. For each transition, we denote the corresponding estimator outputs by $I_{\text{int},t}^{i,\text{LB}}$ and $I_{\text{ind},t}^{i,\text{UB}}$, respectively. We then construct $\widehat{D}_t^i = I_{\text{int},t}^{i,\text{LB}} - \beta I_{\text{ind},t}^{i,\text{UB}}$. This asymmetric construction is conservative by favoring synergy signals only when the interaction-conditioned dependency reliably exceeds the individual dependency.

Interaction-Conditioned Dependency Lower Bound. We estimate $I(Y_t; A_t^i \mid S_t, A_t^{-i})$ using conditional InfoNCE (Oord et al., 2018; Poole et al., 2019). Let $C_t^i = (S_t, A_t^{-i})$ and let $f_\phi(Y_t, A_t^i, C_t^i)$ denote a learnable compatibility score. For each transition, we keep C_t^i fixed and sample $N-1$ alternative actions from the behavior policy $\pi_{\text{old}}^i(\cdot \mid o_t^i)$ as context-matched negatives. Contrasting the observed action–return pair against these alternatives gives $I_{\text{int}}^{i,\text{LB}} = \log N - \mathcal{L}_{\text{NCE}}$, which provides a lower bound on the corresponding interaction-conditioned mutual information under the standard InfoNCE formulation. Further details are provided in Appendix B.

Individual Dependency Upper Bound. For the individual dependency, we adopt variational CLUB (Cheng et al., 2020). We learn a conditional density $q_\psi(Y_t \mid S_t, A_t^i)$ by maximizing the likelihood of observed returns. State-matched negatives are then constructed by keeping S_t fixed and resampling A_t^i from the behavior policy, yielding the variational CLUB estimate $I_{\text{ind}}^{i,\text{UB}}$, which approximates an upper bound when the learned conditional distribution is sufficiently accurate.

3.3 Synergy-Guided Policy Optimization

We next integrate synergistic dependency into existing on-policy CTDE algorithms through advantage-based policy optimization. Specifically, we retain the positive component $[\widehat{\mathcal{D}}_t^i]_+$ and normalize its scale for stable optimization, yielding the agent-wise synergy signal $\widetilde{\mathcal{D}}_t^i$. Since these signals characterize the synergistic contributions of different agents toward the same team objective, we aggregate them into a team-level signal, $\widetilde{\mathcal{D}}_t = \frac{1}{N}\sum_{i=1}^N \widetilde{\mathcal{D}}_t^i$. The mean aggregation provides a permutation-invariant summary of team synergy while keeping its scale insensitive to the number of agents. Our objective is to promote synergistic dependency along directions that improve the team task. Accordingly, we apply an advantage gate $g_t^{\text{adv}} = \mathbb{I}[\mathcal{A}_{t,\text{env}} > 0]$ to activate synergy only for decisions favored by the environment objective. Rather than using reward shaping, we inject the gated synergy signal directly into the actor advantage:

$$\widehat{\mathcal{A}}_{t,\text{SDM}} = \mathcal{A}_{t,\text{env}} + \lambda g_t^{\text{adv}} \widetilde{\mathcal{D}}_t, \tag{5}$$

while leaving task-value estimation unchanged. λ controls the contribution of synergistic dependency to policy optimization. The environment advantage therefore determines the direction of policy improvement, while SD strengthens updates that additionally exhibit team-level synergy. Meanwhile, the critic remains trained solely on environment returns.

We instantiate SDM on MAPPO and HAPPO by replacing their original advantage with $\widehat{\mathcal{A}}_{t,\text{SDM}}$ while preserving their respective policy optimization procedures. Omitting entropy regularization for clarity, both actor updates can be written in the unified form

$$\mathcal{J}_{\text{policy}}^i = \mathbb{E}_t\left[\omega_t^i \min\Big(\rho_t^i \widehat{\mathcal{A}}_{t,\text{SDM}}, \text{clip}\left(\rho_t^i, 1-\epsilon, 1+\epsilon\right)\widehat{\mathcal{A}}_{t,\text{SDM}}\Big)\right], \tag{6}$$

where ρ_t^i is the importance ratio of agent i. For MAPPO, $\omega_t^i = 1$, whereas for HAPPO, ω_t^i denotes the sequential correction factor accounting for policy changes of agents updated before agent i. Thus, SDM modifies only the policy-improvement signal while preserving the original optimization mechanisms of the underlying on-policy CTDE algorithms.

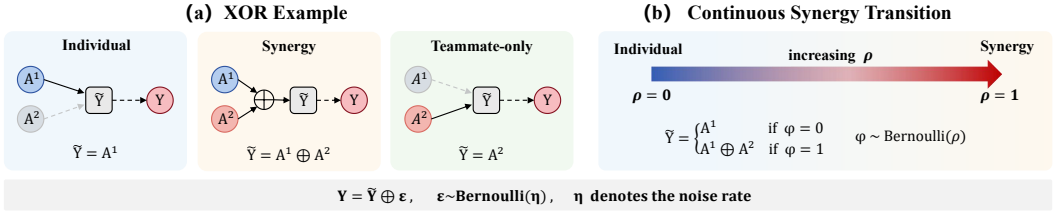

Figure 2: Motivating example of synergistic dependency. (a) Three binary interaction structures: Individual, Synergy, and Teammate-only. (b) A Bernoulli selector with parameter ρ continuously interpolates from individual to synergistic dependence.

Table 2: Dependency estimation in the binary motivating example ($\eta = 0.2$, $\beta = 1$). By symmetry, only agent 1 is shown. $\star$ denotes the analytical value, and $\mathrm{InfoNCE}^\star$ and $\mathrm{CLUB}^\star$ the corresponding ideal estimator values. all quantities are in nats.

	$I(Y; A^1 \mid A^2)$			$I(Y; A^1)$			SD of Agent 1		
Case	$\star$	$\mathrm{InfoNCE}^\star$	InfoNCE	$\star$	$\mathrm{CLUB}^\star$	CLUB	$\star$	$\widehat{\mathcal{D}}^1$	$[\widehat{\mathcal{D}}^1]_+$
Individual	0.193	0.193	0.190	0.193	0.416	0.416	0.000	-0.226	0.000
Synergy	0.193	0.193	0.190	0.000	0.000	0.000	0.193	0.190	0.190
Teammate-only	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

4 Experiments

We evaluate SDM on four cooperative MARL benchmarks: MPE (Mordatch & Abbeel, 2017; Lowe et al., 2017), MAMuJoCo (Peng et al., 2021), SMAC (Samvelyan et al., 2019), and SMACv2 (Ellis et al., 2023). We compare against representative on-policy CTDE methods, including MAPPO (Yu et al., 2022), HAPPO, HATRPO, and HAA2C (Zhong et al., 2024), using the HARL framework (Zhong et al., 2024). To assess the coordination signal itself, we further compare with representative action–action influence (Jaques et al., 2019; Kim et al., 2020; Hou et al., 2025b) and action–state influence methods (Wang et al., 2019; Liu et al., 2023; Li et al., 2024), whose influence objectives are implemented within the same HARL-based on-policy framework for a controlled comparison. All methods use identical shared settings and random seeds, and results are reported as mean $\pm$ standard deviation. Full experimental details are provided in Appendix D.

4.1 Motivating Example

To illustrate the meaning of synergistic dependency in Eq. 4 and the effect of conservative dependency estimation, we first examine a simple synergy example. See Appendix D.3 for details.

XOR Example. As illustrated in Fig. 2(a), we consider two independent binary actions $A^1, A^2 \sim \mathrm{Bernoulli}(1/2)$ and three dependency structures: *Individual*, $\widetilde{Y} = A^1$; *Synergy*, $\widetilde{Y} = A^1 \oplus A^2$; and *Teammate-only*, $\widetilde{Y} = A^2$. To introduce stochasticity, the observed return is $Y = \widetilde{Y} \oplus \epsilon$, with $\epsilon \sim \mathrm{Bernoulli}(\eta)$ and $\eta = 0.2$. Let $C(\eta) = \log 2 - H_b(\eta)$, where $H_b(p) = -p \log p - (1-p)\log(1-p)$ is the binary entropy. For agent 1 with $\beta = 1$, the closed-form values of $(I(Y; A^1 \mid A^2), I(Y; A^1), \mathcal{D}^1)$ are $(C, C, 0)$, $(C, 0, C)$, and $(0, 0, 0)$ for the Individual, Synergy, and Teammate-only cases, respectively.

As shown in Table 2, conditional InfoNCE closely recovers the analytical interaction-conditioned dependency, benefiting from the binary action space where the sole alternative action forms an exact negative candidate. CLUB overestimates the individual dependency in the Individual case, but closely matches its analytical CLUB value, indicating bound looseness rather than estimation error. Importantly, this overestimation makes the resulting SD estimate conservative. The Individual case yields $\widehat{\mathcal{D}}^1 = -0.226$ and is suppressed by $[\widehat{\mathcal{D}}^1]_+$, whereas the Synergy case retains a positive estimate of 0.190, close to the analytical value 0.193. Overall, the conservative estimator suppresses non-synergistic effects while preserving task-relevant dependency that emerges through interaction.

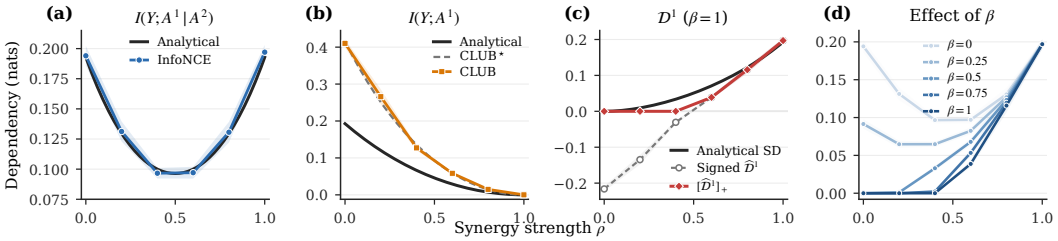

Figure 3: Continuous transition from the Individual ($\rho = 0$) to the Synergy ($\rho = 1$) regime. (a) Interaction-conditioned dependency. (b) Individual dependency with analytical and learned upper bounds. (c) Analytical SD and its conservative estimate. (d) Effect of β on the positive SD signal.

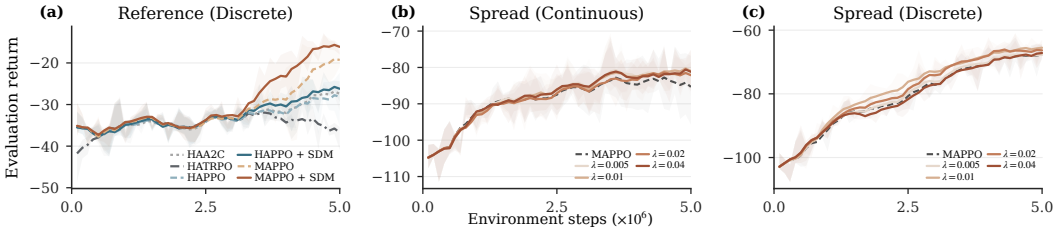

Figure 4: MPE learning curves and sensitivity to the SDM coefficient λ. (a) Performance on Reference with discrete actions. (b–c) MAPPO+SDM with different λ on Spread.

Continuous Synergy Transition. We further test whether SD tracks synergy strength continuously. As shown in Fig. 2(b), we sample $\varphi \sim \text{Bernoulli}(\rho)$ and set $\widetilde{Y} = A^1$ if $\varphi = 0$, and $\widetilde{Y} = A^1 \oplus A^2$ otherwise. Thus, $\rho \in [0, 1]$ interpolates from the Individual ($\rho = 0$) to the Synergy ($\rho = 1$) regime. The corresponding dependencies are $I(Y; A^1 \mid A^2) = \frac{1}{2}[\log 2 - H_b(\eta)] + \frac{1}{2}[\log 2 - H_b(\eta + \rho(1 - 2\eta))]$, and $I(Y; A^1) = \log 2 - H_b\big(\eta + \frac{\rho}{2}(1 - 2\eta)\big)$. Hence, $\mathcal{D}^1_\beta(\rho) = I(Y; A^1 \mid A^2) - \beta I(Y; A^1)$.

Figure 3 shows that the interaction-conditioned dependency alone is insufficient to characterize synergy, as $I(Y; A^1 \mid A^2)$ varies non-monotonically with ρ. In contrast, for $\beta = 1$, SD increases with the strength of the underlying synergistic interaction. Conservative estimation further suppresses weak or ambiguous interaction effects, while β controls the penalty on individual task dependency.

4.2 Main Results

MPE. We evaluate SDM on Reference, Speaker Listener, and Spread under both continuous and discrete action spaces. SDM is integrated with HAPPO and MAPPO and compared with HAA2C, HATRPO, and the corresponding backbones. Across all backbone comparisons, SDM improves performance in 10/12 cases and achieves the best result in five of the six settings. Figure 4(a) shows that SDM improves the learning performance of both HAPPO and MAPPO on discrete Reference. SDM also consistently improves both backbones on Spread. Figures 4(b–c) further show that SDM remains effective across a range of λ on Spread under both action spaces, indicating robustness to the strength of the synergy signal. Full results are provided in the Appendix E.1.

MAMuJoCo. We further evaluate SDM on seven MAMuJoCo tasks. Figure 5 shows learning curves on three representative tasks together with the normalized IQM aggregated over all seven tasks (Gorsane et al., 2022). SDM improves both HAPPO and MAPPO on the representative tasks and achieves a higher aggregate IQM. In the aggregate results, MAPPO with SDM reaches a performance level comparable to HAPPO. on Swimmer 10×2, showing that the gains persist in tasks with more agents. Additional results are provided in Appendix E.2.

We additionally compare with HAA2C and HATRPO, as well as action–action influence (AA) (Jaques et al., 2019; Kim et al., 2020; Hou et al., 2025b) and action–state influence (AS) (Wang et al., 2019; Liu et al., 2023; Li et al., 2024). For a controlled comparison, AA and AS are im-

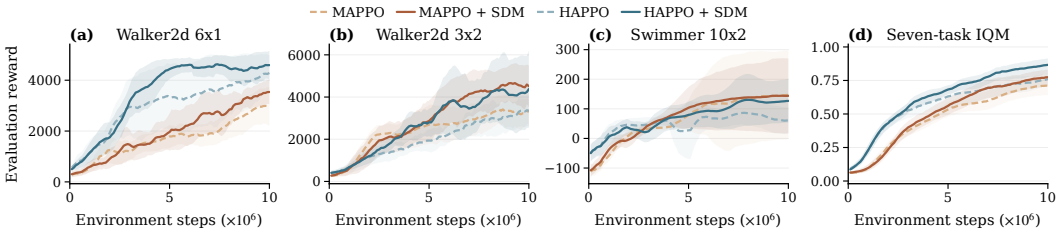

Figure 5: Performance on MAMuJoCo. Panels (a) to (c) show evaluation returns on three representative tasks, while (d) reports the normalized IQM across seven tasks.

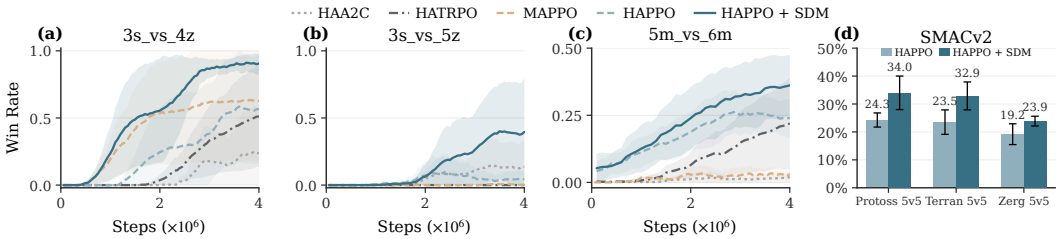

Figure 6: Performance on SMAC and SMACv2. Panels (a)–(c) show evaluation win rates on three SMAC scenarios, while panel (d) reports final-five evaluation win rates on three SMACv2 tasks.

Table 3: Final performance on six MAMuJoCo tasks. Results report the final five evaluations.

Method	Ant 2x4	Ant 4x2	HalfCheetah 6x1	Walker2d 2x3	Walker2d 3x2	Walker2d 6x1
HAA2C	4618.72 ± 362.75	4850.55 ± 222.21	3606.72 ± 1401.99	2034.75 ± 1255.87	3452.67 ± 535.71	1774.82 ± 312.49
HATRPO	5670.54 ± 50.37	5269.07 ± 163.00	4105.60 ± 782.77	3378.28 ± 418.75	4015.20 ± 622.36	3196.87 ± 678.71
MAPPO	5866.79 ± 55.63	5823.97 ± 371.97	6031.40 ± 171.79	3208.51 ± 600.54	3238.85 ± 544.21	2917.87 ± 694.60
MAPPO+SDM	**5905.72** ± 243.78	**6011.35** ± 175.29	**6118.02** ± 297.94	**4010.43** ± 632.35	**4456.96** ± 945.89	**3669.90** ± 441.65
HAPPO	5594.27 ± 114.03	5787.96 ± 75.76	6388.61 ± 83.41	4211.43 ± 251.66	3380.08 ± 693.56	4292.75 ± 671.24
HAPPO+AA	5705.16 ± 35.58	5854.49 ± 44.17	6016.50 ± 365.41	2874.36 ± 844.86	2787.56 ± 1108.06	3488.65 ± 417.10
HAPPO+AS	5639.68 ± 93.31	5822.43 ± 89.13	5860.25 ± 119.56	3827.56 ± 557.47	3265.85 ± 1037.67	4181.01 ± 1337.86
HAPPO+SDM	**5797.71** ± 73.72	**5978.19** ± 61.74	**6648.58** ± 129.02	**5071.44** ± 902.06	**4596.62** ± 2015.02	**4568.77** ± 613.47

plemented on HAPPO with the same advantage-gating pipeline as SDM, while the critic uses only environment returns. See Appendix D.4. Table 3 reports the final performance on all six tasks. Both HAPPO+SDM and MAPPO+SDM improve their corresponding backbones on every task. Moreover, an SDM variant achieves the best performance on all six tasks: MAPPO+SDM performs best on the two Ant tasks, while HAPPO+SDM achieves the highest returns on HalfCheetah and all three Walker2d tasks. SDM also shows a clear advantage over the AA and AS influence baselines, further supporting the effectiveness of explicitly optimizing synergistic dependency.

SMAC and SMACv2. We further evaluate SDM on SMAC and SMACv2 (Samvelyan et al., 2019; Ellis et al., 2023), which assess cooperative micromanagement under partial observability. We instantiate SDM on HAPPO and compare it with representative baselines under identical training and evaluation protocols. On SMAC, we consider 3s_vs_4z, 3s_vs_5z, and 5m_vs_6m. As shown in Fig. 6(a–c), HAPPO+SDM achieves the highest final win rate across all three scenarios. On SMACv2, we evaluate Protoss 5v5, Terran 5v5, and Zerg 5v5. Figure 6(d) shows that SDM improves the final win rate on all three tasks, with the largest gain on Protoss.

Ablation and Sensitivity Analysis. Figure 7(a) confirms the contributions of both the individual dependency term and the advantage gate. Figure 7(b) studies the effect of the multi-step return horizon H, with $H = 3$ performing best among the evaluated settings. Figures 7(c)–(d) show that $\beta = 1$ is a strong default, while $\beta = 0.2$ also performs well for MAPPO+SDM, indicating that task-specific tuning of β can further improve performance. Accordingly, we adopt $H = 3$ and $\beta = 1$ as the default settings unless otherwise specified. Panels (e)–(h) further analyze the synergistic dependency estimator. InfoNCE (N)–CLUB (C) achieves strong and stable performance, while the

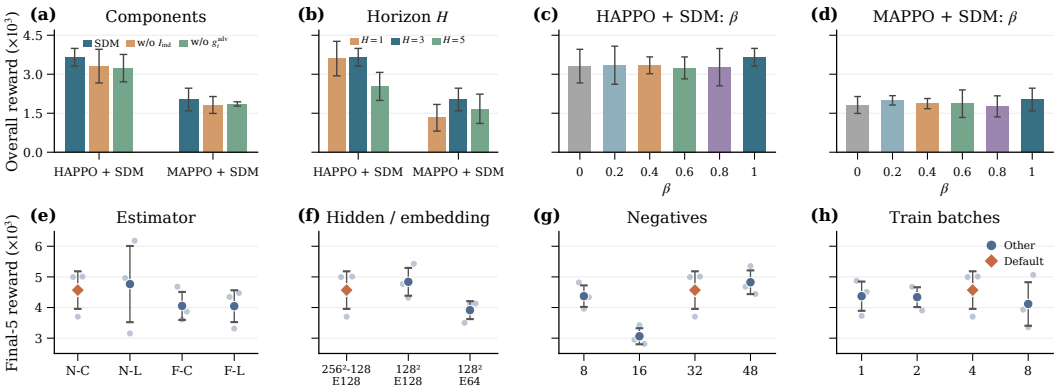

Figure 7: Ablation and sensitivity analysis of SDM. Panels (a)–(d) examine the component design, return horizon H, and different β values on Walker2d-v2 6x1. Panels (e)–(h) analyze the dependency estimator in terms of estimator type, network size, number of negatives, and training epochs.

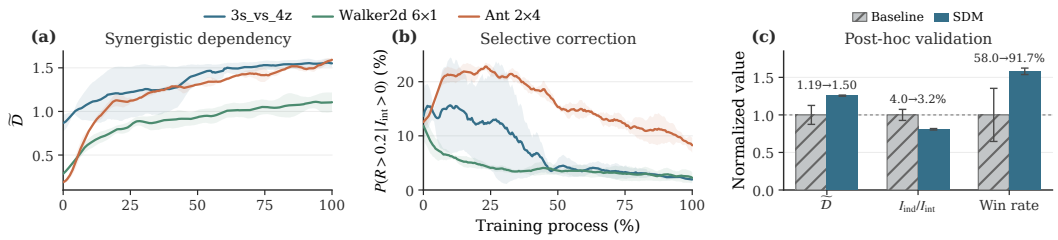

Figure 8: Mechanism analysis of SDM. (a) Estimated synergistic dependency during training across three tasks. (b) Fraction of positive interaction-conditioned dependencies requiring substantial individual correction. (c) Post-hoc validation on `3s_vs_4z`.

theoretically tighter FLO (F) and L1Out (L) alternatives offer no consistent empirical gains(Guo et al., 2022; Poole et al., 2019; Cheng et al., 2020). Increasing model capacity or training batches does not consistently improve performance, whereas more negative samples generally yield better performance. Overall, the estimator remains reasonably stable across different configurations.

Mechanism Analysis. Figure 8 reveals a consistent evolution of the SDM signal during training. Across tasks and seeds, the mean estimated synergistic dependency $\widetilde{\mathcal{D}}$ increases, while the fraction of samples requiring substantial individual correction, $P(R > 0.2 \mid I_{\text{int}} > 0)$, where $R = I_{\text{ind}}/(I_{\text{int}} + \epsilon)$, steadily decreases. This suggests that the individual term acts primarily as a selective correction early in training, while the interaction-conditioned dependency becomes increasingly dominant as training progresses. Panel (c) provides complementary post-hoc evidence on `3s_vs_4z`. Compared with HAPPO, SDM increases $\widetilde{\mathcal{D}}$ from 1.19 to 1.50, reduces $I_{\text{ind}}/I_{\text{int}}$ from 4.0% to 3.2%, and improves the final-five win rate from 58.0% to 91.7%. Together, these results indicate that SDM progressively isolates a stronger synergistic signal associated with improved cooperative performance. See Appendix E.3 for additional analyses.

5 Conclusions

We introduced Synergistic Dependency Maximization (SDM), which explicitly captures task-relevant dependencies arising from inter-agent interaction. SDM employs conservative estimation of synergistic dependency and incorporates the resulting signal into policy optimization. Experiments across four MARL benchmarks show consistent performance gains, while mechanism analyses reveal increasingly strong and reliable synergistic signals during training. SDM introduces additional dependency estimators and thus requires extra training resources. Future work will explore more efficient estimators for larger-scale multi-agent settings. Computational cost analysis is provided in Appendix E.4.

AI USE STATEMENT

Generative AI tools were used to assist with language refinement, and code debugging. The authors independently verified all technical content, experimental results, and analyses, and take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the algorithm and experimental implementation. An anonymized implementation and experimental configurations are available at `https://anonymous.4open.science/r/SDM-780E`. The code will be made publicly available upon acceptance.

REFERENCES

Adam B Barrett. Exploration of synergistic and redundant information sharing in static and dynamical gaussian systems. *Physical Review E*, 91(5):052802, 2015.

Sumeet Batra, Zhehui Huang, Aleksei Petrenko, Tushar Kumar, Artem Molchanov, and Gaurav S Sukhatme. Decentralized control of quadrotor swarms with end-to-end deep reinforcement learning. In *Conference on robot learning*, pp. 576–586. PMLR, 2022.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Nils Bertschinger, Johannes Rauh, Eckehard Olbrich, Jürgen Jost, and Nihat Ay. Quantifying unique information. *Entropy*, 16(4):2161–2183, 2014.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*, pp. 1779–1788. PMLR, 2020.

Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE transactions on intelligent transportation systems*, 21(3): 1086–1095, 2019.

Thomas M Cover, Joy A Thomas, and John Kieffer. *Elements of information theory*, volume 2. wiley New York, 1991.

Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Nicolaus Foerster, and Shimon Whiteson. SMACv2: An improved benchmark for cooperative multi-agent reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL `https://openreview.net/forum?id=5OjLGiJW3u`.

Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip HS Torr, Pushmeet Kohli, and Shimon Whiteson. Stabilising experience replay for deep multi-agent reinforcement learning. In *International conference on machine learning*, pp. 1146–1155. PMLR, 2017.

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.

Harsh Goel, Mohammad Omama, Behdad Chalaki, Vaishnav Tadiparthi, Ehsan Moradi Pari, and Sandeep Chinchali. R3dm: Enabling role discovery and diversity through dynamics models in multi-agent reinforcement learning. *arXiv preprint arXiv:2505.24265*, 2025.

Rihab Gorsane, Omayma Mahjoub, Ruan John de Kock, Roland Dubb, Siddarth Singh, and Arnu Pretorius. Towards a standardised performance evaluation protocol for cooperative marl. *Advances in Neural Information Processing Systems*, 35:5510–5521, 2022.

Qing Guo, Junya Chen, Dong Wang, Yuewei Yang, Xinwei Deng, Jing Huang, Larry Carin, Fan Li, and Chenyang Tao. Tight mutual information estimation with contrastive fenchel-legendre optimization. *Advances in Neural Information Processing Systems*, 35:28319–28334, 2022.

Yaqing Hou, Jie Kang, Haiyin Piao, Yifeng Zeng, Yew-Soon Ong, Yaochu Jin, and Qiang Zhang. Cooperative multiagent learning and exploration with min–max intrinsic motivation. *IEEE Transactions on Cybernetics*, 55(6):2852–2864, 2025a.

Yaqing Hou, Jie Kang, Haiyin Piao, Yifeng Zeng, Yew-Soon Ong, Yaochu Jin, and Qiang Zhang. Cooperative multiagent learning and exploration with min–max intrinsic motivation. *IEEE Transactions on Cybernetics*, 55(6):2852–2864, 2025b.

Dom Huh and Prasant Mohapatra. Multi-agent reinforcement learning: A comprehensive survey. *arXiv preprint arXiv:2312.10256*, 2023.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pp. 3040–3049. PMLR, 2019.

Yonghyeon Jo, Sunwoo Lee, Junghyuk Yeom, and Seungyul Han. Fox: Formation-aware exploration in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12985–12994, 2024.

Woojun Kim, Whiyoung Jung, Myungsik Cho, and Youngchul Sung. A maximum mutual information framework for multi-agent reinforcement learning. *arXiv preprint arXiv:2006.02732*, 2020.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251*, 2021.

Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:3991–4002, 2021.

Pengyi Li, Hongyao Tang, Tianpei Yang, Xiaotian Hao, Tong Sang, Yan Zheng, Jianye Hao, Matthew E Taylor, Wenyuan Tao, Zhen Wang, et al. Pmic: Improving multi-agent reinforcement learning with progressive mutual information collaboration. *arXiv preprint arXiv:2203.08553*, 2022.

Xinran Li, Zifan Liu, Shibo Chen, and Jun Zhang. Individual contributions as intrinsic exploration scaffolds for multi-agent reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 28387–28402, 2024.

Boyin Liu, Zhiqiang Pu, Yi Pan, Jianqiang Yi, Yanyan Liang, and Du Zhang. Lazy agents: A new perspective on solving sparse reward problem in multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 21937–21950. PMLR, 2023.

Jiarong Liu, Yifan Zhong, Siyi Hu, Haobo Fu, Qiang Fu, Xiaojun Chang, and Yaodong Yang. Maximum entropy heterogeneous-agent reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 57333–57374, 2024.

Joseph T Lizier, Nils Bertschinger, Jürgen Jost, and Michael Wibral. Information decomposition of target effects from multi-source interactions: Perspectives on previous, current and future work, 2018.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

William J McGill. Multivariate information transmission. *Psychometrika*, 19(2):97–116, 1954.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908*, 2017.

Ofir Nachum, Michael Ahn, Hugo Ponte, Shixiang Gu, and Vikash Kumar. Multi-agent manipulation via locomotion using hierarchical sim2real. *arXiv preprint arXiv:1908.05224*, 2019.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003.

Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Facmac: Factored multi-agent centralised policy gradients. *Advances in neural information processing systems*, 34:12208–12221, 2021.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International conference on machine learning*, pp. 5171–5180. PMLR, 2019.

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.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

Lukas M Schmidt, Johanna Brosig, Axel Plinge, Bjoern M Eskofier, and Christopher Mutschler. An introduction to multi-agent reinforcement learning and review of its application to autonomous mobility. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1342–1349. IEEE, 2022.

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.

Kyunghwan Son, Sungsoo Ahn, Roben Delos Reyes, Jinwoo Shin, and Yung Yi. Qtran++: Improved value transformation for cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2006.12010*, 2020.

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. *arXiv preprint arXiv:1706.05296*, 2017.

Nicholas Timme, Wesley Alford, Benjamin Flecker, and John M Beggs. Synergy, redundancy, and multivariate information measures: an experimentalist's perspective. *Journal of computational neuroscience*, 36(2):119–140, 2014.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. *arXiv e-prints*, pp. arXiv–2008, 2020a.

Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. *arXiv preprint arXiv:1910.05512*, 2019.

Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*, 2020b.

Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. Rode: Learning roles to decompose multi-agent tasks. *arXiv preprint arXiv:2010.01523*, 2020c.

Paul L Williams and Randall D Beer. Nonnegative decomposition of multivariate information. *arXiv preprint arXiv:1004.2515*, 2010.

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.

Haohan Yu, Jinmiao Cong, Shengzhi Wang, Lu Wang, and Chanjuan Liu. Magic: Multi-step advantage-gated causal influence for multi-agent reinforcement learning. *arXiv preprint arXiv:2605.01805*, 2026.

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.

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. URL `http://jmlr.org/papers/v25/23-0488.html`.

A More Related Works

Partial Information Decomposition. Characterizing information that emerges only through the joint consideration of multiple sources is a longstanding problem in multivariate information theory, with early formulations based on interaction information and co-information (McGill, 1954). Partial Information Decomposition (PID) later provided a more explicit framework for separating the information that multiple sources contribute about a target into redundant, unique, and synergistic components (Williams & Beer, 2010; Bertschinger et al., 2014). This perspective has since motivated extensive studies of synergy and redundancy in multivariate systems (Timme et al., 2014; Barrett, 2015; Lizier et al., 2018). For two sources A^i and A^{-i} and target Y, PID decomposes their joint information as $I(Y; A^i, A^{-i}) = R + U_i + U_{-i} + \text{Syn}$, with $I(Y; A^i) = R + U_i$ and $I(Y; A^i \mid A^{-i}) = U_i + \text{Syn}$. It follows that $I(Y; A^i \mid A^{-i}) - I(Y; A^i) = \text{Syn} - R$, showing that the increase in dependency revealed by teammate context exactly captures synergy beyond redundancy. Under the conventional sign definition, this quantity corresponds to negative co-information (Bertschinger et al., 2014; Barrett, 2015).

This relation provides the information-theoretic foundation of SDM. Rather than measuring statistical dependence or influence alone, SDM uses the multi-step team return as the target and asks whether teammate context strengthens the task-relevant dependency of an agent's action beyond its individual dependency. By further conditioning on the current state, SDM isolates task-relevant dependency strengthened by teammate context beyond dependencies already explained by the environment context. In this way, SDM operationalizes the PID notion of net complementary information into a practical coordination objective for cooperative MARL. Moreover, because the criterion is expressed directly as a difference between the interaction-conditioned and individual dependencies, SDM avoids explicitly estimating the individual PID atoms and does not require choosing a particular redundancy functional (Williams & Beer, 2010; Bertschinger et al., 2014).

Influence-Based Coordination in MARL. Existing explicit coordination methods mainly characterize either action–action dependence or action–state influence. Social Influence (Jaques et al., 2019) measures how an agent's action changes the action distribution of its teammates through a counterfactual influence objective. VM3-AC (Kim et al., 2020) promotes coordination by maximizing mutual information among agents' actions through a variational objective. E2M (Hou et al.,

2025a) combines social-influence maximization with surprise minimization to encourage coordinated exploration. Influence-based exploration (Wang et al., 2019) quantifies how individual actions influence subsequent environment transitions and uses this influence to guide exploration. LAIES (Liu et al., 2023) measures the causal effect of an agent's action on external states and constructs counterfactual intrinsic rewards to mitigate lazy agents. ICES (Li et al., 2024) evaluates individual contributions through Bayesian surprise over global latent-state transitions and uses them as exploration signals. PMIC (Li et al., 2022) characterizes collaboration through mutual information between global states and joint actions while distinguishing superior and inferior trajectories. MAGIC (Yu et al., 2026) estimates the multi-step causal effect of an agent's current action on its teammates' future states through counterfactual rollouts, and uses advantage-based gating to reinforce task-beneficial effects. While these methods progressively capture behavioral dependence, environmental influence, they do not explicitly test whether teammate context strengthens an agent's dependency on the team objective. SDM instead contrasts $I(Y; A^i \mid S, A^{-i})$ with $I(Y; A^i \mid S)$, directly isolating the additional task-relevant dependency revealed by teammate actions and thereby characterizing the interaction coupling underlying synergistic coordination.

Role-Based and Complementary Cooperation. Another line of cooperative MARL research promotes cooperation through role specialization and behavioral diversity. ROMA (Wang et al., 2020b) learns identifiable and specialized roles, while RODE (Wang et al., 2020c) structures decision making through role-based action decomposition. Recent work further explores diversity and specialization as mechanisms for coordinated behavior (Goel et al., 2025). SDM can also promote heterogeneous yet coordinated behavior without explicitly imposing roles or diversity objectives. By favoring task-relevant synergy over redundancy, SDM discourages overlapping contributions and places greater emphasis on complementary interactions among agents.

B Dependency Estimation Details

We consider four variational estimators for synergistic dependency. For the interaction-conditioned dependency, InfoNCE (Oord et al., 2018; Poole et al., 2019) provides a simple contrastive lower bound, while FLO (Guo et al., 2022) offers a theoretically tighter alternative through a more elaborate variational objective. For the individual dependency, CLUB (Cheng et al., 2020) estimates an upper bound through contrastive conditional likelihoods, whereas L1Out (Poole et al., 2019) provides a tighter leave-one-out upper bound. Combining the two lower- and upper-bound estimators yields four variants, InfoNCE–CLUB (N-C), InfoNCE–L1Out (N-L), FLO–CLUB (F-C), and FLO–L1Out (F-L). Although FLO and L1Out can provide tighter variational bounds, their increased estimation complexity does not translate into consistent empirical gains and can lead to less stable optimization in our experiments. In contrast, InfoNCE–CLUB provides a favorable balance between estimation reliability and computational efficiency, achieving strong and stable performance across our ablations. We therefore adopt N-C as the default estimator throughout the main experiments.

B.1 InfoNCE–CLUB Implementation

Our default implementation uses conditional InfoNCE (Oord et al., 2018; Poole et al., 2019) for the interaction-conditioned dependency and variational CLUB (Cheng et al., 2020) for the individual dependency. Both estimators are trained from the same on-policy trajectories used for policy optimization, without requiring additional environment interactions.

Interaction-conditioned dependency. For agent i, we denote the interaction context by $C_t^i = (S_t, A_t^{-i})$ and estimate $I(Y_t; A_t^i \mid C_t^i)$ using conditional InfoNCE. Given an observed tuple (Y_t, A_t^i, C_t^i), the executed action A_t^i forms the positive sample. We keep C_t^i fixed and sample $N-1$ alternative actions $\{\widetilde{A}_{t,n}^i\}_{n=1}^{N-1}$ independently from the behavior policy $\pi_{\text{old}}^i(\cdot \mid o_t^i)$. Let $f_\phi(Y, A^i, C^i)$ denote the learnable compatibility score. The per-sample contrastive loss is

$$\ell_{\text{NCE},t}^i = -\log \frac{\exp f_\phi(Y_t, A_t^i, C_t^i)}{\exp f_\phi(Y_t, A_t^i, C_t^i) + \sum_{n=1}^{N-1} \exp f_\phi(Y_t, \widetilde{A}_{t,n}^i, C_t^i)}.$$

The corresponding interaction-conditioned estimate is $I_{\text{int},t}^{i,\text{LB}} = \log N - \ell_{\text{NCE},t}^i$. Importantly, only the action of agent i is resampled, while the state and teammate actions remain unchanged. This

context-matched negative sampling prevents variations in S_t or A_t^{-i} from being absorbed into the dependency estimate and directly targets the additional information carried by A_t^i under the same teammate context.

Individual dependency. We estimate $I(Y_t; A_t^i \mid S_t)$ using variational CLUB. Specifically, a conditional density model $q_\psi(Y_t \mid S_t, A_t^i)$ is first fitted to observed samples by minimizing the negative log-likelihood

$$\mathcal{L}_{\mathrm{CLUB}} = -\mathbb{E}_t \left[\log q_\psi(Y_t \mid S_t, A_t^i) \right].$$

To estimate the individual dependency, we again construct matched negatives, but now only the state S_t is fixed. Alternative actions $\{\widetilde{A}_{t,m}^i\}_{m=1}^M$ are sampled from $\pi_{\mathrm{old}}^i(\cdot \mid o_t^i)$, yielding the per-sample estimate

$$I_{\mathrm{ind},t}^{i,\mathrm{UB}} = \log q_\psi(Y_t \mid S_t, A_t^i) - \frac{1}{M} \sum_{m=1}^{M} \log q_\psi(Y_t \mid S_t, \widetilde{A}_{t,m}^i).$$

When the learned conditional density sufficiently approximates the true conditional distribution, this variational construction provides an upper-bound approximation to the individual dependency. Unlike the InfoNCE estimator, teammate actions are not included in the conditioning variables, since this term is intended to capture the task-relevant information available from the individual action alone.

Conservative estimate. The two estimates are combined asymmetrically as $\widehat{D}_t^i = I_{\mathrm{int},t}^{i,\mathrm{LB}} - \beta I_{\mathrm{ind},t}^{i,\mathrm{UB}}$. Using a lower-bound estimate for the interaction-conditioned term and an upper-bound estimate for the individual term biases estimation against spuriously large synergistic dependency. The positive component $[\widehat{D}_t^i]_+$ is subsequently normalized and aggregated across agents before being incorporated into the policy advantage.

Negative sampling. All negative actions are drawn from the same behavior policy π_{old} that generated the on-policy trajectories. For discrete policies, negatives are sampled from the corresponding categorical action distribution, whereas for continuous policies they are sampled directly from the policy distribution. Consequently, negative construction requires only additional policy sampling and estimator forward passes, rather than extra environment interaction. We additionally provide batch-based and KNN-based negative construction strategies as alternative implementations, allowing SDM to accommodate different sampling schemes without changing the dependency estimation objective.

Estimator Architecture and Training. The conditional InfoNCE estimator consists of separate MLP encoders for the interaction context, individual action, and multi-step return, followed by an MLP score head. The CLUB estimator uses an MLP to parameterize a Gaussian conditional density over the multi-step return. The two estimators are optimized jointly with a dedicated Adam optimizer, independently of the actor and critic optimizers. At each training iteration, transitions collected from the on-policy rollout are used to construct the multi-step return targets and train the dependency estimators. Specifically, four mini-batches of size 256 are sampled from the rollout data, resulting in four gradient updates of the dependency estimators (`train_batches`=4). We use a warm-up period of 100 training iterations before incorporating the estimated synergistic dependency signal into policy optimization. In our default configuration, the estimator networks use hidden dimensions $[128, 128]$, an InfoNCE embedding dimension of 128, and 32 negative samples.

C ALGORITHM

Algorithm 1 Training Procedure of Synergistic Dependency Maximization (SDM)

Initialize agent policies $\{\pi_{\theta_i}\}_{i=1}^{N}$, centralized critic, and dependency estimators
for each training iteration **do**
  Collect an on-policy trajectory batch $\mathcal{D}$, where each agent samples $a_t^i \sim \pi_{\text{old}}^i(\cdot \mid o_t^i)$
  Construct the H-step team return y_t for each valid transition
  for $k = 1, \ldots, K_{\text{est}}$ **do**
    Sample a mini-batch $\mathcal{B} \subset \mathcal{D}$ and construct negative actions
    Compute the InfoNCE and CLUB objectives over all agents
    Update the dependency estimators
  end for
  for each valid transition t and agent i **do**
    Estimate the synergistic dependency

$$\widehat{\mathcal{D}}_t^i = I_{\text{int},t}^{i,\text{LB}} - \beta I_{\text{ind},t}^{i,\text{UB}}$$

    Retain and normalize $[\widehat{\mathcal{D}}_t^i]_+$ to obtain $\widetilde{\mathcal{D}}_t^i$
  end for
  Aggregate the agent-wise signals as $\widetilde{\mathcal{D}}_t = \frac{1}{N}\sum_{i=1}^{N} \widetilde{\mathcal{D}}_t^i$
  Compute the environment advantage $\mathcal{A}_{t,\text{env}}$ and the advantage gate $g_t^{\text{adv}} = \mathbb{I}[\mathcal{A}_{t,\text{env}} > 0]$
  Construct the SDM-augmented advantage

$$\widehat{\mathcal{A}}_{t,\text{SDM}} = \mathcal{A}_{t,\text{env}} + \lambda g_t^{\text{adv}} \widetilde{\mathcal{D}}_t$$

  Update the policies using $\widehat{\mathcal{A}}_{t,\text{SDM}}$ following the underlying MAPPO/HAPPO optimization
  Update the centralized critic using environment returns only
end for

D EXPERIMENTAL DETAILS

D.1 SOFTWARE AND HARDWARE SETUP

All experiments are implemented using the HARL framework (Zhong et al., 2024) and conducted on Linux servers equipped with NVIDIA GeForce RTX 3090, GeForce RTX 2080 Ti, or Tesla M40 GPUs. The reported results are aggregated across independent random seeds. Unless otherwise specified, all reported results are aggregated over three independent random seeds, $\{1, 2, 3\}$.

D.2 ENVIRONMENTS AND TASKS

We first use the motivating example to analyze the characteristics of synergistic dependency (SD), and then evaluate SDM on four cooperative multi-agent benchmarks: MPE (Mordatch & Abbeel, 2017; Lowe et al., 2017), MAMuJoCo (Peng et al., 2021), SMAC (Samvelyan et al., 2019), and SMACv2 (Ellis et al., 2023). Together, these benchmarks cover both discrete and continuous action spaces and a diverse range of coordination settings.

MPE. The Multi-Agent Particle Environment (MPE) (Lowe et al., 2017) is a particle-world benchmark for cooperative multi-agent decision making, where agents interact under decentralized observations through navigation and, in some tasks, communication. We consider three representative tasks, Reference, Speaker–Listener, and Spread, under both continuous and discrete action spaces, resulting in six evaluation settings.

MAMuJoCo. Multi-Agent MuJoCo (MAMuJoCo) (Peng et al., 2021) extends standard MuJoCo continuous-control tasks to cooperative multi-agent settings by partitioning the control dimensions of a single physical system among multiple agents. We evaluate seven tasks with different agent decompositions: `Ant 2x4`, `Ant 4x2`, `HalfCheetah 6x1`, `Walker2d 2x3`, `Walker2d`

3x2, and Walker2d 6x1. We additionally evaluate Swimmer 10x2 to examine scalability to a larger number of agents.

SMAC. The StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019) is a cooperative micromanagement benchmark built on StarCraft II. Each allied unit is controlled by an individual agent that selects actions from local observations, while global state information is available only during centralized training. Agents must coordinate movement, target selection, and attack behaviors to defeat enemy units controlled by the built-in game AI. The benchmark therefore presents a partially observable cooperative setting in which successful policies require effective coordination among decentralized agents. We evaluate three challenging scenarios: 3s_vs_4z, 3s_vs_5z, and 5m_vs_6m.

SMACv2. SMACv2 (Ellis et al., 2023) extends SMAC to provide a more stochastic and diverse evaluation of cooperative policies. Instead of relying on fixed scenarios, SMACv2 procedurally generates team compositions and initial unit positions for each episode, and uses unit-specific sight and attack ranges. Consequently, agents must adapt their coordination strategies to varying team configurations and spatial layouts rather than relying on fixed open-loop behavior. We evaluate the symmetric 5v5 settings for all three StarCraft II races: Protoss, Terran, and Zerg. Each race contains multiple unit types with different combat characteristics, further increasing the diversity of coordination patterns encountered across episodes.

D.3 Detailed Analysis of the Motivating Example

XOR Example. We consider two independent binary actions $A^1, A^2 \sim \mathrm{Bernoulli}(1/2)$ and three dependency structures: *Individual*, $\widetilde{Y} = A^1$; *Synergy*, $\widetilde{Y} = A^1 \oplus A^2$; and *Teammate-only*, $\widetilde{Y} = A^2$. The observed return is $Y = \widetilde{Y} \oplus \epsilon$, where $\epsilon \sim \mathrm{Bernoulli}(\eta)$. Let

$$C(\eta) = \log 2 - H_b(\eta), \qquad H_b(p) = -p \log p - (1-p)\log(1-p).$$

For agent 1, the corresponding dependencies admit closed-form solutions. In the Individual case, A^2 is independent of (A^1, Y), giving

$$I(Y; A^1 \mid A^2) = I(Y; A^1) = C(\eta),$$

and therefore

$$\mathcal{D}_\beta^1 = (1-\beta)C(\eta).$$

In the Synergy case, conditioning on A^2 reduces the relation between A^1 and Y to a binary symmetric channel, whereas marginalizing over the uniform A^2 makes Y independent of A^1. Hence,

$$I(Y; A^1 \mid A^2) = C(\eta), \qquad I(Y; A^1) = 0, \qquad \mathcal{D}_\beta^1 = C(\eta).$$

In the Teammate-only case, Y is independent of A^1 both marginally and conditionally, yielding

$$I(Y; A^1 \mid A^2) = I(Y; A^1) = \mathcal{D}_\beta^1 = 0.$$

Thus, for $\beta = 1$,

$$\big(I(Y; A^1 \mid A^2), I(Y; A^1), \mathcal{D}_1^1\big) = \begin{cases} (C, C, 0), & \text{Individual}, \\ (C, 0, C), & \text{Synergy}, \\ (0, 0, 0), & \text{Teammate-only}, \end{cases}$$

where $C = C(\eta)$. For $\eta = 0.2$, $C \approx 0.193$ nats. Similarly, for agent 2 with $\beta = 1$,

$$\big(I(Y; A^2 \mid A^1), I(Y; A^2), \mathcal{D}_1^2\big) = \begin{cases} (0, 0, 0), & \text{Individual}, \\ (C, 0, C), & \text{Synergy}, \\ (C, C, 0), & \text{Teammate-only}. \end{cases}$$

Thus, the Synergy case yields the same positive synergistic dependency for both agents, whereas the Individual and Teammate-only cases contain only agent-specific dependency and are suppressed when $\beta = 1$.

Ideal InfoNCE and CLUB Estimates. We further derive the ideal values of the two estimators used by SDM. For conditional InfoNCE (Oord et al., 2018; Poole et al., 2019), the observed action A^i is treated as the positive candidate, while the only alternative binary action $1 - A^i$ serves as the negative candidate. Hence, the two candidates exhaust the binary action space. Under a population-optimal critic, the corresponding softmax recovers the posterior $p(A^i \mid Y, A^{-i})$, and the optimal contrastive loss is

$$\mathcal{L}^{\star}_{\mathrm{NCE}} = H(A^i \mid Y, A^{-i}).$$

Since A^i is uniform and independent of A^{-i}, $H(A^i \mid A^{-i}) = \log 2$, and therefore

$$\mathrm{InfoNCE}^{\star} = \log 2 - \mathcal{L}^{\star}_{\mathrm{NCE}} = I(Y; A^i \mid A^{-i}).$$

Thus, for the Individual and Synergy cases, where the interaction-conditioned dependency reduces to a binary symmetric channel with crossover probability η,

$$\mathrm{InfoNCE}^{\star} = C(\eta) = \log 2 - H_b(\eta),$$

whereas $\mathrm{InfoNCE}^{\star} = 0$ when Y and A^i are conditionally independent given A^{-i}. This exact equality holds here because the two contrastive candidates exhaust the binary action space; in general, InfoNCE provides only a lower bound on the conditional mutual information.

For CLUB (Cheng et al., 2020), assuming an ideal conditional model $q^{\star}(Y \mid A^i) = p(Y \mid A^i)$, the population estimate is

$$\mathrm{CLUB}^{\star} = \mathbb{E}_{p(a^i, y)} \left[\log p(y \mid a^i)\right] - \mathbb{E}_{p(a^i)p(y)} \left[\log p(y \mid a^i)\right].$$

For a binary symmetric channel with crossover probability η, this becomes

$$\begin{aligned}\mathrm{CLUB}^{\star} &= (1-\eta)\log(1-\eta) + \eta \log \eta \\ &\quad - \frac{1}{2}\log(1-\eta) - \frac{1}{2}\log \eta \\ &= \left(\frac{1}{2} - \eta\right) \log \frac{1-\eta}{\eta} \triangleq U(\eta).\end{aligned}$$

When Y and A^i are independent, $\mathrm{CLUB}^{\star} = 0$. Accordingly, for agent 1,

$$\left(\mathrm{InfoNCE}^{\star}, \mathrm{CLUB}^{\star}\right) = \begin{cases} (C, U), & \text{Individual}, \\ (C, 0), & \text{Synergy}, \\ (0, 0), & \text{Teammate-only}, \end{cases}$$

whereas for agent 2,

$$\left(\mathrm{InfoNCE}^{\star}, \mathrm{CLUB}^{\star}\right) = \begin{cases} (0, 0), & \text{Individual}, \\ (C, 0), & \text{Synergy}, \\ (C, U), & \text{Teammate-only}. \end{cases}$$

For $\eta = 0.2$, we obtain $C(\eta) \approx 0.193$ and $U(\eta) \approx 0.416$ nats.

Continuous Synergy Transition. We further consider a continuous transition between individual and synergistic dependency. Let $\varphi \sim \mathrm{Bernoulli}(\rho)$ and define

$$\widetilde{Y} = \begin{cases} A^1, & \varphi = 0, \\ A^1 \oplus A^2, & \varphi = 1, \end{cases} \qquad \rho \in [0, 1].$$

When conditioning on $A^2 = 0$, the effective crossover probability between A^1 and Y is η. For $A^2 = 1$, it becomes

$$q_\rho = (1-\rho)\eta + \rho(1-\eta) = \eta + \rho(1-2\eta).$$

Since A^2 is uniform,

$$I(Y; A^1 \mid A^2) = \frac{1}{2}C(\eta) + \frac{1}{2}C(q_\rho).$$

Without conditioning on A^2, marginalizing over the two branches gives the effective crossover probability

$$\bar{q}_\rho = (1-\rho)\eta + \frac{\rho}{2} = \eta + \frac{\rho}{2}(1-2\eta),$$

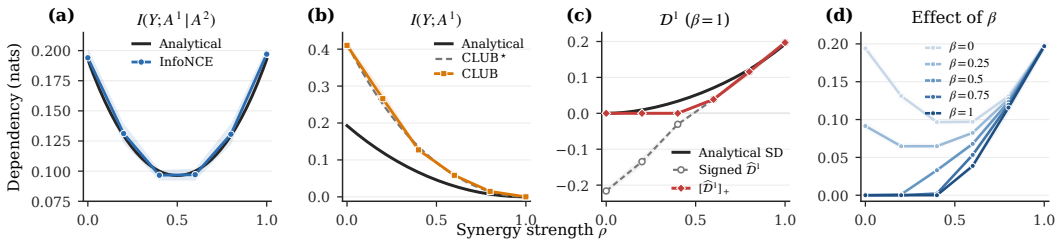

Figure 9: Continuous transition from individual to synergistic dependency, reproduced from Fig. 3 for convenience. (a) Interaction-conditioned dependency. (b) Individual dependency with analytical and learned upper bounds. (c) Analytical SD and its conservative estimate. (d) Effect of β on the positive SD signal.

and therefore

$$I(Y;A^1) = C(\bar{q}_\rho).$$

The resulting synergistic dependency is

$$\mathcal{D}^1_\beta(\rho) = \frac{1}{2}C(\eta) + \frac{1}{2}C(q_\rho) - \beta C(\bar{q}_\rho).$$

For $\beta = 1$, since $\bar{q}_\rho = (\eta + q_\rho)/2$,

$$\mathcal{D}^1_1(\rho) = H_b(\bar{q}_\rho) - \frac{1}{2}H_b(\eta) - \frac{1}{2}H_b(q_\rho).$$

By the concavity of binary entropy, $\mathcal{D}^1_1(\rho) \geq 0$, with

$$\mathcal{D}^1_1(0) = 0, \qquad \mathcal{D}^1_1(1) = C(\eta),$$

showing that SD continuously increases from purely individual dependency to pure synergy in this construction.

Figure 9 provides a detailed view of how the dependency terms evolve as the interaction transitions from individual to synergistic. In panel (a), $I(Y; A^1 \mid A^2)$ is high at both endpoints but decreases and then increases with ρ, showing that interaction-conditioned dependency alone cannot distinguish individual from synergistic dependence. Panel (b) shows that $I(Y; A^1)$ decreases from $C(\eta)$ to zero as the dependency becomes increasingly determined by the joint action rather than A^1 alone, while the CLUB estimate exhibits the same overall trend. Combining the two terms, panel (c) shows that $\mathcal{D}^1_1$ increases from zero to $C(\eta)$ with ρ, isolating the additional task-relevant dependency revealed by teammate context. The conservative estimate is more selective, suppressing weak interaction effects and retaining a positive SD signal only when the synergistic component becomes sufficiently strong. Finally, panel (d) illustrates the role of β: smaller values retain more individual task dependency and therefore produce a positive signal even near the Individual regime, whereas $\beta = 1$ fully removes the individual component and provides the clearest separation between individual and synergistic dependence.

D.4 Baselines and Implementation

MARL Baselines. We compare SDM against representative on-policy CTDE methods implemented in the HARL framework (Zhong et al., 2024), including HAA2C, HATRPO, HAPPO, and MAPPO. MAPPO (Yu et al., 2022) combines decentralized policies with centralized value estimation, while HATRPO and HAPPO (Kuba et al., 2021; Liu et al., 2024) employ sequential policy updates for heterogeneous-agent optimization. We instantiate SDM on both MAPPO and HAPPO, denoted as MAPPO+SDM and HAPPO+SDM, respectively. In both cases, SDM modifies only the actor-side policy-improvement signal through the augmented advantage, while preserving the original policy optimization procedure and environment-return-based critic training.

Influence-based Baselines and Controlled Implementation. To directly compare SD with existing explicit coordination signals, we construct controlled influence-based baselines under a unified

HARL implementation. PMIC (Li et al., 2022) targets task-aware policy optimization rather than cooperative exploration, following a different motivation from our work. MAGIC (Yu et al., 2026) remains within the action–state influence family and had no public implementation available at the time of our experiments. We therefore compare coordination objectives under the same HAPPO backbone and optimization pipeline. AA represents action–action influence objectives that promote statistical dependence among agents' actions (Jaques et al., 2019; Kim et al., 2020; Hou et al., 2025b), while AS-INF represents action–state influence objectives that quantify the influence of an agent's action on subsequent state dynamics (Wang et al., 2019; Liu et al., 2023; Li et al., 2024). Both influence-based baselines are implemented on top of HAPPO using the same policy-optimization pipeline as SDM. Specifically, AA uses the conditional mutual information $I(A_{t+1}^{-i}; A_t^i \mid S_t, A_t^{-i})$ to capture action–action dependence, while AS-INF estimates $I(\Delta S_{t+1}; A_t^i \mid S_t, A_t^{-i})$ to characterize the influence of an agent's action on subsequent state changes. For a controlled comparison, the resulting agent-wise influence scores are processed identically to SD: they are rectified, normalized, averaged across agents, and incorporated into the shared actor advantage through the same advantage-gating mechanism. Meanwhile, the centralized critic is trained solely on environment returns. This design isolates the effect of the coordination signal from differences in the underlying optimization pipeline. All methods are implemented within HARL using the same environment settings, training budgets, evaluation protocols, and random seeds for each task.

D.5 Training and Evaluation Protocol

For each task, all methods are trained under the same environment configuration, training budget, and evaluation schedule. Independent random seeds are used for repeated runs, and the same set of seeds is shared across compared methods to reduce variability arising from initialization and environment stochasticity. During training, policies are periodically evaluated without updating model parameters, and the corresponding episode return or win rate is recorded.

We use episode return as the primary metric for MPE and MAMuJoCo, and win rate for SMAC and SMACv2. Learning curves report the mean performance across random seeds, with variability shown by the corresponding standard deviation. For final-performance comparisons, we average the last five evaluation checkpoints of each run before aggregating across seeds, which reduces sensitivity to single-checkpoint fluctuations. The same protocol is applied to all methods within each task.

For cross-task comparison, we additionally report the normalized interquartile mean (IQM) (Gorsane et al., 2022). We first normalize performance across tasks to account for differences in return scales, and then compute the IQM over the normalized scores. IQM averages the central 50% of the score distribution, making the aggregate metric less sensitive to extreme scores than the mean while providing a statistically efficient summary of overall performance across tasks.

D.6 Hyperparameters

We follow the standard HARL configurations for the HAPPO and MAPPO backbones, as summarized in Table 4. The additional hyperparameters introduced by SDM are reported in Table 5. Unless otherwise specified, the default SDM settings are fixed across tasks. For each task, we first tune the SDM coefficient λ with β fixed to 1, and then tune β while keeping the selected λ fixed. In practice, we find that fixing $\beta = 1$ and tuning only λ already yields strong performance on most tasks, while further tuning β provides additional flexibility when needed.

E Additional Experimental Results

This section presents complete task-wise results and additional evaluations across the considered benchmarks to complement the main experimental analysis.

E.1 Results on MPE

We provide the complete results on MPE, which are omitted from the main text for space. We consider three cooperative scenarios, *Reference*, *Speaker Listener*, and *Spread*, under both continuous

Table 4: Main hyperparameters of the HAPPO and MAPPO backbones.

Hyperparameter	HAPPO	MAPPO
Hidden dimensions	$[128, 128]$	$[128, 128]$
Activation	ReLU	ReLU
Feature normalization	Yes	Yes
Initialization	Orthogonal	Orthogonal
Actor learning rate	5×10^{-4}	5×10^{-4}
Critic learning rate	5×10^{-4}	5×10^{-4}
PPO clipping ratio	0.2	0.2
Actor update epochs	5	5
Critic update epochs	5	5
Actor mini-batches	1	1
Critic mini-batches	1	1
Entropy coefficient	0.01	0.01
Value loss coefficient	1.0	1.0
Discount factor γ	0.99	0.99
GAE parameter	0.95	0.95
Maximum gradient norm	10.0	10.0
Value normalization	Yes	Yes
Parameter sharing	No	Yes

Table 5: Default SDM hyperparameters and values considered in sensitivity analyses.

Hyperparameter	Default	Values considered
Multi-step horizon H	3	$\{1, 3, 5\}$
Dependency coefficient β	1.0	$\{0, 0.25, 0.5, 0.75, 1.0\}$
SDM coefficient λ	0.01	$\{0.005, 0.01, 0.02, 0.04\}$
Estimator learning rate	5×10^{-4}	–
Hidden dimensions	$[128, 128]$	$\{[128, 128], [256, 256, 128]\}$
Embedding dimension	128	$\{64, 128\}$
Activation	SiLU	–
Layer normalization	Yes	–
InfoNCE temperature	0.2	–
Number of negatives	32	$\{8, 16, 32, 64\}$
Estimator updates per iteration	4	$\{1, 2, 4, 8\}$
Mini-batch size	256	–
Warm-up iterations	100	–
Maximum gradient norm	5.0	–

and discrete action spaces, resulting in six tasks. SDM is instantiated on both HAPPO and MAPPO and compared with HAA2C, HATRPO, and the corresponding backbones. All methods are trained for 5M environment steps under the same training and evaluation protocol.

Table 6 reports the final performance on all six tasks. Across the 12 backbone comparisons, SDM improves its underlying backbone in 10 cases and achieves the best performance in five of the six tasks. The improvement is particularly pronounced on discrete *Reference*, where MAPPO+SDM improves the return from -20.01 to -16.12. On *Spread*, SDM improves both HAPPO and MAPPO under continuous and discrete action spaces. The gains on *Speaker Listener* are smaller, where the backbone methods already achieve similar performance. Overall, these results show that SDM provides consistent benefits across different policy-optimization backbones and action-space settings.

E.2 Results on MAMuJoCo

Figure 10 provides the complete learning curves of HAPPO, MAPPO, and their SDM variants on the six MAMuJoCo tasks considered in the main experiments, together with the *Swimmer* 10×2 task used to evaluate scalability. While the main text reports representative task-wise curves and final-performance comparisons, the results here provide a more complete view of the learning dynamics across different agent decompositions.

Table 6: Final performance on MPE after 5M environment steps. Results are reported as mean ± standard deviation across random seeds. Higher values are better. Best results are shown in **bold**.

	Reference		Speaker Listener		Spread	
Method	Continuous	Discrete	Continuous	Discrete	Continuous	Discrete
HAA2C	-41.10 ± 2.05	-27.57 ± 1.78	-31.35 ± 5.63	-12.24 ± 3.27	-109.10 ± 2.28	-72.85 ± 1.36
HATRPO	-23.64 ± 1.38	-36.31 ± 2.06	-10.19 ± 2.50	-15.09 ± 3.49	-79.83 ± 1.35	-66.35 ± 0.64
MAPPO	-13.25 ± 0.82	-20.01 ± 1.95	$\mathbf{-9.56} \pm 2.44$	-12.04 ± 3.23	-84.82 ± 2.25	-68.13 ± 1.17
MAPPO+SDM	$\mathbf{-13.22} \pm 0.87$	$\mathbf{-16.12} \pm 1.26$	-9.67 ± 2.43	$\mathbf{-12.02} \pm 3.25$	$\mathbf{-81.11} \pm 2.11$	$\mathbf{-66.04} \pm 1.31$
HAPPO	$\mathbf{-16.27} \pm 1.24$	-28.81 ± 2.25	-9.63 ± 2.44	-12.07 ± 3.20	-67.90 ± 1.37	-71.55 ± 0.38
HAPPO+SDM	-16.43 ± 1.35	$\mathbf{-26.54} \pm 1.79$	$\mathbf{-9.59} \pm 2.41$	$\mathbf{-12.04} \pm 3.24$	$\mathbf{-67.09 \pm 1.51}$	$\mathbf{-70.78} \pm 0.72$

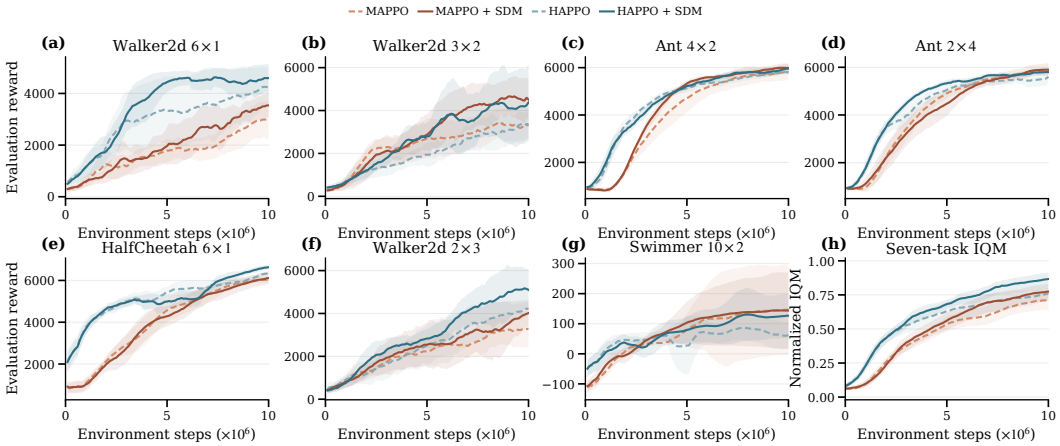

Figure 10: Complete learning curves on MAMuJoCo. Panels (a)–(g) compare HAPPO, MAPPO, and their SDM variants on six main MAMuJoCo tasks and the additional *Swimmer* 10×2 scalability task. Solid lines denote SDM variants and dashed lines denote their corresponding backbones; shaded regions indicate variability across random seeds. Panel (h) reports the normalized interquartile mean (IQM) aggregated across all seven tasks, with shaded regions showing 95% bootstrap confidence intervals.

Across the considered tasks, incorporating SDM consistently improves the performance of the underlying HAPPO and MAPPO backbones, with the gains persisting throughout training on most environments. The improvements are observed across *Ant*, *HalfCheetah*, and *Walker2d*, indicating that the benefit of synergistic dependency is not tied to a particular morphology or agent decomposition. SDM also improves both backbones on *Swimmer* 10×2, suggesting that the dependency-based coordination signal remains effective as the number of agents increases.

To summarize performance across tasks with different reward scales, we additionally report the normalized interquartile mean (IQM) in Fig. 10(h). Returns are normalized separately for each task before aggregation, and the IQM is computed over the pooled task–seed scores. The shaded region denotes the 95% bootstrap confidence interval. The aggregate curves show that both HAPPO+SDM and MAPPO+SDM improve their corresponding backbones at the cross-task level, supporting the robustness of SDM across continuous-control settings.

E.3 Additional Mechanism Analyses

Post-hoc Synergy Validation. To verify that the learned SDM policy induces trajectory distributions with stronger synergistic dependency rather than merely increasing the training-time estimator output, we perform an independent post-hoc evaluation for Fig. 8(c). Specifically, we freeze the final baseline and SDM policies and use them to collect fresh trajectories under matched environment and action-sampling seeds. The collected episodes from the two policies are split into training and held-out test sets using identical episode indices. We then train a new dependency evaluator

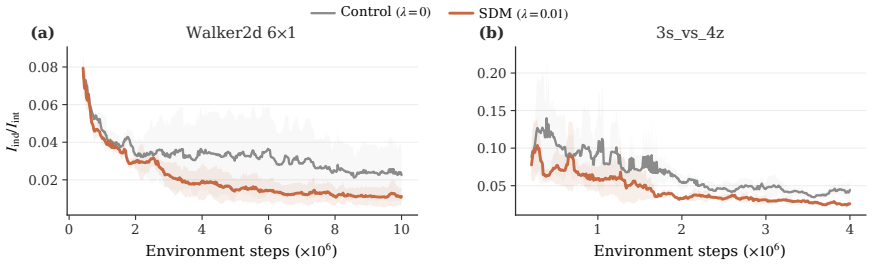

Figure 11: Additional mechanism analysis of SDM with HAPPO. We report the estimated dependency ratio $I_{\text{ind}}/I_{\text{int}}$ on (a) Walker2d 6×1 and (b) $3s_vs_4z$, comparing the control setting ($\lambda = 0$) with SDM. Lower values indicate that a smaller fraction of the interaction-conditioned dependency is attributable to individual action dependence. Warm-up samples are excluded, and curves show mean $\pm$ standard deviation across random seeds after per-seed rolling-median smoothing.

from scratch on a balanced mixture of trajectories generated by both policies, without using any estimator parameters obtained during policy training. After training, this shared evaluator is frozen and applied separately to the held-out trajectories of the baseline and SDM policies to estimate synergistic dependency and the corresponding dependency statistics. Using the same independently trained evaluator for both policies ensures that differences in the measured dependency arise from their induced trajectory distributions rather than from separately co-adapted estimators.

To further understand how SDM affects the learned coordination structure, we analyze the relative strength of individual and interaction-conditioned dependencies during training. Specifically, we report the estimated dependency ratio $I_{\text{ind}}/I_{\text{int}}$ on Walker2d 6×1 and $3s_vs_4z$, comparing the control setting ($\lambda = 0$) with SDM. A lower ratio indicates that a smaller fraction of the interaction-conditioned dependency can be explained by the agent's action alone.

As shown in Figure 11, SDM consistently yields a lower $I_{\text{ind}}/I_{\text{int}}$ ratio across both tasks. This suggests that the learned dependency structure shifts toward interaction-conditioned effects, with a smaller relative contribution from individual action dependence. These results provide additional mechanism evidence that SDM promotes interaction-dependent coordination.

E.4 Computational Cost

We evaluate the computational cost of SDM by comparing the training throughput of HAPPO and HAPPO+SDM. All experiments are conducted on NVIDIA Tesla M40 GPUs with 24 GB memory, with one task assigned to each GPU. For each task, both methods use the same hardware, environment parallelism, and training configuration. We select one representative task from each benchmark: Reference (Discrete) from MPE, Walker2d 6×1 from MAMuJoCo, $3s_vs_4z$ from SMAC, and Protoss 5v5 from SMACv2. Each method is trained for 500,000 environment steps using a single random seed. FPS is measured by HARL as the number of environment steps processed per second of elapsed wall-clock time.

As shown in Table 7, SDM reduces training throughput because of the additional optimization of the dependency estimators. Nevertheless, the resulting training cost remains manageable across all four benchmarks, particularly considering the consistent performance improvements obtained by SDM. Moreover, the additional computation is confined to training: SDM does not modify the decentralized policy architecture and therefore introduces no additional computation during execution.

Table 7: Training throughput of HAPPO and HAPPO+SDM on representative tasks. Higher FPS indicates higher training throughput.

Benchmark	Task	HAPPO	HAPPO+SDM
MPE	Reference (D)	5541	4086
MAMuJoCo	Walker2d 6×1	2584	1818
SMAC	3s_vs_4z	863	529
SMACv2	Protoss 5v5	314	233