Focusing Influence Mechanism for Multi-Agent Reinforcement Learning

Yisak Park **Sunwoo Lee** **Seungyul Han***
Graduate School of Artificial Intelligence
UNIST
Ulsan, South Korea 44919
{isaac1018,sunwoolee,syhan}@unist.ac.kr

Abstract

Cooperative multi-agent reinforcement learning (MARL) under sparse rewards remains fundamentally challenging because agents often fail to concentrate their influence, leading to insufficiently coordinated exploration. To address this, we propose the Focusing Influence Mechanism (FIM), a framework that encourages agents to focus their influence on under-explored parts of the state space through an entropy-based criterion, while leveraging eligibility traces to enable multiple agents to consistently align and sustain their influence on the same parts of the state space when beneficial, thereby promoting coordinated and persistent joint behavior. By emphasizing under-explored regions of the state space, FIM facilitates more efficient and structured exploration even under extremely sparse rewards. Across diverse MARL benchmarks, FIM consistently improves cooperative performance over strong baselines. Our implementation is available at https://unistmlic.github.io/FIM/.

1 Introduction

Cooperative multi-agent reinforcement learning (MARL) is a powerful framework for sequential decision-making problems involving multiple agents, with applications in autonomous driving [1], multi-robot coordination [2], and real-time strategy games [3]. These environments involve partial observability, making decentralized partially observable Markov decision processes (Dec-POMDPs) [4] a natural modeling choice. To address the challenges arising from limited observability, the centralized training with decentralized execution (CTDE) [5, 6] paradigm has been widely adopted. In CTDE, policies are trained using access to the global state and all agents' observations, but are executed using only local observations. Prominent CTDE methods such as VDN [7], QMIX [8], and QPLEX [9] leverage value decomposition to promote coordinated policy learning.

Despite the success of CTDE, many CTDE-based MARL methods struggle in sparse-reward settings where effective exploration is essential [10–12]. Several approaches have been proposed to address this challenge, including maximizing mutual influence between agents [11], prioritizing under-visited but important states [13], diversifying trajectories [14], and leveraging communication [15, 16]. While promising, we observe that these methods can still fail in challenging sparse-reward environments, particularly under extremely sparse rewards, because they do not explicitly guide how agents should coordinate and allocate their influence to escape local optima. Liu et al. [17] improve exploration efficiency by concentrating an agent's influence on an external state; however, selecting external-state dimensions relies on human knowledge and prior task information, which can be difficult to scale as the dimensionality of the external state increases or when such information is unavailable.

*Correspondence to: Seungyul Han

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To address these issues, we identify two factors that make sparse-reward MARL difficult. First, even when agents can influence the environment, their influence tends to remain scattered across multiple targets without converging on shared targets, leading to diluted and ineffective exploration. Second, agents lack a principled criterion for prioritizing parts of the state space that are under-explored relative to the current behavior policy. Motivated by these observations, we propose the **Focusing Influence Mechanism (FIM)**, a MARL framework that addresses both challenges through two complementary components: (i) an *agent focusing influence (AFI)* module that concentrates multiple agents' influence on shared targets by accumulating collective influence via eligibility traces and amplifying intrinsic rewards for components that receive consistent joint attention, thereby enabling persistent coordinated behavior, and (ii) a *state focusing influence (SFI)* module that uses an entropy-based criterion to prioritize under-explored parts of the state space, guiding AFI toward more meaningful targets. As a result, FIM promotes more efficient and structured exploration in challenging sparse-reward environments, and extensive experiments across diverse MARL benchmarks demonstrate improved performance over existing methods.

2 Related Works

Intrinsic Motivation for MARL. Intrinsic motivation is widely used to promote exploration in sparse-reward environments. Curiosity-driven objectives seek novel states [18, 13, 19–23], while trajectory diversity methods expand state-space coverage [24–27]. Committed exploration is induced via a shared latent variable [28, 29], and spatial formations reduce redundant exploration [30]. Subgoal-based methods decompose tasks into smaller objectives [31–34]. Exploration can also be focused in low-dimensional subspaces [12, 35, 36], and expectation alignment enables agents to adapt to anticipated peer behaviors [37].

Counterfactual Reasoning Based Credit Assignment. Counterfactual reasoning enables credit assignment by measuring each agent's contribution to the team reward. COMA estimates individual action advantages using counterfactual baselines [38–41], while predictive counterfactual models support value decomposition [42, 43]. Shapley value–based methods assign local credit by marginalizing individual contributions to the global reward [44–46]. In offline settings, counterfactual conservatism [47] and sample averaging [48] improve stability. Counterfactual reasoning also aids in identifying important agents [49] and salient states [50].

Influence-Driven Coordination. Influence-based methods aim to promote coordination by inducing causally significant changes. Social influence quantifies how an agent's actions affect the behaviors of its teammates [10, 51–54] and guides communication decisions [55]. Opponent modeling enables agents to influence policy updates of others [56–59]. Influence-aware exploration affects future dynamics [11, 60] or induces novel observations [61]. Influence has been extended to incentivize beneficial behaviors in others [62], discourage undesirable actions [63], or shape the returns of other agents [64], as well as to affect external states [17] or latent representations of the environment [65]. LAIES [17] measures the deviation of the realized transition from its counterfactual expectation, whereas FIM measures the influence on the magnitude of state change, directly capturing whether agents can enlarge the change of a target dimension. Our work builds on this line by aligning and concentrating influence across agents on key parts of the state space, enabling more persistent and structured coordination.

3 Preliminary

Decentralized POMDP and CTDE Setup. In MARL, the environment is typically modeled as a Dec-POMDP [4], defined by the tuple $\langle \mathcal{N}, S, A, P, R, O, \mathcal{O}, \gamma \rangle$, where $\mathcal{N}$ is a set of n agents, S is the global state space, $A = A^0 \times \cdots \times A^{n-1}$ is the joint action space, and γ is the discount factor. At each timestep t, each agent $i \in \mathcal{N}$ receives a local observation $o_t^i = \mathcal{O}(s_t, i)$ and chooses an action a_t^i from its policy π^i, based on its trajectory $\tau_t^i = (o_0^i, a_0^i, \ldots, o_t^i)$. The state s_t is defined as a D-dimensional vector, i.e., $s_t = (s_t^0, \cdots, s_t^{D-1})$, and for a given $(s_t, \mathbf{a}_t)$ pair, the environment transitions to $s_{t+1} \sim P(\cdot \mid s_t, \mathbf{a}_t)$ and returns a shared reward $r_t = R(s_t, \mathbf{a}_t)$. The goal is to learn a joint policy $\boldsymbol{\pi} = \prod_{i=1}^{n} \pi^i$ that maximizes the expected return $\sum_{t=0}^{\infty} \gamma^t r_t$. In this paper, we adopt the CTDE paradigm [8], where agents are trained using the global state to optimize a total value function Q^{tot}, while each agent executes actions based solely on local observations during deployment.

Credit Assignment via Counterfactual Reasoning. In the CTDE paradigm, credit assignment mechanisms [8, 47, 17] estimate each agent's contribution to team performance, supporting not only the optimization of a global value function but also promoting effective exploration [14], information sharing [30], and communication [66]. A widely adopted technique is counterfactual reasoning [38, 47, 17], which quantifies causal influence by comparing the actual outcome to a counterfactual one where only an individual agent's action is replaced. COMA [38], for example, defines credit for agent i as:

$$\text{credit}_t^i = f(s_t, \boldsymbol{\tau}_t, \mathbf{a}_t) - \mathbb{E}_{a_t^i \sim \pi^i(\cdot|o_t^i)} \left[f(s_t, \boldsymbol{\tau}_t, a_t^i, \mathbf{a}_t^{-i}) \right], \tag{1}$$

where $f = Q^{tot}$. This formulation has been leveraged not only for advantage estimation but also for shaping exploration and coordination via intrinsic rewards.

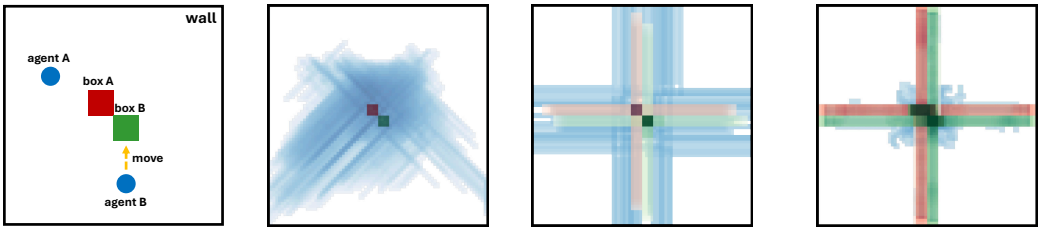
Eligibility Trace. Eligibility traces are used to implement TD(λ) online by propagating the current TD error to future timesteps for value updates [67]. At each timestep t, the trace $e_t(s)$ is updated as:

$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) + 1, & \text{if } s = s_t, \\ \gamma\lambda e_{t-1}(s), & \text{otherwise}, \end{cases} \tag{2}$$

where λ is the decay factor. This mechanism accumulates eligibility for recently visited states and decays it over time, thereby concentrating value updates on recently visited states. In this work, we adapt this concept to sustain coordinated influence, encouraging multiple agents to persistently focus on under-explored state dimensions.

4 Methodology

4.1 The Need for Advanced Focusing Influence

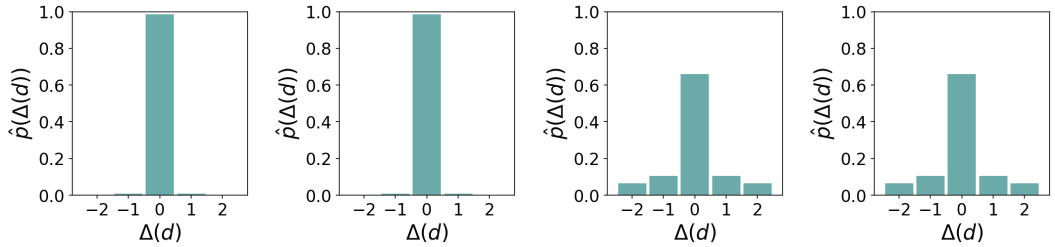


(a) Push-2-Box (b) Vanilla QMIX (c) QMIX with AFI (d) QMIX with AFI and SFI (proposed FIM)

Figure 1: Comparative results in the Push-2-Box environment: (a) shows an enlarged view of the environment, and (b–d) show average visitation counts of two agents (blue) and two boxes (Box A: red, Box B: green) over 3M timesteps across 100 seeds. Darker areas indicate more frequent visits.

In cooperative MARL, agents must often discover coordinated behaviors through exploration, particularly in tasks where progress depends on cooperative joint actions [10–12]. While each agent individually possesses the capacity to influence the environment, sparse rewards provide insufficient learning signal to coordinate their influences. As a result, agents act in an uncoordinated manner with scattered, non-cooperative influence, failing to converge on shared targets. To illustrate, we consider the Push-2-Box environment [11] in Fig. 1(a), which involves two agents and two boxes. A single agent moves a box by one cell, while simultaneous pushing moves it by two cells; agents must coordinate to reach a wall within the episode time limit. Positive rewards are provided only when a box reaches the wall, and thus success requires persistent, synchronized pushing of the same box. Under such reward sparsity, CTDE methods often fail to discover the need for coordinated pushing during training, resulting in limited variation in the box position and task failure. Fig. 1(b) illustrates this failure mode for QMIX, where scattered exploration and weak coordination prevent success.

These observations motivate the need to (i) focus multi-agent influence on shared targets, and (ii) prioritize under-explored state dimensions under the current behavior policy. To this end, we propose the **Focusing Influence Mechanism (FIM)**, which promotes cooperation through two components: *agent focusing influence (AFI)* and *state focusing influence (SFI)*. AFI encourages agents to align their influence on the same targets, for example by focusing on the same box, by accumulating collective



(a) Box A (QMIX) (b) Box B (QMIX) (c) Box A (with AFI) (d) Box B (with AFI)

Figure 2: Empirical distribution $\hat{p}$ of temporal changes $\Delta(d)$ for box positions in Push-2-Box. (a–b): Without AFI. (c–d): With AFI.

influence via eligibility traces and amplifying intrinsic rewards for dimensions associated with consistent joint attention, thereby driving coordinated behavior. As shown in Fig. 1(c), incorporating AFI into QMIX increases collective influence on shared targets; however, since AFI operates across all state dimensions, this influence can be spread too broadly, leaving insufficient effort on the dimensions that truly require coordination, such as box positions. SFI addresses this by weighting state dimensions using an entropy-based criterion, assigning higher priority to under-explored ones. Fig. 1(d) shows that combining AFI and SFI further concentrates joint influence on box positions, enabling successful task completion. Several prior methods also incorporate state influence or exploration objectives [14, 17, 30]. In Section 5, we compare these methods on Push-2-Box and show that, despite the simplicity of the task, they fail to solve it reliably under sparse rewards, whereas FIM succeeds by providing the coordination and persistence required in this setting.

4.2 Agent Focusing Influence

To enable coordinated behavior, agents must not only act but also exert *aligned influence* on shared state dimensions. We therefore quantify how much a joint action contributes to changes in each state dimension beyond what would arise without coordination. To this end, we adopt a counterfactual formulation [38] to measure the collective influence on dimension d at timestep t:

$$\mathrm{Inf}_t^d(s_t, a_t) = \sum_{i=0}^{n-1} \left\{ \left| \hat{s}_{t+1}^d(s_t, a_t) - s_t^d \right| - \mathbb{E}_{a_t^i \sim \rho} \left[\left| \hat{s}_{t+1}^d(s_t, a_t^i, a_t^{-i}) - s_t^d \right| \right] \right\}, \tag{3}$$

where $\hat{s}(\cdot)$ approximates the environment dynamics, and ρ denotes a random policy that removes the coordinated contribution of agent i. This formulation isolates the additional state change attributable to coordinated joint actions, rather than independent or uncoordinated behavior.

While Inf_t^d captures instantaneous coordination, effective cooperation requires *sustained* and temporally consistent influence. To account for temporal accumulation, we introduce an eligibility trace:

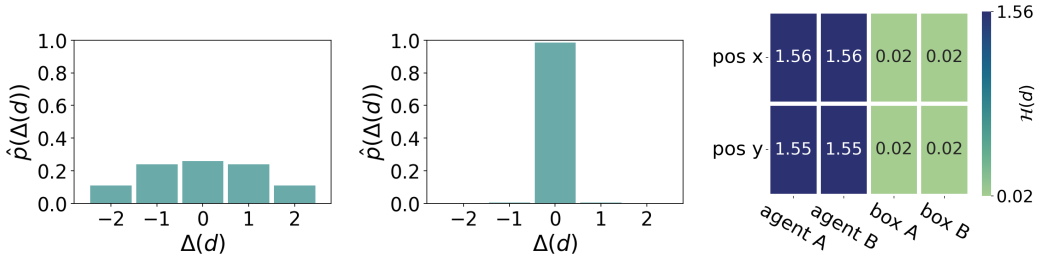
$$e_t^d = \gamma \cdot e_{t-1}^d + \mathrm{Inf}_t^d, \tag{4}$$

which aggregates past influence signals and reflects how consistently a dimension has been jointly affected over time. Using this trace, we define the **Agent Focusing Influence (AFI)** intrinsic reward:

$$r_t^{\mathrm{AFI}} = \sum_{d \in \mathcal{D}} \mathrm{Inf}_t^d \cdot \max(e_{t-1}^d, 1), \tag{5}$$

Here, $\mathcal{D} = \{0, \ldots, D-1\}$ indexes the state dimensions, and $\max(e_{t-1}^d, 1)$ keeps the trace multiplier at least one, while reinforcing dimensions that have received consistent collective attention. By weighting the current influence Inf_t^d with the accumulated trace e_{t-1}^d, AFI amplifies dimensions where coordinated influence is repeatedly exerted, encouraging agents to sustain coherent joint behavior rather than transient and unstructured interactions.

To illustrate the effect of AFI, Fig. 2 shows the empirical distribution of temporal changes in box positions in Push-2-Box, with and without AFI. Without AFI, panels (a) and (b) show that both boxes exhibit similarly low variation, reflecting the dispersed influence of vanilla QMIX.



(a) Agent positions (b) Box positions (c) Entropy $\mathcal{H}(d)$

Figure 3: (a–b) Empirical distribution $\hat{p}$ of temporal changes $\Delta(d)$ for agent and box positions. (c) Entropy $\mathcal{H}(d)$ for each state dimension in Push-2-Box.

With AFI, panels (c) and (d) show increased variation in both box positions, indicating stronger collective influence. However, since AFI aggregates influence across all state dimensions, this effect is distributed broadly, including agent positions that do not require coordination. As a result, insufficient influence is concentrated on box positions, limiting effective task completion. This limitation motivates a mechanism that selectively prioritizes which dimensions to focus on, which we address in Section 4.3.

4.3 State Focusing Influence and FIM Framework

To concentrate the coordinated influence induced by AFI on dimensions that matter for exploration, we introduce **State Focusing Influence (SFI)**, which identifies and prioritizes state dimensions that are under-explored under the current joint behavior policy $\boldsymbol{\beta}$. Our key observation is that under-explored dimensions exhibit low diversity in their induced transitions, which can be characterized by low conditional entropy $\mathbb{E}_{\boldsymbol{\beta}}\big[\mathcal{H}(s^d_{t+1} \mid s_t)\big]$. Intuitively, if a dimension rarely changes under the current policy, it receives little effective exploration, even over long horizons.

Proposition 4.1. *Let $\mathcal{E}_d := \sup_{t\geq 0} \mathcal{H}(s^d_{t+1} \mid s^d_t)$ be positive. To achieve a target exploration level H^d_{target} such that $\mathcal{H}(s^d_0, \ldots, s^d_T) \geq H^d_{\text{target}}$, the required horizon satisfies*

$$T \geq \frac{H^d_{\text{target}} - \mathcal{H}(s^d_0)}{\mathcal{E}_d}. \tag{6}$$

Proof. Proof of Proposition 4.1 is provided in Appendix A.

This result shows that when $\mathcal{H}(s^d_{t+1} \mid s^d_t)$ is small, each transition contributes little to exploration, requiring disproportionately long horizons. Therefore, such dimensions must be explicitly targeted to achieve sufficient coverage. However, directly comparing entropy across dimensions is inappropriate due to scale differences, since those with larger natural variation trivially yield higher entropy. To address this issue, we normalize each dimension by its expected change magnitude under $\boldsymbol{\beta}$:

$$\tilde{s}^d_t := \frac{s^d_t}{\mathbb{E}_{\boldsymbol{\beta}}[|s^d_{t+1} - s^d_t|] + \epsilon}, \tag{7}$$

which removes scale effects and enables fair comparison across dimensions, where ϵ is a small constant added for numerical stability.

We then approximate the conditional entropy using the marginal entropy of normalized changes:

$$\mathcal{H}(d) = \mathbb{E}_{\boldsymbol{\beta}}\big[-\log p\big(\Delta^d(s_t, s_{t+1})\big)\big], \tag{8}$$

where $\Delta^d := \tilde{s}^d_{t+1} - \tilde{s}^d_t$ and p is estimated empirically. This approximation captures how diverse the observed transitions are along each dimension under the current policy.

Based on this, SFI assigns higher importance to dimensions with lower entropy, i.e., those that are insufficiently explored. Integrating SFI with AFI yields the **Focusing Influence Mechanism (FIM)**:

$$r^{\text{FIM}}_t = \sum_{d\in\mathcal{D}} w_d \cdot \text{Inf}^d_t \cdot \max(e^d_{t-1}, 1), \tag{9}$$

where $w_d = \mathrm{Softmax}(-\mathcal{H}(d))$ emphasizes under-explored dimensions. To ensure stable adaptation to the evolving behavior policy, the weights w_d are updated using an exponential moving average:

$$w_d \leftarrow (1-\phi)\, w_d + \phi\, w_d^{\text{new}}, \tag{10}$$

where ϕ is the update rate. Crucially, FIM does not explicitly maximize entropy. Instead, it promotes exploration by allocating coordinated influence to under-explored dimensions, increasing their variability through joint behavior rather than stochasticity alone.

To visualize the effect of SFI, Fig. 3 shows the empirical distribution of temporal changes for agent and box positions, along with the corresponding entropy $\mathcal{H}(d)$ for each state dimension in Push-2-Box. Agent positions exhibit high entropy due to frequent changes, whereas box positions have low entropy because they require coordinated actions. Consequently, SFI assigns higher weight to box-position dimensions, directing AFI's influence toward targets that require joint effort.

Finally, the overall reward is defined as

$$r_t = r_{\text{ext},t} + \alpha \cdot r_t^{\text{FIM}},$$

where α is the intrinsic reward scale. We use QMIX [8] as the backbone, although FIM is agnostic to the underlying CTDE algorithm. The complete training procedure is summarized in Algorithm 1, with algorithmic details in Appendix B.1 and entropy-estimation settings in Appendix B.2.

5 Experiment

In this section, we evaluate the effectiveness of FIM. We begin with the Push-2-Box task introduced in Section 4.1, comparing various combinations of FIM components. We then extend the evaluation to more complex benchmarks, including the StarCraft Multi-Agent Challenge (SMAC) [68], SMACv2 [69], and Google Research Football (GRF) [70]. In all performance plots, the mean across 5 random seeds is shown as a solid line, and the standard deviation is represented by a shaded area.

5.1 Performance Comparison on Push-2-Box

We revisit the Push-2-Box task introduced in Fig. 1(a). In Push-2-Box, an external reward of $+100$ is given when either box reaches the wall, and -1 is applied if the task fails. The task is solved when agents push a box to the wall within the limited episode length, and detailed settings are provided in Appendix C. For comparison, Fig. 4 reports success rates for several MARL baselines: **LAIES** [17], which encourages influence over heuristic external state features; **CDS** [14], which promotes trajectory diversity; **FoX** [30], which leverages formation-aware exploration; and **vanilla QMIX**, trained using only extrinsic rewards. We also include component-wise variants: **QMIX+AFI**, which applies r_t^{AFI} with uniform weights across all state dimensions, and **QMIX+SFI**, which applies $\sum_{d\in\mathcal{D}} w_d \cdot \mathrm{Inf}_t^d$ without eligibility traces.

We observe that, despite the simplicity of the environment, only FIM solves the task. All other MARL baselines, including vanilla QMIX and exploration-oriented methods, fail on Push-2-Box. Notably, LAIES also fails even though it targets the same state dimensions in Push-2-Box, suggesting that identifying the correct dimensions is insufficient without AFI. Among our ablations, QMIX+AFI alone fails because concentrated influence without targeted dimension weighting is spread across all state dimensions, leaving insufficient joint effort on box positions to achieve the task, as shown in Fig. 1(c). QMIX+SFI increases interaction with hard-to-change components that require joint effort, but still fails due to inconsistent focus on a single target. Together, these results show that both SFI and AFI are necessary, and that their combination enables effective influence focusing.

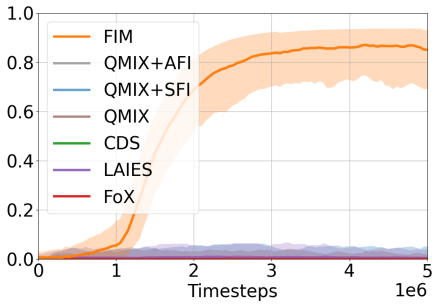


Figure 4: Performance comparison on the Push-2-Box environment.

5.2 Performance Comparison on Diverse Multi-Agent Environments

Next, we evaluate our method on three standard MARL benchmarks: SMAC, SMACv2, and GRF. SMAC is a multi-agent combat environment built on StarCraft II, where agents must coordinate to

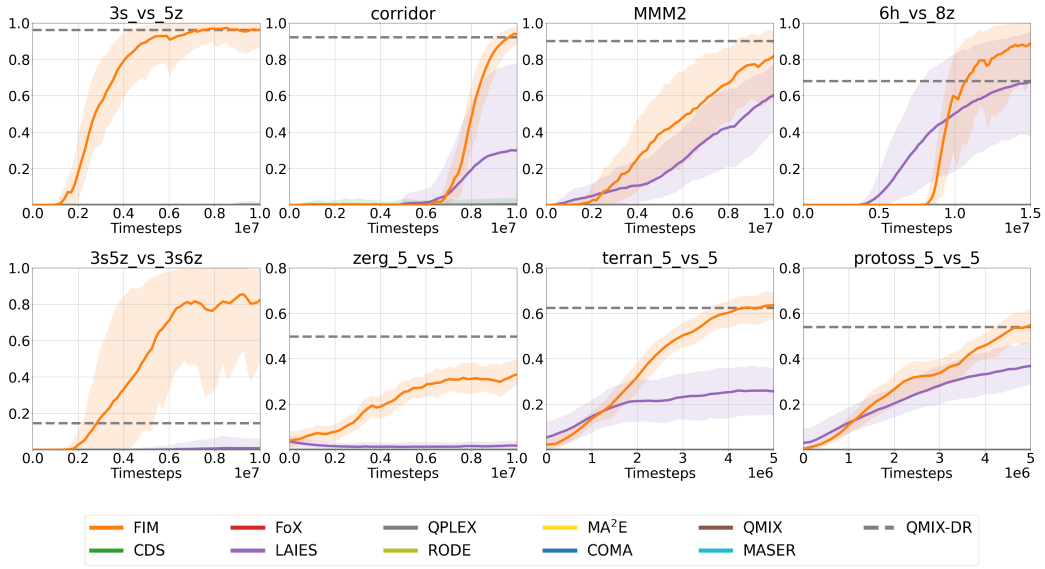


Figure 5: Performance comparison on SMAC and SMACv2 environments

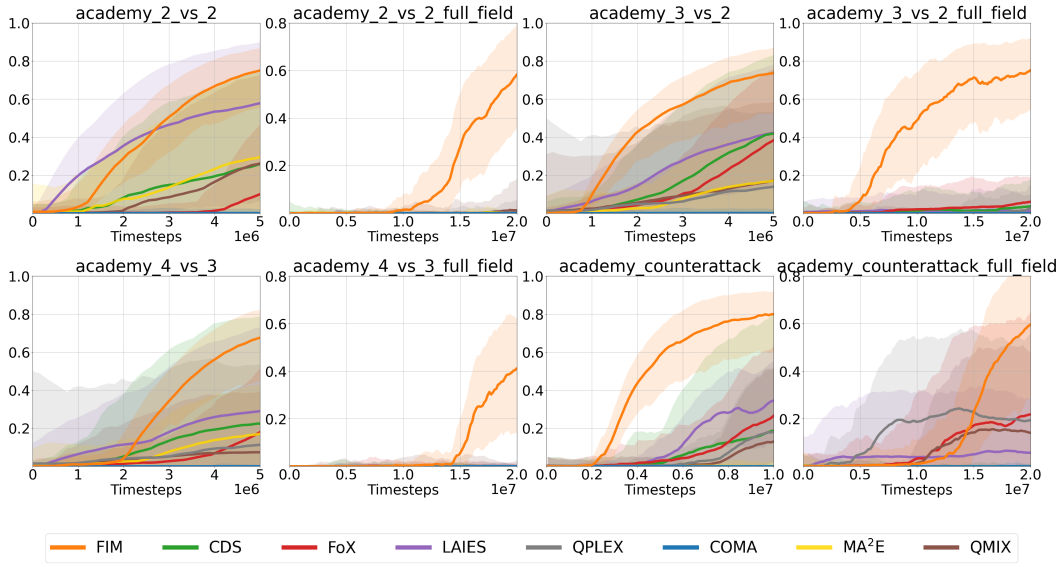


Figure 6: Performance comparison on GRF environments

defeat enemy units. We use a truly sparse reward setting in which agents receive +1 for a win, 0 for a draw, and -1 for a loss. Evaluation is conducted on five challenging scenarios: one hard map (`3s_vs_5z`) and four super-hard maps (`corridor`, `MMM2`, `6h_vs_8z`, `3s5z_vs_3s6z`), where `s`, `z`, and `h` refer to stalker, zealot, and hydralisk units, respectively. We further evaluate on SMACv2, which introduces randomized initial configurations and diverse unit types, using three scenarios (`zerg_5_vs_5`, `terran_5_vs_5`, `protoss_5_vs_5`) under the same sparse reward setting. GRF is a multi-agent soccer environment where teams compete to score goals under sparse rewards: +100 for a win and -1 for a loss. We evaluate on eight scenarios, including four half-field settings (`academy_2_vs_2`, `academy_3_vs_2`, `academy_4_vs_3`, `academy_counterattack`) and their corresponding full-field versions, which are more challenging due to the increased field size. Further environment details and visualizations are provided in Appendix C.

For SMAC and SMACv2, in addition to **Vanilla QMIX**, **LAIES**, **CDS**, and **FoX**, we include baselines compared with FIM variants: **COMA** [38], which uses counterfactual baselines for credit assignment; **MASER** [32], which identifies subgoals based on Q-values; **RODE** [71], which assigns roles to agents; **MA^{2}E** [72], which applies masking to reconstruct global information; and **QPLEX** [9],

which adopts a dueling architecture. For GRF, we compare against Vanilla QMIX, LAIES, CDS, FoX, COMA, MA^{2}E, and QPLEX, omitting methods without publicly available GRF results. We also include QMIX-DR for SMAC, trained with dense rewards, as a dense-reward reference rather than a strict upper bound. Its damage-based reward can bias learning toward distributed attacks, whereas the original sparse reward allows FIM to discover the focused-fire strategy that is particularly effective in `6h_vs_8z`. All baselines are evaluated using author-provided implementations, while our method uses the best hyperparameters identified via ablations. Further details, including algorithm descriptions and hyperparameter configurations, are provided in Appendix D.1 and Appendix D.2, respectively. Here, SFI in FIM typically assigns higher weights to opponent state dimensions that are difficult to change, whereas LAIES heuristically targets all enemy-related dimensions, highlighting a clear difference from our approach.

Fig. 5 and Fig. 6 present success rate comparisons on the SMAC, SMACv2, and GRF benchmarks. Across these environments, FIM consistently achieves the highest success rates among all baselines. In SMAC and SMACv2, sparse rewards pose a significant challenge because agents must eliminate all enemies without intermediate feedback. While LAIES remains competitive in some scenarios, it struggles on complex maps such as `3s_vs_5z` and `corridor`, since it treats all enemies as external state dimensions without focusing influence. In contrast, FIM remains robust by weighting under-explored dimensions. In GRF, although some baselines perform well on simpler half-field tasks, they largely fail on full-field maps with rare scoring opportunities. By concentrating influence on hard-to-change elements, FIM maintains strong performance across all scenarios. Overall, these results show that FIM promotes effective cooperation, enabling agents to solve challenging tasks even under highly sparse rewards.

To evaluate FIM beyond structured vector states, we consider Overcooked [73], whose spatial binary tensor encodes agents and ingredients, allowing FIM to identify under-explored dimensions such as the soup location. We also evaluate continuous actions in the Multi-Agent Particle Environment (MPE) [74, 75] with a MAPPO backbone. Both evaluations are detailed in Appendix E.1.

5.3 In-depth Analysis and Ablation Studies

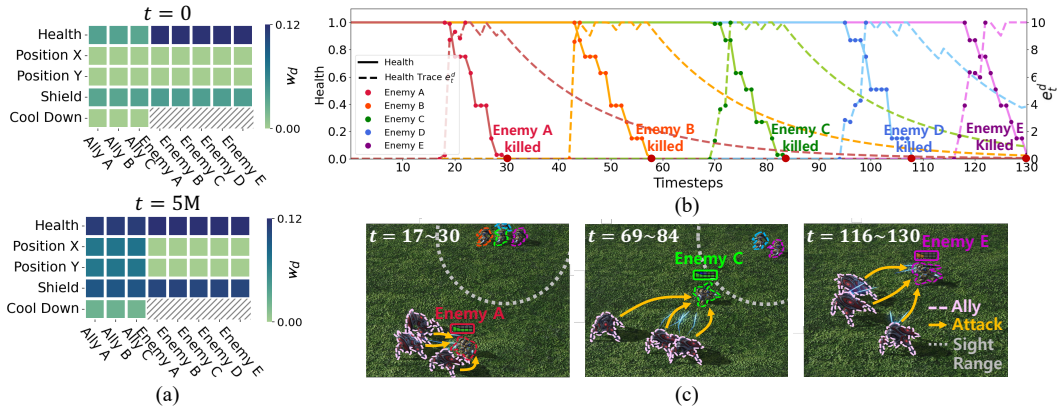
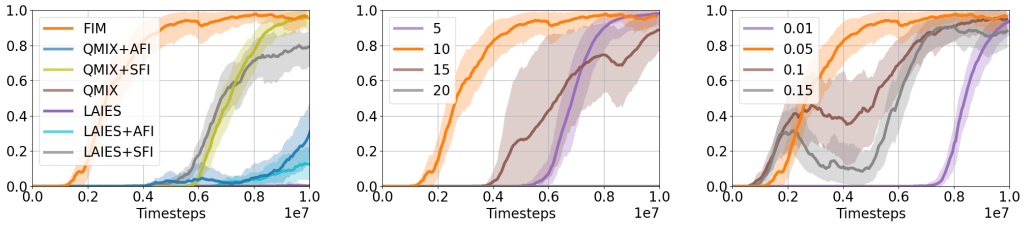


Figure 7: Behavior analysis in `3s_vs_5z`: (a) SFI weights w_d at 0 and 5M steps; larger weights prioritize low-entropy dimensions. (b) Changes in enemy health and its trace e_t^d. (c) Rendered frames for highlighting agents' coordination.

To better understand FIM's focusing mechanisms, we conduct detailed analyses and ablations in the environments where it shows the largest gains: SMAC `3s_vs_5z` and GRF `academy_3_vs_2_full_field`. In SMAC `3s_vs_5z`, SFI highlights enemy features such as health as under-explored state dimensions, as they remain relatively stable without coordination and thus become targets for joint influence. As shown in Appendix E.2, similar state dimensions exhibit low entropy in other SMAC scenarios. In GRF, the keeper's position is frequently up-weighted by SFI, since it is difficult for agents to manipulate. Overall, these results show that FIM focuses influence on meaningful state dimensions, improving performance by prioritizing under-explored features such as health in SMAC and the keeper's position in GRF.

Behavior Analysis: To further illustrate the effect of FIM, Fig. 7(a) reports the dimensional weights w_d across training iterations ($t = 0$ and 5M) in `3s_vs_5z`. Throughout training, the health of enemy units consistently receives high weights from SFI, since these dimensions change substantially only when agents coordinate their attacks to focus fire on the same target. Notably, as training progresses, ally health dimensions are also gradually up-weighted, since agents tend to avoid battle to preserve their own health. By assigning higher weight to ally health, SFI encourages agents to engage more actively with enemies. Fig. 7(b) shows how the corresponding eligibility traces evolve over the enemy-health dimensions, while Fig. 7(c) highlights key timesteps at which enemy units are eliminated within the trajectory. Agents trained with FIM learn to pull enemies into sight and focus fire sequentially: around $t \approx 20$, they concentrate on the first red enemy, increase its health trace, and eliminate it by $t \approx 30$; once removed, its influence drops to zero and attention shifts to the next enemy (e.g., orange at $t \approx 40$), repeating the same process. This strategy resembles human gameplay in StarCraft II. We provide a complementary analysis for GRF in Appendix F.1, showing that agents learn to disrupt the keeper's behavior, thereby increasing goal-scoring opportunities. Together, these results demonstrate that FIM promotes structured and effective cooperation even in sparse-reward environments.



(a) Component evaluation (b) Reward scale α (c) Update rate ϕ

Figure 8: Ablation studies on SMAC `3s_vs_5z`

Component Ablations: We evaluate the contribution of each component on `3s_vs_5z`. Fig. 8(a) compares **Vanilla QMIX**, **QMIX+AFI**, **QMIX+SFI**, **LAIES**, **LAIES+AFI**, which rescales LAIES's influence using per-dimension eligibility traces, **LAIES+SFI**, which applies w_d as a per-dimension weight on LAIES's influence signal, and **FIM**. The results show that Vanilla QMIX fails to solve the task, while SFI and AFI each provide consistent improvements regardless of the backbone: both QMIX+AFI and LAIES+AFI benefit from enforcing consistent focus on shared targets via eligibility traces, and both QMIX+SFI and LAIES+SFI benefit from concentrating influence on a small number of under-explored dimensions. Combining SFI and AFI yields faster convergence and higher final success rates, confirming that the two components play essential and synergistic roles in influence-based exploration. An ablation of the influence definition is provided in Appendix F.2.

Hyperparameter Analysis: Panels (b)-(c) of Fig. 8 examine the effects of the intrinsic reward scale α and dimension-weight update rate ϕ. For α, too-small values provide insufficient incentive for agents to concentrate their influence, while overly large values cause agents to over-prioritize intrinsic rewards over extrinsic feedback, leading to unstable training. In practice, α is environment-dependent, but can be effectively chosen by scaling it as large as possible without dominating the extrinsic reward signal. For ϕ, too-small values cause the dimension weights w_d to adapt too slowly to the evolving behavior policy, whereas too-large values result in frequent weight shifts that destabilize the focus target and prevent sustained coordinated influence. Although larger ϕ leads to faster adaptation, the weights eventually converge over time, suggesting that ϕ primarily controls the adaptation timescale rather than the final behavior. We use $\alpha = 10$ and $\phi = 0.05$ as default hyperparameters.

Additional analyses, including comparisons with alternative dimension-weighting strategies and ablations on more scenarios, are provided in Appendix F.2. The learned dynamics model is further analyzed in Appendix F.3, where we show that its evolving prediction error correlates with exploratory effectiveness and overall performance. Additionally, Appendix F.4 reports computational complexity, showing that FIM achieves superior performance with training time comparable to QMIX.

Table 1: Final win rates (mean ± std) on SMAC `3s5z_vs_3s6z`.

QMIX	QMIX + RND	QMIX + ICM	QMIX + Count	FIM (count)	FIM (frozen)	**FIM**
0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.04 ± 0.18	0.71 ± 0.27	$\mathbf{0.82 \pm 0.24}$

Table 2: Scalability on SMAC: final win rates and FIM training steps to convergence.

Scenario	Agents	LAIES	**FIM**	FIM steps to converge
`3s5z_vs_3s6z`	8	0.04 ± 0.05	$\mathbf{0.82 \pm 0.24}$	$\approx$ 7M
`3s6z_vs_3s7z`	9	0.00 ± 0.00	$\mathbf{0.80 \pm 0.25}$	$\approx$ 11M
`3s7z_vs_3s8z`	10	0.00 ± 0.00	$\mathbf{0.79 \pm 0.27}$	$\approx$ 25M

Entropy Criterion and Curiosity Controls. Low transition entropy can reflect strong control, but also a limited range of observed transitions; broader coverage can help escape local optima. We isolate focused influence from curiosity using FIM with a pre-trained, frozen dynamics model and QMIX + RND [76], ICM [77], or Count [78], with intrinsic rewards scaled to avoid dominating extrinsic feedback. FIM(count) replaces entropy-based weights with a count-based metric. On `3s5z_vs_3s6z`, updating the dynamics model improves on frozen dynamics, the curiosity baselines fail, and FIM(count) performs substantially worse (Table 1). These results support focused influence and entropy-based selection. Entropy-surrogate comparisons and estimation sensitivity are analyzed in Appendices F.5 and F.6.

Scalability. As teams grow, FIM must evaluate more agent-specific counterfactuals and prioritize influence over a larger state representation. We therefore test whether its focusing mechanisms remain effective as the coordination problem expands. Starting from the super-hard SMAC map `3s5z_vs_3s6z`, we increase the allied team from 8 to 10 agents and the opposing team from 9 to 11 units. Table 2 shows that FIM maintains final win rates of 0.82, 0.80, and 0.79, whereas convergence requires approximately 7M, 11M, and 25M steps. Thus, FIM preserves final performance in these larger scenarios, while the growing interaction and state spaces increase the training required.

6 Limitation and Future Work

Although FIM achieves strong performance, it still has some limitations. FIM incurs additional computational cost due to training the dynamics model, as analyzed in Appendix F.4. However, our analysis shows that this overhead is moderate relative to the performance gains, and it could be further reduced by using more lightweight model approximations. Second, although we validate FIM with spatial binary tensor states through Overcooked experiments, as shown in Appendix E.1, it is not yet specifically optimized for image-based environments. In such settings, learning to identify and focus on semantically meaningful regions or objects in the observation space could further improve performance. Deeper integration with image-based representations, for example through object-centric or detection-based mechanisms, remains an important direction for future work.

7 Conclusion

In this paper, we address the challenge of efficient cooperation in sparse-reward MARL by proposing FIM, a framework that guides agent influence toward under-explored state dimensions and sustains coordinated focus through eligibility traces. By integrating entropy-based dimension weighting with structured intrinsic rewards based on counterfactual reasoning, FIM enables agents to induce targeted and persistent state transitions. Empirical results across Push-2-Box, SMAC, SMACv2, and GRF demonstrate that FIM significantly improves learning efficiency and coordination, outperforming state-of-the-art baselines. These findings highlight the potential of influence-guided learning to enable robust multi-agent cooperation in sparsely rewarded environments.

Acknowledgments and Disclosure of Funding

This work was supported partly by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2020-II201336, Artificial Intelligence Graduate School Support (UNIST)), (No. RS-2025-25442824, AI Star-Fellowship Program (UNIST)), and (No. RS-2022-II220469, Development of Core Technologies for Task-oriented Reinforcement Learning for Commercialization of Autonomous Drones), and partly by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-23523191, LLM-Based Multi-Agent Reinforcement Learning for End-to-End Large Autonomous Swarm Control).

References

[1] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

[2] Adolfo Perrusquía, Wen Yu, and Xiaoou Li. Multi-agent reinforcement learning for redundant robot control in task-space. *International Journal of Machine Learning and Cybernetics*, 12: 231–241, 2021.

[3] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature*, 575(7782):350–354, 2019.

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

[5] Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

[6] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and YI WU. The surprising effectiveness of ppo in cooperative multi-agent games. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24611–24624. Curran Associates, Inc., 2022.

[7] 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*, pages 2085–2087, 2018.

[8] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4295–4304. PMLR, 2018.

[9] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2021.

[10] 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*, pages 3040–3049. PMLR, 2019.

[11] Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. In *International Conference on Learning Representations*, 2020.

[12] Iou-Jen Liu, Unnat Jain, Raymond A Yeh, and Alexander Schwing. Cooperative exploration for multi-agent deep reinforcement learning. In *International conference on machine learning*, pages 6826–6836. PMLR, 2021.

[13] Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34:3757–3769, 2021.

[14] 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.

[15] Sangjun Bae, Yisak Park, Sanghyeon Lee, and Seungyul Han. Llm-guided communication for cooperative multi-agent reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026.

[16] Yechan Hwang, Sangjun Bae, Jeongmo Kim, Sangwoo Bang, and Seungyul Han. MARBO: Relational belief grounding for LLM agents in social deduction games. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, 2026. URL `https://arxiv.org/abs/2609.06563`.

[17] 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*, pages 21937–21950. PMLR, 2023.

[18] Shariq Iqbal and Fei Sha. Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning. *arXiv preprint arXiv:1905.12127*, 2019.

[19] Jiahui Li, Kun Kuang, Baoxiang Wang, Xingchen Li, Fei Wu, Jun Xiao, and Long Chen. Two heads are better than one: A simple exploration framework for efficient multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:20038–20053, 2023.

[20] Shaowei Zhang, Jiahan Cao, Lei Yuan, Yang Yu, and De-Chuan Zhan. Self-motivated multi-agent exploration. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 476–484, 2023.

[21] Kai Yang, Zhirui Fang, Xiu Li, and Jian Tao. Cmbe: Curiosity-driven model-based exploration for multi-agent reinforcement learning in sparse reward settings. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

[22] Pei Xu, Junge Zhang, and Kaiqi Huang. Population-based diverse exploration for sparse-reward multi-agent tasks. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 283–291, 2024.

[23] Jian Tao, Yangkun Chen, Yang Zhang, Kai Yang, and Xiu Li. Multi-agent exploration with sub-state entropy estimation. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2024.

[24] Yucong Zhang and Chao Yu. Expode: Exploiting policy discrepancy for efficient exploration in multi-agent reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 58–66, 2023.

[25] Tianxu Li and Kun Zhu. Toward efficient multi-agent exploration with trajectory entropy maximization. In *The Thirteenth International Conference on Learning Representations*, 2025.

[26] Tianxu Li and Kun Zhu. Learning joint behaviors with large variations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23249–23257, 2025.

[27] Mingu Kang, Sunwoo Lee, Yonghyeon Jo, and Seungyul Han. Shaping zero-shot coordination via state blocking. *arXiv preprint arXiv:2605.11688*, 2026.

[28] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *Advances in neural information processing systems*, 32, 2019.

[29] Yonghyeon Jo, Sunwoo Lee, and Seungyul Han. Retaining suboptimal actions to follow shifting optima in multi-agent reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

[30] 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, pages 12985–12994, 2024.

[31] Hongyao Tang, Jianye Hao, Tangjie Lv, Yingfeng Chen, Zongzhang Zhang, Hangtian Jia, Chunxu Ren, Yan Zheng, Zhaopeng Meng, Changjie Fan, et al. Hierarchical deep multiagent reinforcement learning with temporal abstraction. *arXiv preprint arXiv:1809.09332*, 2018.

[32] Jeewon Jeon, Woojun Kim, Whiyoung Jung, and Youngchul Sung. Maser: Multi-agent reinforcement learning with subgoals generated from experience replay buffer. In *International conference on machine learning*, pages 10041–10052. PMLR, 2022.

[33] Jaebak Hwang, Sanghyeon Lee, Jeongmo Kim, and Seungyul Han. Strict subgoal execution: Reliable long-horizon planning in hierarchical reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=cMpOvMuyYa`.

[34] Junhyeon Bae, Jaebak Hwang, Gwanwoo Choi, Minung Kim, and Seungyul Han. Escaping path mirages in offline goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, 2026.

[35] Pei Xu, Junge Zhang, Qiyue Yin, Chao Yu, Yaodong Yang, and Kaiqi Huang. Subspace-aware exploration for sparse-reward multi-agent tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11717–11725, 2023.

[36] Xin He, Hongwei Ge, Yaqing Hou, and Jincheng Yu. Saeir: sequentially accumulated entropy intrinsic reward for cooperative multi-agent reinforcement learning with sparse reward. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 4107–4115, 2024.

[37] Zixian Ma, Rose Wang, Fei-Fei Li, Michael Bernstein, and Ranjay Krishna. Elign: Expectation alignment as a multi-agent intrinsic reward. *Advances in Neural Information Processing Systems*, 35:8304–8317, 2022.

[38] 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.

[39] Andrew Cohen, Ervin Teng, Vincent-Pierre Berges, Ruo-Ping Dong, Hunter Henry, Marwan Mattar, Alexander Zook, and Sujoy Ganguly. On the use and misuse of absorbing states in multi-agent reinforcement learning. *arXiv preprint arXiv:2111.05992*, 2021.

[40] Jianhao Wang, Zhizhou Ren, Beining Han, Jianing Ye, and Chongjie Zhang. Towards understanding cooperative multi-agent q-learning with value factorization. *Advances in Neural Information Processing Systems*, 34:29142–29155, 2021.

[41] Heiko Hoppe, Tobias Enders, Quentin Cappart, and Maximilian Schiffer. Global rewards in multi-agent deep reinforcement learning for autonomous mobility on demand systems. In *6th Annual Learning for Dynamics & Control Conference*, pages 260–272. PMLR, 2024.

[42] Hanhan Zhou, Tian Lan, and Vaneet Aggarwal. Pac: Assisted value factorization with counterfactual predictions in multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 35:15757–15769, 2022.

[43] Jiajun Chai, Yuqian Fu, Dongbin Zhao, and Yuanheng Zhu. Aligning credit for multi-agent cooperation via model-based counterfactual imagination. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 281–289, 2024.

[44] Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley q-value: A local reward approach to solve global reward games. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7285–7292, 2020.

[45] 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*, pages 934–942, 2021.

[46] Jianhong Wang, Yuan Zhang, Yunjie Gu, and Tae-Kyun Kim. Shaq: Incorporating shapley value theory into multi-agent q-learning. *Advances in Neural Information Processing Systems*, 35:5941–5954, 2022.

[47] Jianzhun Shao, Yun Qu, Chen Chen, Hongchang Zhang, and Xiangyang Ji. Counterfactual conservative q learning for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:77290–77312, 2023.

[48] Jinming Ma and Feng Wu. Learning to coordinate from offline datasets with uncoordinated behavior policies. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 1258–1266, 2023.

[49] Jianming Chen, Yawen Wang, Junjie Wang, Xiaofei Xie, Jun Hu, Qing Wang, and Fanjiang Xu. Understanding individual agent importance in multi-agent system via counterfactual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 15785–15794, 2025.

[50] Zelei Cheng, Xian Wu, Jiahao Yu, Wenhai Sun, Wenbo Guo, and Xinyu Xing. Statemask: Explaining deep reinforcement learning through state mask. *Advances in Neural Information Processing Systems*, 36:62457–62487, 2023.

[51] Pengyi Li, Hongyao Tang, Tianpei Yang, Xiaotian Hao, Tong Sang, Yan Zheng, Jianye Hao, Matthew E Taylor, Wenyuan Tao, and Zhen Wang. Pmic: Improving multi-agent reinforcement learning with progressive mutual information collaboration. In *International Conference on Machine Learning*, pages 12979–12997. PMLR, 2022.

[52] 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*, 2025.

[53] Sunwoo Lee, Jaebak Hwang, Yonghyeon Jo, and Seungyul Han. Wolfpack adversarial attack for robust multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 33025–33056. PMLR, 2025.

[54] Sunwoo Lee, Mingu Kang, Yonghyeon Jo, and Seungyul Han. Interaction-breaking adversarial learning framework for robust multi-agent reinforcement learning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 63907–63939. PMLR, 2026. URL `https://proceedings.mlr.press/v306/lee26s.html`.

[55] Ziluo Ding, Tiejun Huang, and Zongqing Lu. Learning individually inferred communication for multi-agent cooperation. *Advances in neural information processing systems*, 33:22069–22079, 2020.

[56] Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 122–130, 2018.

[57] Alistair Letcher, Jakob Foerster, David Balduzzi, Tim Rocktäschel, and Shimon Whiteson. Stable opponent shaping in differentiable games. In *International Conference on Learning Representations*, 2019.

[58] Annie Xie, Dylan Losey, Ryan Tolsma, Chelsea Finn, and Dorsa Sadigh. Learning latent representations to influence multi-agent interaction. In *Conference on robot learning*, pages 575–588. PMLR, 2021.

[59] Dong-Ki Kim, Matthew Riemer, Miao Liu, Jakob Foerster, Michael Everett, Chuangchuang Sun, Gerald Tesauro, and Jonathan P How. Influencing long-term behavior in multiagent reinforcement learning. *Advances in Neural Information Processing Systems*, 35:18808–18821, 2022.

[60] Zeyang Liu, Lipeng Wan, Xinrui Yang, Zhuoran Chen, Xingyu Chen, and Xuguang Lan. Imagine, initialize, and explore: An effective exploration method in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17487–17495, 2024.

[61] Haobin Jiang, Ziluo Ding, and Zongqing Lu. Settling decentralized multi-agent coordinated exploration by novelty sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17444–17452, 2024.

[62] Jiachen Yang, Ang Li, Mehrdad Farajtabar, Peter Sunehag, Edward Hughes, and Hongyuan Zha. Learning to incentivize other learning agents. *Advances in Neural Information Processing Systems*, 33:15208–15219, 2020.

[63] Kyrill Schmid, Lenz Belzner, and Claudia Linnhoff-Popien. Learning to penalize other learning agents. In *Artificial Life Conference Proceedings 33*, volume 2021, page 59. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , 2021.

[64] John L Zhou, Weizhe Hong, and Jonathan Kao. Reciprocal reward influence encourages cooperation from self-interested agents. *Advances in Neural Information Processing Systems*, 37:59491–59512, 2024.

[65] Xinran Li, Zifan Liu, Shibo Chen, and Jun Zhang. Individual contributions as intrinsic exploration scaffolds for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 28387–28402. PMLR, 2024.

[66] Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. Learning nearly decomposable value functions via communication minimization. In *International Conference on Learning Representations*, 2020.

[67] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018. URL `http://incompleteideas.net/book/the-book-2nd.html`.

[68] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

[69] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Foerster, and Shimon Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:37567–37593, 2023.

[70] 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, pages 4501–4510, 2020.

[71] Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. Rode: Learning roles to decompose multi-agent tasks. In *International Conference on Learning Representations*, 2021.

[72] Sehyeok Kang, Yongsik Lee, Gahee Kim, Song Chong, and Se-Young Yun. MA^{2}E: Addressing partial observability in multi-agent reinforcement learning with masked auto-encoder. In *The Thirteenth International Conference on Learning Representations*, 2025.

[73] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32, 2019.

[74] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.neurips.cc/paper_files/paper/2017/hash/68a9750337a418a86fe06c1991a1d64c-Abstract.html.

[75] J Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043, 2021.

[76] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1lJJnR5Ym.

[77] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[78] Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/hash/afda332245e2af431fb7b672a68b659d-Abstract.html.

[79] Jian Hu, Siyang Jiang, Seth Austin Harding, Haibin Wu, and Shih wei Liao. Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2102.03479*, 2021.

[80] Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Garðar Ingvarsson, Timon Willi, Ravi Hammond, Akbir Khan, Christian Schroeder de Witt, Alexandra Souly, Saptarashmi Bandyopadhyay, Mikayel Samvelyan, Minqi Jiang, Robert Tjarko Lange, Shimon Whiteson, Bruno Lacerda, Nick Hawes, Tim Rocktäschel, Chris Lu, and Jakob Nicolaus Foerster. Jaxmarl: Multi-agent rl environments and algorithms in jax. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

A Proof of Proposition 4.1

Proof. By the chain rule of entropy:

$$\mathcal{H}(s_0^d, s_1^d, \ldots, s_T^d) = \mathcal{H}(s_0^d) + \sum_{t=0}^{T-1} \mathcal{H}(s_{t+1}^d \mid s_0^d, \ldots, s_t^d) \tag{11}$$

$$\leq \mathcal{H}(s_0^d) + \sum_{t=0}^{T-1} \mathcal{H}(s_{t+1}^d \mid s_t^d) \tag{12}$$

$$\leq \mathcal{H}(s_0^d) + T \cdot \mathcal{E}_d, \tag{13}$$

where the first inequality follows from the fact that conditioning reduces entropy, and the second from the definition of $\mathcal{E}_d := \sup_{t \geq 0} \mathcal{H}(s_{t+1}^d \mid s_t^d)$. Therefore, to achieve $\mathcal{H}(s_0^d, s_1^d, \ldots, s_T^d) \geq H_{\text{target}}^d$, we need:

$$\mathcal{H}(s_0^d) + T \cdot \mathcal{E}_d \geq H_{\text{target}}^d, \tag{14}$$

which gives:

$$T \geq \frac{H_{\text{target}}^d - \mathcal{H}(s_0^d)}{\mathcal{E}_d}. \tag{15}$$

□

B Implementation Details

We first present the algorithmic details in Appendix B.1, followed by the entropy-estimation settings in Appendix B.2.

B.1 Algorithmic Details of FIM

Our implementation is available at `https://unistmlic.github.io/FIM/`.

Computing the influence Inf_t^d in Eq. (3) relies on predicting next-state outcomes. The dynamics model is initialized randomly and trained online together with the policy. FIM implements the transition model $\hat{s}$ as a three-layer MLP and minimizes the mean squared error:

$$\mathcal{L}_{\hat{s}} = \mathbb{E}_{s_t, \mathbf{a}_t, s_{t+1}} \left[\left\| \hat{s}(s_t, \mathbf{a}_t) - s_{t+1} \right\|^2 \right] \tag{16}$$

The FIM framework follows the CTDE paradigm, using QMIX to learn a joint action-value function. Each agent i has a Q-function $Q^i(\tau_t^i, a_t^i)$ based on its action-observation history τ_t^i and action a_t^i. These per-agent utilities are combined via a mixing network to produce a global joint Q-value, $Q_\theta^{\text{tot}}(s_t, \mathbf{a}_t)$, where θ denotes the parameters of the agent and mixing networks. To stabilize learning, FIM employs a target mixing network $Q_{\theta^-}^{\text{tot}}$, which is periodically updated by overwriting its parameters. The temporal difference (TD) loss is computed using a Bellman update:

$$\mathcal{L}_{\text{TD}}(\theta) = \mathbb{E}_{s, \mathbf{a}, r, s'} \left[\left(r_t + \gamma \max_{\mathbf{a}'} Q_{\theta^-}^{\text{tot}}(s_{t+1}, \mathbf{a}') - Q_\theta^{\text{tot}}(s_t, \mathbf{a}_t) \right)^2 \right] \tag{17}$$

The parameters θ are updated using the Adam optimizer, while the target network parameters θ^- are synchronized at fixed intervals. The complete training procedure of FIM is summarized in Algorithm 1.

Algorithm 1 FIM framework

Initialize: Q network, dynamics model $\hat{s}$
Collect trajectories under behavior policy
Approximate $\mathcal{H}(d)$ with the obtained trajectories based on Eq. (8)
Compute w_d for each $d \in \mathcal{D}$
for each training iteration **do**
  for each timestep t **do**
    Sample transition $(s_t, \mathbf{o}_t, \mathbf{a}_t, s_{t+1}, \mathbf{o}_{t+1})$ using $\boldsymbol{\pi}$,
      where $\mathbf{o}_t = (o_t^0, \cdots, o_t^{n-1})$
    for each $d \in \mathcal{D}$ **do**
      Compute influence Inf_t^d by Eq. (3)
      Update eligibility trace e_t^d by Eq. (4)
    end for
    Compute intrinsic reward r_t^{FIM} by Eq. (9)
  end for
  Update value function Q^{tot} and dynamics model $\hat{s}$
  if interval of 0.5M timesteps **then**
    Re-estimate $\mathcal{H}(d)$ under current behavior policy based on Eq. (8)
    Update w_d for each $d \in \mathcal{D}$ by Eq. (10)
  end if
end for

B.2 Empirical Estimation of $\mathcal{H}(d)$

To estimate $\mathcal{H}(d)$, we construct the empirical distribution $\hat{p}\left(\Delta^d(s_t, s_{t+1})\right)$ by counting occurrences discretized to two decimal places from 100K episodes collected under the current behavior policy. The entropy is then computed as:

$$\mathcal{H}(d) \approx \mathbb{E}_{\boldsymbol{\beta}}\left[-\log \hat{p}(\Delta^d(s_t, s_{t+1}))\right] \tag{18}$$

To ensure comparability across environments, $\mathcal{H}(d)$ values are min-max normalized to the range $[0, 1]$ within each environment.

C Environment Details

Push-2-Box: Push-2-Box is a cooperative multi-agent environment where two agents must jointly push one of two boxes toward a wall to obtain a reward, as depicted in Fig. 9. A box moves one cell if pushed by a single agent and two cells if pushed simultaneously. Thus, synchronized cooperation is essential for completing the task within the episode time limit. The environment terminates either when a box reaches the wall or when the episode length is exceeded. The **state space** consists of the (x, y) positions of agents and boxes, resulting in an 8-dimensional state vector. Each agent receives the full environment state as its observation. The **action space** is discrete, consisting of eight movement actions corresponding to up, down, left, right, top-right, right-down, down-left, and left-top directions. The **reward function** is described in Table 3.

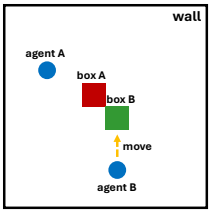


Figure 9: Push-2-Box

Table 3: Reward setting in Push-2-Box

Event	Reward
At least one box reaching the wall	+100 at episode end
No box reaching the wall	-1 at episode end

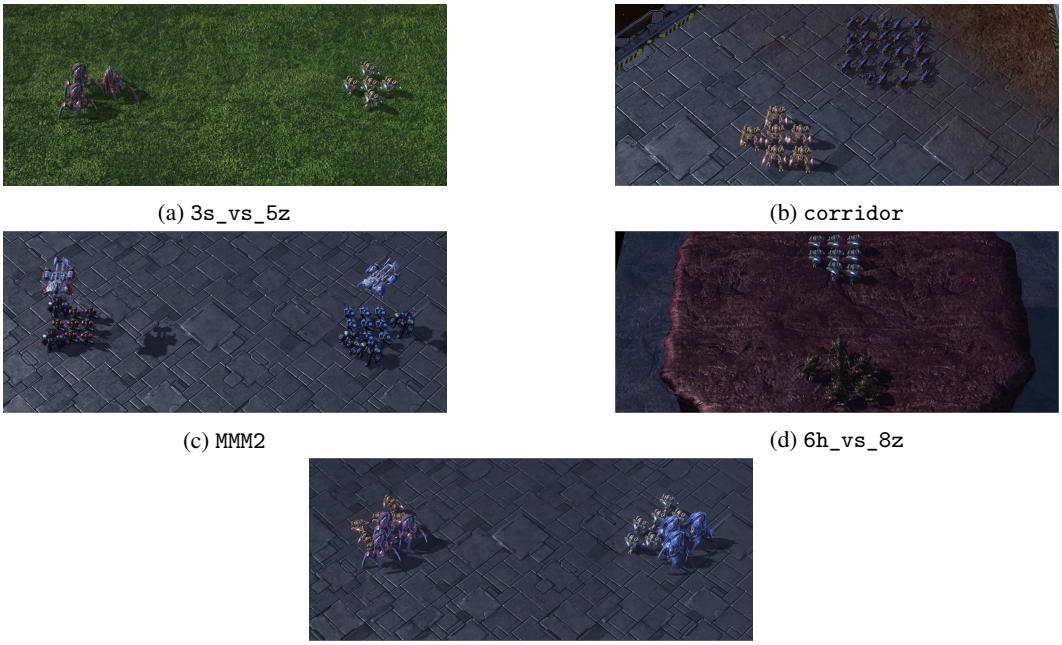
StarCraft Multi-Agent Challenge (SMAC & SMACv2): SMAC [68] is a benchmark for evaluating cooperative multi-agent reinforcement learning. Agents control individual StarCraft II units and coordinate to defeat enemy forces. Scenario visualizations are shown in Fig. 10, and unit compositions are summarized in Table 4. The **state space** contains global information about all units (e.g., positions, health, and shields), while the **observation space** provides each agent with local features of nearby units, including relative (x, y) positions and other attributes. The **action space** is discrete and consists of movement in four directions, attacking visible enemies, stopping, and a no-op action (only available to dead units). The **reward function** is summarized in Table 5, and our experiments focus on the sparse reward setting. We further evaluate on SMACv2 [69], an improved variant that introduces randomized unit compositions and initial positions, making coordination more challenging. The state, observation, and action spaces follow the same structure as SMAC, and we evaluate on three scenarios (`zerg_5_vs_5`, `terran_5_vs_5`, `protoss_5_vs_5`), with unit compositions provided in Table 4.

Table 4: SMAC and SMACv2 scenario configuration

Scenario	Ally	Enemy	State Dim	Obs Dim	Action Dim
`3s_vs_5z`	3 Stalkers	5 Zealots	68	48	11
`corridor`	6 Zealots	24 Zerglings	282	156	30
`MMM2`	1 Medivac, 2 Marauders, 7 Marines	1 Medivac, 3 Marauders, 8 Marines	322	176	18
`6h_vs_8z`	6 Hydralisks	8 Zealots	140	78	14
`3s5z_vs_3s6z`	3 Stalkers, 5 Zealots	3 Stalkers, 6 Zealots	230	136	15
`zerg_5_vs_5`	5 Zerg units	5 Zerg units	65	82	11
`terran_5_vs_5`	5 Terran units	5 Terran units	65	82	11
`protoss_5_vs_5`	5 Protoss units	5 Protoss units	75	92	11

Table 5: Sparse reward setting in SMAC and SMACv2

Event	Reward
Winning the battle	+1 at episode end
Losing the battle	-1 at episode end



(a) 3s_vs_5z (b) corridor

(c) MMM2 (d) 6h_vs_8z

(e) 3s5z_vs_3s6z

Figure 10: Visualization of SMAC scenarios.

(a) zerg_5_vs_5 (b) protoss_5_vs_5 (c) terran_5_vs_5

Figure 11: Visualization SMACv2 scenarios.

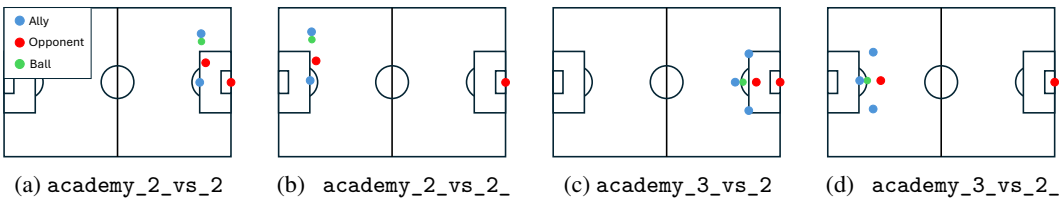
Google Research Football (GRF): GRF [70] provides a soccer simulation, in which agents must cooperate to score goals against an opponent team controlled by a scripted AI. We adopt a sparse reward setting to evaluate cooperative behavior under severely limited feedback. The **state space** consists of player positions and velocities, as well as ball position and velocity. Each **observation space** for an agent includes local information about the ego player, other players, and ball-related features, all expressed relative to the agent's current frame. The **action space** is discrete, covering movement in eight directions, sliding, passing, shooting, sprinting, and standing still. The **reward function** is described in Table 6. For brevity, several GRF scenarios are referred to using abbreviated names: `academy_3_vs_2` refers to `academy_3_vs_1_with_keeper`, `academy_2_vs_2` to `academy_run_pass_and_shoot_with_keeper`, `academy_counterattack` to `academy_counterattack_hard`, and `academy_4_vs_3` to `academy_4_vs_2_with_keeper` in the original GRF environment. As shown in Fig. 12, we design full-field variants of these scenarios by repositioning players to opposite half of the court. Table 7 provides an overview of the unit configurations.

Table 6: Reward setting in GRF

Event	Reward
Winning the match	+100 at episode end
Losing the match	-1 at episode end

Table 7: GRF scenario configuration

Scenario	Ally	Opponent	State Dim	Obs Dim	Action Dim
academy_2_vs_2	2 center back	1 goalkeeper, 1 center back	22	22	19
academy_2_vs_2 _full_field	2 center back	1 goalkeeper, 1 center back	22	22	19
academy_3_vs_2	3 central midfield	1 goalkeeper, 1 center back	26	26	19
academy_3_vs_2 _full_field	3 central midfield	1 goalkeeper, 1 center back	26	26	19
academy_4_vs_3	4 central midfield	1 goalkeeper, 2 center back	34	34	19
academy_4_vs_3 _full_field	4 central midfield	1 goalkeeper, 2 center back	34	34	19
academy _counterattack	1 central midfield, 1 left midfield, 1 right midfield, 1 central front	1 goalkeeper, 2 center back	34	34	19
academy _counterattack _full_field	1 central midfield, 1 left midfield, 1 right midfield, 1 central front	1 goalkeeper, 2 center back	34	34	19



(a) academy_2_vs_2 (b) academy_2_vs_2_full_field (c) academy_3_vs_2 (d) academy_3_vs_2_full_field

(e) academy_4_vs_3 (f) academy_4_vs_3_full_field (g) academy_counter attack (h) academy_counter attack_full_field

Figure 12: Visualization of initial agent positions in GRF scenarios.

D Experimental Details

FIM is implemented on top of the open-source framework from [79], which is also used to run QMIX [8] and QPLEX [9]. LAIES [17], RODE [71], MASER [32], CDS [14], and FoX [30] are evaluated using the original code and settings provided by their respective authors. Experiments are conducted on an NVIDIA RTX 3090 GPU with an Intel Xeon Gold 6348 CPU (Ubuntu 20.04). For the original benchmarks, training completes within two days for Push-2-Box and SMAC, while each GRF scenario requires less than two days to reach 5 million timesteps. We begin by describing the baseline algorithms in Appendix D.1 and outline the hyperparameter setup of FIM in Appendix D.2.

D.1 Detailed Description of Baseline Algorithms

- **QMIX** [8] factorizes the joint action-value into individual utilities combined by a monotonic mixing network. Code: `https://github.com/hijkzzz/pymarl2`
- **QPLEX** [9] extends QMIX with a duplex dueling architecture, decomposing joint value into individual value and advantage while enforcing the IGM principle. Code: `https://github.com/hijkzzz/pymarl2`
- **LAIES** [17] incentivizes agents to influence external task-relevant states via intrinsic rewards for both individual and joint impacts. Code: `https://github.com/liuboyin/LAIES`
- **RODE** [71] employs hierarchical role-based policies where agents periodically select roles to guide low-level actions, enabling scalable specialization. Code: `https://github.com/TonghanWang/RODE`
- **MASER** [32] explores by assigning subgoals from past trajectories, rewarding agents for revisiting informative states. Code: `https://github.com/Jiwonjeon9603/MASER`
- **CDS** [14] encourages policy diversity by maximizing mutual information between agent identity and trajectory. Code: `https://github.com/lich14/CDS`
- **FoX** [30] promotes exploration by maximizing entropy of agent formations and their mutual information with team structure. Code: `https://github.com/hyeon1996/FoX`
- **COMA** [38] is a multi-agent policy gradient method that assigns credit using a centralized critic with counterfactual baselines. Code: `https://github.com/oxwhirl/pymarl`
- **MA^{2}E** [72] addresses partial observability with a masked autoencoder to infer global information from individual observations. Code: `https://github.com/cheesebro329/MA2E`

D.2 Hyperparameter Setup of the Proposed FIM

Scenario-specific settings for the intrinsic reward weight α and update rate ϕ are provided in Table 9, while the softmax temperature in $\mathrm{Softmax}(-\mathcal{H}(d))$ is set to 0.1, and the discount factor γ is fixed at 0.99 across all scenarios. The default hyperparameter settings of FIM, which are generally shared across scenarios, are summarized in Table 8.

Table 8: Common hyperparameter settings of FIM

Hyperparameters	Value
Optimizer	Adam
Replay buffer size	5000
Target update interval	200
Mini-batch size	32
Mixing network dim	32
Discount factor γ	0.99
Learning rate	0.0005
Dynamics model $\hat{s}(\cdot)$ layers	3
Dynamics model $\hat{s}(\cdot)$ dim	128

Table 9: Scenario-specific hyperparameter setup of FIM

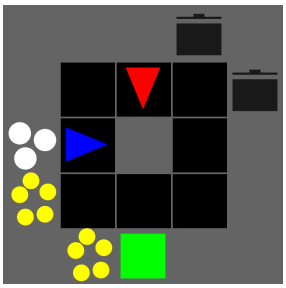
Scenario	α	ϕ
Push-2-Box	5	0.05
StarCraft Multi-agent Challenge (Sparse)		
3s_vs_5z	10	0.05
corridor	20	0.05
MMM2	10	0.05
6h_vs_8z	20	0.05
3s5z_vs_3s6z	10	0.05
StarCraft Multi-agent Challenge v2 (Sparse)		
protoss_5_vs_5	10	0.05
terran_5_vs_5	10	0.05
zerg_5_vs_5	10	0.05
Google Research Football (Sparse)		
academy_2_vs_2	10	0.05
academy_2_vs_2_full_field	10	0.05
academy_3_vs_2	10	0.05
academy_3_vs_2_full_field	10	0.05
academy_4_vs_3	10	0.05
academy_4_vs_3_full_field	10	0.05
academy_counterattack	10	0.05
academy_counterattack_full_field	10	0.05

E Additional Experiments

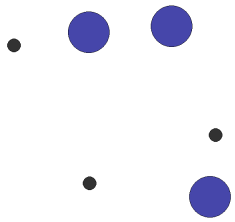
We evaluate alternative state representations and continuous actions in Appendix E.1, followed by SFI-weight visualizations in Appendix E.2.

E.1 Generalization Across State and Action Spaces

We test FIM beyond structured vector states and discrete-action value decomposition using spatial tensor states in Overcooked and continuous actions in MPE (Figure 13).



(a) Overcooked: `coord-ring`



(b) MPE: `simple_spread_v3`

Figure 13: Illustrative states of the additional environments. (a) Two agents prepare and deliver soup in a ring-shaped kitchen. (b) Agents (blue) cover landmarks (gray) while avoiding collisions.

Spatial tensor states in Overcooked. We evaluate Overcooked [73] `coord-ring` using the JaxMARL [80] implementation. Unlike structured representations that explicitly provide object coordinates, the state is a height $\times$ width $\times$ components tensor, with agents and ingredients encoded over a spatial grid. The soup location remains under-explored until preparation is complete. As training progresses, agents explore different positions with the soup and learn efficient goal-directed trajectories. With $\alpha = 1$ and $\phi = 0.05$, FIM improves IPPO's return (Table 10). The observations are spatial binary encodings, not raw images.

Continuous actions in MPE. We use the Multi-Agent Particle Environment (MPE) [74], specifically PettingZoo [75] `simple_spread_v3`, with continuous actions and a MAPPO [6] backbone. Agents coordinate to cover landmarks while avoiding collisions. Continuous actions preclude enumerating a finite action set for counterfactual evaluation. We therefore estimate the expectation in Eq. (3) by Monte Carlo sampling from the continuous policy: for each agent, we sample alternative actions while holding the other agents' actions fixed, and average the magnitudes of the predicted state changes. This preserves the influence-based reward and its SFI/AFI focusing mechanisms while changing how the counterfactual expectation is computed. After 1M training timesteps, FIM improves MAPPO's return (Table 10), supporting applicability to this continuous-action task. Reward scales differ across environments; comparisons are made within each setting.

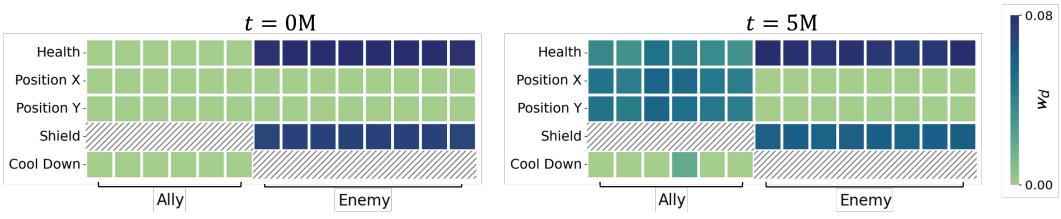
Table 10: Returns on Overcooked and MPE, comparing each backbone with and without FIM.

Method	Return
Overcooked — `coord-ring`	
IPPO	195.7 ± 5.3
IPPO + FIM (ours)	$\mathbf{218.5 \pm 4.7}$
MPE — `simple_spread_v3`	
MAPPO	-80.7 ± 19.6
MAPPO + FIM (ours)	$\mathbf{-63.4 \pm 7.9}$

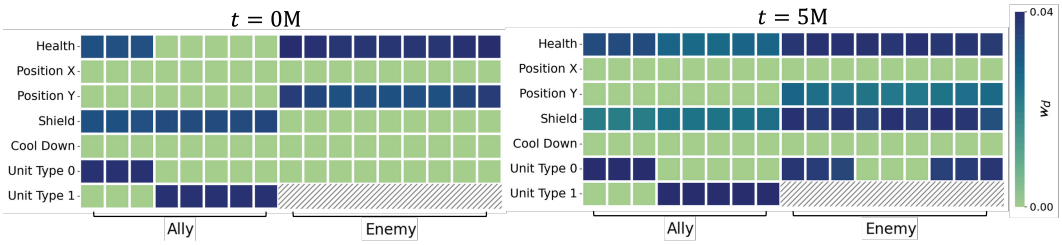
E.2 Visualization of SFI Weights w_d

Fig. 14 and Fig. 15 visualize w_d changes as training proceeds in SMAC and SMACv2, respectively. Across both benchmarks, SFI consistently assigns the highest weights to enemy health dimensions throughout training, reflecting their status as under-explored state dimensions that change substantially only when multiple agents coordinate their attacks on the same target. This makes enemy health a natural focal point for joint influence, as reducing it requires sustained and synchronized action from the team. Ally-specific features such as ally health also receive progressively elevated weights as training advances. This is a consequence of agents' tendency to avoid direct combat in order to preserve their own health, which causes ally health dimensions to remain relatively stable and thus under-explored. By up-weighting these dimensions, SFI counteracts this avoidance behavior and encourages agents to engage more actively in battle, ultimately promoting more effective coordination.

Fig. 16 visualizes w_d in GRF, where the opponent goalkeeper's position and direction dimensions are consistently assigned the highest weights throughout training. The goalkeeper's state exhibits persistently low entropy because it remains largely stationary under the joint behavior policy, shifting only in response to direct pressure from attacking agents. Consequently, SFI identifies these dimensions as primary targets, concentrating the team's cooperative effort on a state that is both stable and impactful. A detailed account of how this weight assignment translates into concrete agent behaviors is provided in Appendix F.1.

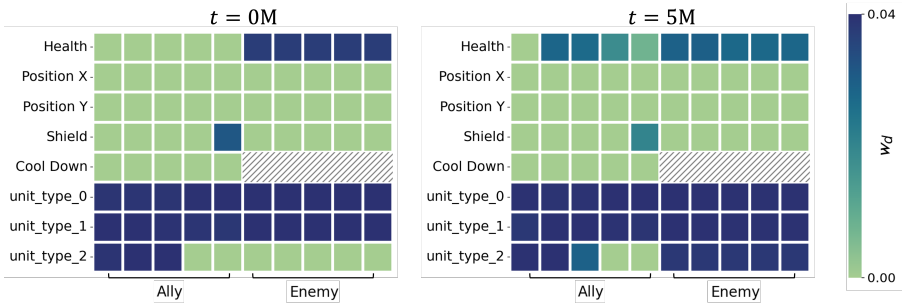


(a) `6h_vs_8z`

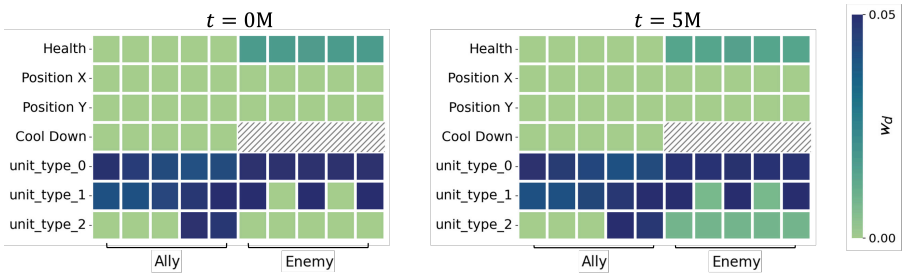


(b) `3s5z_vs_3s6z`

Figure 14: SMAC w_d visualization

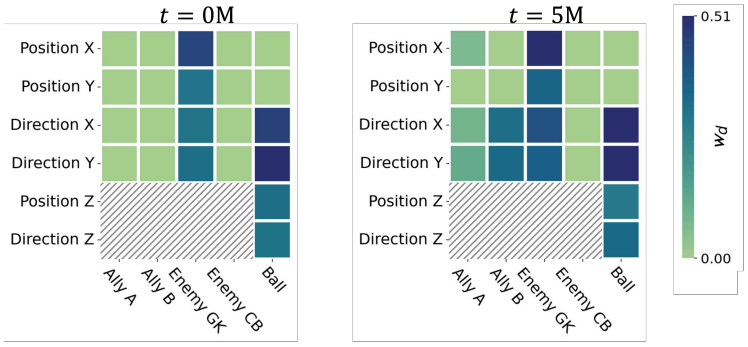


(a) `protoss_5_vs_5`

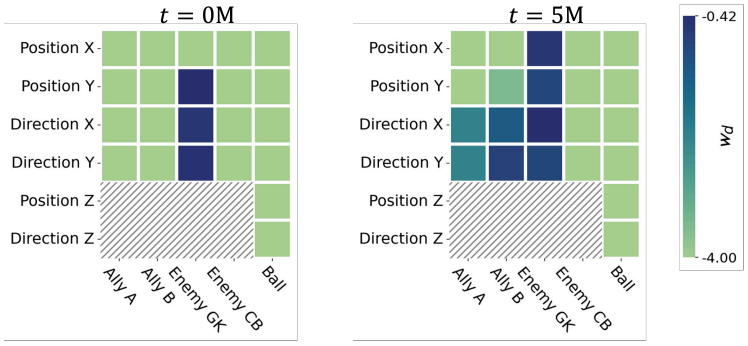


(b) `terran_5_vs_5`

Figure 15: SMACv2 w_d visualization



(a) `academy_2_vs_2`



(b) `academy_2_vs_2_full_field`

Figure 16: GRF w_d visualization

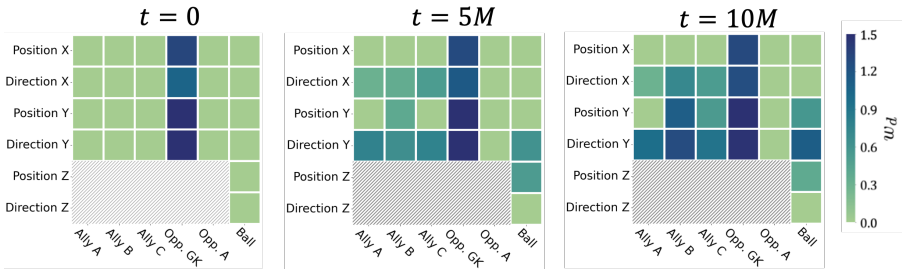
F Additional Analysis and Ablation Studies

We present GRF behavior analysis in Appendix F.1, extended ablations, including the influence-definition comparison, in Appendix F.2, dynamics-model analysis in Appendix F.3, computational costs in Appendix F.4, entropy-surrogate validation in Appendix F.5, and entropy-estimation sensitivity in Appendix F.6.

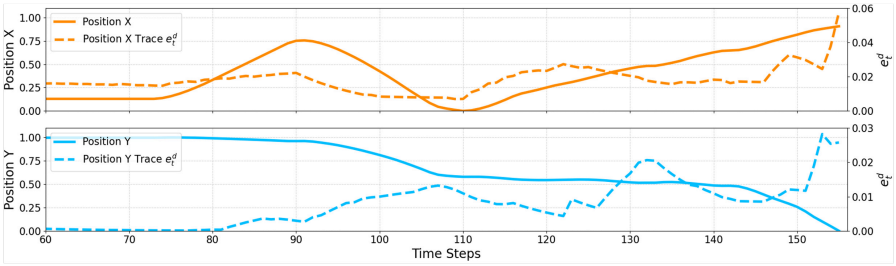
F.1 Strategy Analysis in GRF

In GRF, the dimension weights w_d consistently focus on the opponent goalkeeper's position and direction throughout training, as illustrated in Fig. 17(a). These dimensions exhibit low entropy throughout the training, since the goalkeeper typically remains stationary and only shifts position when a ball-carrying agent approaches the goalpost. This characteristic makes the goalkeeper's state both stable and strategically significant, as displacing it creates scoring opportunities and thus serves as a valuable proxy objective in sparse reward settings. Accordingly, FIM guides agents to influence the goalkeeper's position.

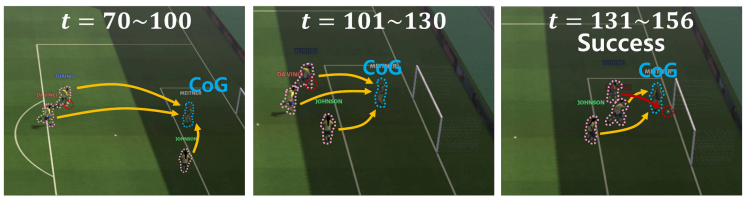
As shown in Fig. 17(b)-(c), this insight is reflected in the agent trajectory. Around $t \approx 70$ within the trajectory, agents begin to receive intrinsic rewards by subtly influencing the goalkeeper's position, even while positioned far from the goal area. By $t \approx 100$, the accumulated eligibility traces further incentivize agents to continue exerting influence over the goalkeeper, enabling a gradual progression toward the goal. Near $t \approx 130$, the goalkeeper briefly moves out of position, and the attacking agent capitalizes on this opportunity to score. Notably, FIM guides agents to approach the goal proactively and maintain persistent influence over the goalkeeper's positioning, which is an under-explored region under the joint behavior policy.



(a) Dimension weights w_d across iterations



(b) Changes in goalkeeper position and its trace e_t^d



(c) Rendered frames for highlighting agents' coordination

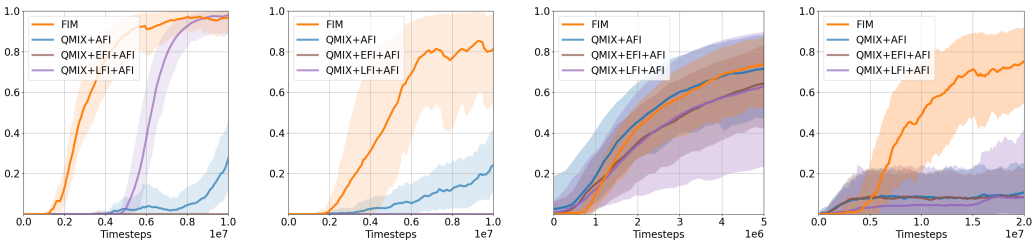
Figure 17: Analysis of FIM's strategy in GRF `academy_3_vs_2_full_field`

F.2 Extended Ablation Studies

To further evaluate the robustness of FIM, we conduct experiments on four challenging scenarios: SMAC `3s_vs_5z`, SMAC `3s5z_vs_3s6z`, GRF `academy_3_vs_2`, and GRF `academy_3_vs_2_full_field`, focusing on (i) alternatives to the SFI state focusing mechanism, (ii) the impact of each module in FIM through a component ablation study, (iii) the reward scaling factor α, and (iv) the update rate ϕ.

Alternatives to the SFI State Selection Mechanism: We investigate alternative strategies to the SFI state focusing mechanism for weighting state dimensions to influence: *uniform-weighting influence*, *external state focusing influence* (EFI), and *least-change state focusing influence* (LFI). In all cases, the intrinsic reward is computed as in FIM, with each variant differing only in weighting the state dimensions in $\mathcal{D}$. The uniform-weighting variant sets $w_d = 1$ for all $d \in \mathcal{D}$, which is equivalent to Eq. 5. EFI only weights external state features, following the approach of LAIES [17]: enemy health, shield, and positions in SMAC; and opponent and ball positions and directions in GRF. LFI assigns higher weights to state dimensions with smaller average temporal changes $|\tilde{s}^d_{t+1} - \tilde{s}^d_t|$ via softmax. In SMAC, this typically includes enemy health and ally positions, while in GRF, it often corresponds to ally direction features due to their relatively small-scale temporal changes.

As shown in Fig. 18, while some SFI variants show comparable performance in (a) and (c), the state dimensions weighted by SFI consistently lead to the highest overall performance. When variants include easily influenced features such as ally position, agents tend to exploit these trivial dimensions, leading to reward hacking and suboptimal behavior. These findings underscore the effectiveness of FIM's entropy-based weighting, which highlights stable and causally meaningful under-explored dimensions.



(a) `3s_vs_5z` (b) `3s5z_vs_3s6z` (c) `academy_3_vs_2` (d) `academy_3_vs_2_full_field`

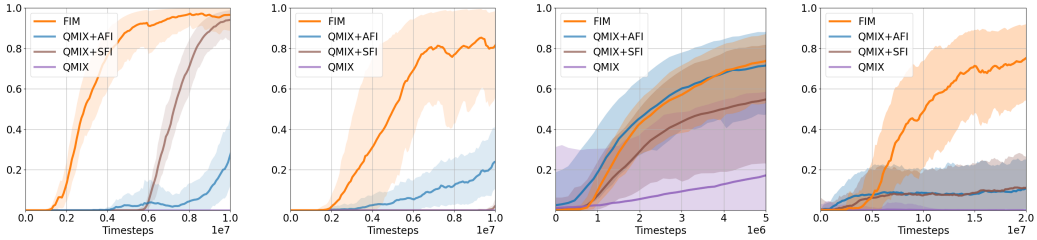
Figure 18: Alternatives to SFI

Effect of the Influence-Measure Definition. Complementing the state-selection alternatives above, we isolate the definition of influence by retaining FIM's SFI and AFI mechanisms and replacing only its influence measure with that of LAIES [17]. On SMACv2 `terran_5_vs_5`, this variant reaches 0.48 ± 0.12, compared with 0.59 ± 0.09 for FIM and 0.38 ± 0.14 for LAIES (Table 11). The comparison separates how influence is measured from how it is weighted and accumulated: the intermediate result supports a benefit from focusing even with the alternative measure, while FIM's state-change-magnitude definition provides a further gain in this experiment.

Table 11: Influence-definition ablation on SMACv2 `terran_5_vs_5`: final win rates.

Method	Win rate
LAIES	0.38 ± 0.14
FIM with the LAIES influence measure	0.48 ± 0.12
FIM (ours)	$\mathbf{0.59 \pm 0.09}$

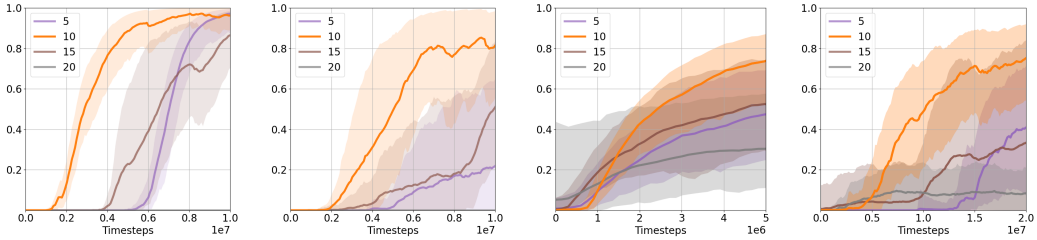
Component Evaluation: Fig. 19 compares four variants: vanilla QMIX, QMIX with state focusing influence (SFI), QMIX with agent focusing influence (AFI), and the full FIM framework that integrates both components. SFI improves performance by narrowing exploration to under-explored state dimensions, while AFI promotes deeper exploration by sustaining coordinated influence toward shared targets. Although each component contributes independently, only their combination yields the highest success rates.



(a) `3s_vs_5z` (b) `3s5z_vs_3s6z` (c) `academy_3_vs_2` (d) `academy_3_vs_2_full_field`

Figure 19: Component evaluation

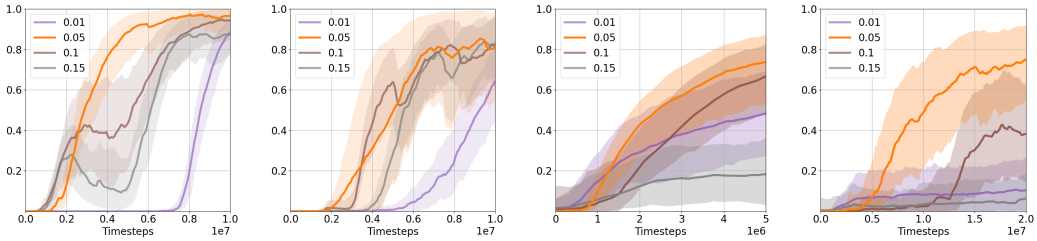
Reward scaling factor α: We examine how the reward scaling factor α affects performance by testing values in $\alpha \in \{5, 10, 15, 20\}$, as shown in Fig. 20. When α is too small, the intrinsic reward signal becomes negligible, preventing agents from effectively learning the influence strategy promoted by FIM. Conversely, setting α too large causes agents to over-prioritize intrinsic rewards, ignoring critical environmental feedback. Thus, we select $\alpha = 10$ as the default value that is well aligned with the extrinsic reward scale, guiding exploration without overwhelming the task objective.



(a) `3s_vs_5z` (b) `3s5z_vs_3s6z` (c) `academy_3_vs_2` (d) `academy_3_vs_2_full_field`

Figure 20: Reward scaling factor α

Update rate ϕ: We examine how the dimension-weight update rate ϕ affects performance by testing values in $\phi \in \{0.01, 0.05, 0.1, 0.15\}$, as shown in Fig. 21. When ϕ is too small, the dimension weights w_d adapt too slowly to the evolving behavior policy, failing to reflect which dimensions are currently under-explored and thus delaying effective coordination. Conversely, setting ϕ too large causes frequent weight shifts that destabilize the focus target, preventing agents from sustaining coordinated influence on any single dimension. Thus, we select $\phi = 0.05$ as the default, which strikes the best balance between stable and responsive adaptation to the current policy.



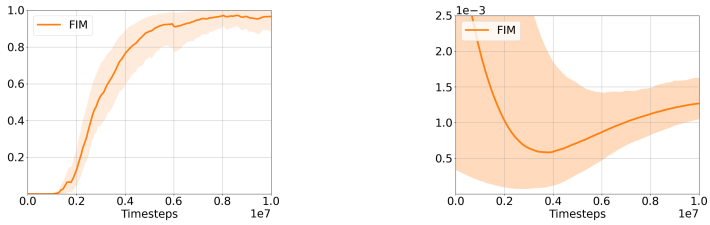
(a) `3s_vs_5z` (b) `3s5z_vs_3s6z` (c) `academy_3_vs_2` (d) `academy_3_vs_2_full_field`

Figure 21: Update rate ϕ

F.3 Analysis of the Learned Dynamics Model

Since the intrinsic reward in FIM is computed from the predictions of the learned dynamics model $\hat{s}$, its accuracy directly influences the reward signal. While a high mean-squared error (MSE) might seem detrimental, our results suggest that prediction inaccuracies can also serve a constructive role by implicitly encouraging exploration of regions with complex or less predictable dynamics. In this sense, model error may act as a form of curiosity, resonating with ideas from curiosity-driven exploration in model-based RL [77].

To examine this effect empirically, we analyzed the SMAC `3s_vs_5z` scenario. As shown in Fig. 22, the forward model's MSE initially decreased and then gradually increased during training, likely reflecting exposure to more diverse transitions. Notably, this trend coincided with a steady improvement in win rate, suggesting that moderate prediction error did not destabilize learning but rather correlated with productive exploration, ultimately supporting performance gains.



(a) FIM Performance (b) Mean squared error loss of $\hat{s}$

Figure 22: Comparison of FIM performance and mean squared error loss of $\hat{s}$ in `3s_vs_5z`

F.4 Comparison of Computational Complexity

FIM computes intrinsic rewards by estimating each agent's influence through counterfactual marginalization over its action set $\mathcal{A}$ for every dimension in $\mathcal{D}$. Storing all counterfactual state predictions requires $O(|\mathcal{N}| \cdot |\mathcal{A}| \cdot |\mathcal{D}|)$ memory per timestep. GPU batching reduces the wall-clock overhead of counterfactual evaluations, while the total arithmetic work still scales with the number of evaluations and state dimensions. FIM uses a lightweight three-layer multilayer perceptron (MLP) as the forward transition model and does not alter the main Q-network architecture, keeping computational overhead minimal. We compare FIM against QMIX with dense rewards (QMIX-DR). As shown in Table 12, FIM's average computation time per 1 million timesteps is comparable to QMIX-DR. In `3s5z_vs_3s6z`, FIM also requires fewer timesteps to reach a 60% success rate, demonstrating strong efficiency. These results show improved performance with modest wall-clock overhead in the evaluated scenarios.

Table 12: Computational cost: training time per 1M timesteps and steps to reach a 60% success rate.

Metric	QMIX-DR	**FIM (ours)**
SMAC — `3s_vs_5z`		
Time per 1M steps (min)	70.65	72.40
Steps to 60% success	4.21M	3.24M
SMAC — `3s5z_vs_3s6z`		
Time per 1M steps (min)	123.23	126.43
Steps to 60% success	13.05M	5.32M

F.5 Marginal and Conditional Entropy

SFI uses marginal transition entropy to rank state dimensions, whereas the theoretical motivation concerns entropy conditioned on the current state. We therefore compare $\mathcal{H}(\Delta^d)$ and $\mathcal{H}(\Delta^d \mid s_t)$, where $\Delta^d = \Delta^d(s_t, s_{t+1})$, on SMAC `3s5z_vs_3s6z` and GRF `academy_2_vs_2`. Conditioning can remove variation attributable to the current state, so these quantities need not be equal. We therefore examine whether the marginal surrogate preserves SFI's low-entropy ranking.

Table 13 shows numerically close estimates on both benchmarks. In SMAC, the dimension groups have the same ordering under both measures, with enemy health receiving the lowest entropy. In GRF, rankings are similar, and the low-entropy ball and goalkeeper features remain distinct from the more variable player features. These comparisons support the marginal surrogate in the evaluated settings; they do not establish equivalence for arbitrary state distributions.

Table 13: Marginal and conditional transition entropy for the listed feature groups. Shaded rows distinguish environments; GK denotes goalkeeper.

Feature group	Marginal $\mathcal{H}(\Delta^d)$	Conditional $\mathcal{H}(\Delta^d \mid s_t)$
SMAC — `3s5z_vs_3s6z`		
Enemy health	0.06	0.05
Enemy shield	0.61	0.57
Enemy position X	0.70	0.65
Enemy position Y	0.09	0.08
Ally health	0.40	0.37
Ally shield	0.12	0.10
Ally cooldown	0.88	0.83
Ally position X	0.74	0.69
Ally position Y	0.66	0.61
GRF — `academy_2_vs_2`		
Ally A position	0.73	0.72
Ally A direction	0.82	0.80
Ally B position	0.73	0.71
Ally B direction	0.78	0.73
Opponent GK position	0.27	0.21
Opponent GK direction	0.32	0.26
Opponent A position	0.74	0.70
Opponent A direction	0.96	0.94
Ball position	0.47	0.45
Ball direction	0.24	0.23

F.6 Sensitivity to Entropy Estimation

We examine whether SFI's preferred feature group depends strongly on the numerical choices used to estimate entropy. Starting from 2-decimal discretization, 100K episodes, and min–max normalization, we vary one factor at a time: discretization precision, the number of estimation episodes, or the normalization method. Discretization controls the resolution of empirical transition counts, the episode count controls the amount of estimation data, and normalization changes the scale on which transitions are compared.

Table 14 reports the top-weighted group at the start of training and after 5M timesteps on `3s5z_vs_3s6z`. Enemy health remains the highest-priority group for every tested setting at both checkpoints. This supports stable feature prioritization across these estimation choices. This comparison assesses feature-group selection at two checkpoints.

Table 14: Top-weighted state group under different entropy-estimation settings on SMAC `3s5z_vs_3s6z`. Only the named factor changes from the default.

Estimation setting	Top-weighted at 0M	Top-weighted at 5M
Default	Enemy health	Enemy health
1-decimal discretization	Enemy health	Enemy health
3-decimal discretization	Enemy health	Enemy health
50K estimation episodes	Enemy health	Enemy health
150K estimation episodes	Enemy health	Enemy health
No normalization	Enemy health	Enemy health
z-score normalization	Enemy health	Enemy health

G Broader Impact

This work advances cooperative multi-agent systems by introducing a framework that fosters coordinated behavior through influence-based intrinsic motivation. Enhanced cooperation among agents holds strong potential for positive societal impact in domains such as autonomous vehicle coordination, collaborative robotics, disaster response, and environmental monitoring. In these settings, the ability of agents to reason about and influence under-explored aspects collaboratively can lead to more robust, adaptive, and efficient team performance. As a foundational contribution, this research supports the development of AI systems that are better aligned with collective goals, promoting safer and more effective deployment in real-world multi-agent environments.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Section 1) clearly and accurately reflect the paper's main contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses two limitations of FIM: additional computational overhead from training the dynamics model, and limited verification for image-based environments.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: For the theoretical result, we state the assumption and provide complete proof in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: The method and training procedure are described in Section 4 and Appendix B; benchmark settings, baseline sources, and hyperparameters are provided in Appendices C and D.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our implementation is available at `https://unistmlic.github.io/FIM/`. The method and training procedure are described in Appendix B, with benchmark settings, baseline sources, and hyperparameters in Appendices C and D.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides all necessary training and testing details, including hyperparameter settings, ablation studies, and optimizer specifications.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Performance plots report the mean across 5 random seeds with shaded areas denoting standard deviation, as stated in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section D specifies the GPU/CPU setup and Section F.4 discusses the time of execution.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses standard simulation benchmarks without involving human subjects or sensitive data and adheres to NeurIPS ethical guidelines throughout.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: Appendix G discusses potential societal benefits. Negative societal impacts and mitigation strategies are not analyzed in this work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no significant risk of misuse, as it does not release models or datasets with dual-use concerns.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The environments and baselines are cited, with baseline code URLs in Appendix D.1. A complete account of the asset versions and license terms is not yet included.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our implementation is available at `https://unistmlic.github.io/FIM/`. Its method, training procedure, and experimental settings are documented in Section 4 and Appendices B–D.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research does not involve any human subjects or crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve experiments with human participants requiring IRB or equivalent approval.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used in the development of the core methodology of this work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.