2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Safe Offline-to-Online Multi-Agent Decision Transformer: A Safety Conscious Sequence Modeling Approach

Aamir Bader Shah[1]*, Yu Wen[1], Jiefu Chen[1], Xuqing Wu[2] and Xin Fu[1]

***Abstract*— We introduce the Safe Offline-to-Online Multi-Agent Decision Transformer (SO2-MADT), an innovative framework that revolutionizes safety considerations in Multi-agent Reinforcement Learning (MARL) through a novel sequence modeling approach. Leveraging the dynamic capabilities inherent in Decision Transformers, our methodology seamlessly incorporates safety protocols as a cornerstone element, ensuring secure operations throughout both the offline pre-training phase and the adaptive online fine-tuning phase. At the core of our framework lie two pivotal innovations: the Safety-To-Go (STG) token, embedding safety at a macro level, and the Agent Prioritization Module (APM), facilitating explicit credit assignment at a micro level. Through extensive testing against the challenging environments of the StarCraft Multi-Agent Challenge (SMAC) and Multi-agent MuJoCo, our SO2-MADT not only excels in offline pre-training but also demonstrates superior performance during online fine-tuning, without any degradation in performance. The implications of our work provide a pathway for deployment in critical real-world applications where safety is paramount and non-negotiable. The code is available at `https://github.com/shahaamirbader/SO2-MADT`.**

I. INTRODUCTION

Multi-agent reinforcement learning (MARL) [1] has significantly advanced in recent years, empowering agents to tackle intricate tasks ranging from coordinating autonomous vehicles [2], to mastering multi-player strategy games [3], and orchestrating collaborative multi-robot systems [4]. However, safety concerns have significantly constrained the practical application of MARL in real-world scenarios, as agents may inadvertently execute actions that pose risks to themselves, other agents, or the environment, limiting their usability. Take bionic amphibious robots [5] as an example, while these robots may be capable of utilizing online learning effectively on land without extra risk controls, the challenging conditions of underwater exploration like restricted communication and visibility necessitate a greater reliance on safe offline learning for pre-trained actions with extra consideration. This underscores the critical importance of ensuring that all agents consistently adhere to safe policies, regardless of the learning method employed.

A key challenge in MARL is the seamless integration of safety constraints throughout both offline and online learning processes. This is crucial because the traditional MARL setups [6] prioritize agents' primary objective of maximizing rewards, which often leads to an unrestricted exploration in un-safe zones for reward maximization, as illustrated in Figure 1(a). In contrast, safe RL enforces a safety threshold as a hard boundary, ensuring agents operate strictly within safe zones, as shown in Figure 1(b). This comparison highlights the lack of inherent safety mechanisms in the traditional RL approach, potentially leading to irresponsible agent behavior.

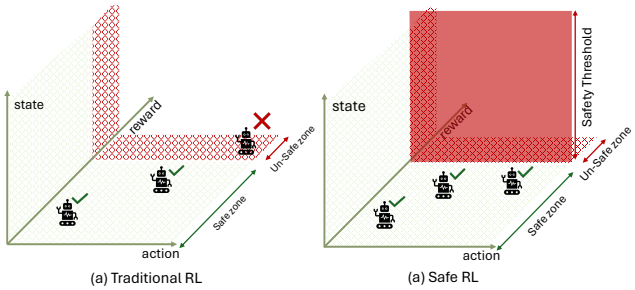

Fig. 1: Comparison of (a) Traditional RL with Risky Actions and (b) Safe RL with Safety Constraints.

A naive solution to this issue involves adopting a centralized approach, treating the multi-agent system as a single entity. While this enables the utilization of safe RL algorithms [7], [8], it exposes inherent scalability limitations. The exponential expansion of the joint action space with additional agents results in severe computational bottlenecks, making the approach impractical for real-world applications.

To address the limitations of the naive approach, one prevalent strategy in the field is the Centralized Training with Decentralized Execution (CTDE) paradigm [9], [10]. Under the CTDE paradigm, agents are centrally trained with global state access to optimize the team's joint action value by maximizing each agent's action-value function. Although CTDE demonstrates practical utility, it still presents several challenges. Firstly, the inherent non-stationarity of the environment from the perspective of each agent, shaped by the collective actions of all agents, may lead to convergence and stability issues. Secondly, the implicit methods of aggregating local agent values into a global Q-value often fail to provide clear feedback on individual agent performance. This results in the credit assignment problem [11], significantly hindering policy optimization for decentralized agents. Moreover, when applying the CTDE paradigm to bionic amphibious robots, the successful task outcomes will depend on specific sequences of actions coordinated among multiple robots. However, implicit credit assignments within CTDE may obscure the individual contribution of each agent, impeding combined learning and future coordination optimization for each robot. To address this issue, explicit

*Corresponding author: Aamir Bader Shah (ashah29@cougarnet.uh.edu)
[1]Aamir Bader Shah, Yu Wen, Jiefu Chen, Xin Fu are with the Department of Electrical and Computer Engineering, University of Houston, USA
[2]Xuqing Wu is with the Department of Information Science Technology, University of Houston, USA

credit assignment provides direct feedback, enabling the system to prioritize actions from critical robots within the sequence to maximize performance.

The recent integration of Transformer in RL represents a promising advancement [12], given their exceptional proficiency in capturing long-range dependencies within sequential data [13]. This capability makes transformers ideally suited for addressing sequential challenges such as RL, where past experiences play a pivotal role in determining optimal actions. Researchers have utilized the generative trajectory modeling capabilities of transformers to tackle RL as a sequence modeling problem, where Decision Transformer (DT) [14] stands out as one of the pioneering models to effectively address traditional RL tasks using sequence modeling, marking a significant breakthrough in the field. To effectively balance the overall system objectives, We adopt a dual-level sequence modeling approach to effectively balance the overall system objectives. At the macro level, we implement decision-making that governs collective actions and strategic goals, utilizing DT architecture for both offline pre-training and online fine-tuning while integrating safety constraints. At the micro level, we focus on detailed actions and contributions of individual agents within the MARL framework, incorporating explicit credit assignment and agent prioritization using DT's attention mechanism. This hierarchical approach optimizes both safety and efficiency in the multi-agent environment.

In this paper, we introduce SO2-MADT (Safe Offline to Online Multi-Agent Decision Transformer), a novel DT-based architecture designed for safety-constrained MARL, aiming to enhance MARL with powerful yet safe sequential modeling techniques. To our knowledge, SO2-MADT is the first sequence modeling approach for MARL that seamlessly integrates safety constraints during both offline and online learning. Our goal is to develop a practical RL framework that enables agents to learn safe actions through offline pre-training and then consistently adhere to those safety constraints during online exploration. We evaluate SO2-MADT on two multi-agent benchmark datasets: StarCraft Multi-agent Challenge (SMAC) [16] and Multi-agent MuJoCo [15] to evaluate its performance against state-of-the-art (SOTA) works. Our contributions can be summarized as follows:

- We address the safe offline RL problem using supervised learning and propose a novel safety-constrained DT-based sequential model that seamlessly incorporates safety into both offline and online learning.
- We introduce the Safety-To-Go (STG) token for macro-level safety monitoring, enabling agents to adjust actions proactively and prevent safety violations by quantifying their remaining risk tolerance.
- At the micro level, we propose the Agent Prioritization Module (APM) to enhance individual agent accountability, utilizing attention weights to assess contributions and steer decision-making towards optimal outcomes.
- Our comprehensive experiments on two MARL benchmark datasets demonstrate that our method i) ensures safety without compromising task performance, and ii) outperforms several competing MARL approaches on both safety and overall metrics.

II. RELATED WORK

A. Safe Reinforcement Learning

Constrained optimization plays a crucial role in addressing safe RL problems [34]. There are two commonly employed strategies: Lagrangian-based methods, which utilize a multiplier to penalize violations of constraints [35], [36]; and correction-based approaches, which project unsafe actions onto a safe set and often integrate domain-specific knowledge to facilitate safer exploration [37]. SaFormer [22] and CDT [33] are recent works based on DT, that address safety in offline RL by introducing constraint tokens. However, they are limited to single-agent scenarios.

B. Pre-trained Offline Reinforcement Learning

Recent developments in both offline and online RL have garnered significant attention. Although offline RL may derive effective strategies from static datasets, it faces a significant challenge known as exploration error compared to online RL. This occurs when learned policies generate actions beyond the distribution of the dataset [17], [18]. To bridge the gap between static offline datasets and dynamic online interactions, researchers have been actively working and proposed several methods like [17], [19]. Moreover, with the introduction of transformer architectures into offline RL, different methods have been developed based on novel architectures like Decision Transformer [14] and Trajectory Transformer [20]. Additionally, ODT [26] utilizes the sequence modeling nature of DT to incorporate stochastic policies for single-agent scenarios, while MADT [21] pre-trains a generalized policy on offline datasets and then integrates online multi-agent RL for universal policy training. Conversely, while these methods leverage advanced architectures, they fall short in addressing critical aspects such as safety constraints and explicit credit assignment.

C. Credit Assignment in Reinforcement Learning

In cooperative MARL, effective coordination relies on appropriately assigning a shared team reward based on the contribution of each agent, i.e., the credit assignment problem. Traditional implicit credit assignment methods, commonly used for reward decomposition, often lack clear guidance for policy optimization due to their inherent ambiguity. For example, VDN [23], with its linear decomposition approach, disregards crucial state information, while QMIX [24], despite its effectiveness, is constrained by its monotonous mixing network. Similarly, extensions like VMIX [25] face limitations in correlating individual rewards with contributions. In contrast, explicit methods, which directly attribute rewards, encounter challenges in dealing with complex agent interactions. For instance, COMA [27] relies on a biased baseline, and methods like SQDDPG [28] and SCC [29] are constrained by theoretical assumptions. Overcoming the shortcomings of both implicit and explicit methods, our approach leverages the powerful attention mechanism of

TABLE I: Comparison of different methods that utilize DT in RL. SA - Single-agent, MA - Multi-agent, SC - Safety Constraint ECA - Explicit Credit Assignment.

Method	Year	Domain	SC	ECA
DT (original) [14]	2021	—	N	N
ODT [26]	2022	SA	N	N
CDT [33]	2023	SA	Y	N
SaFormer [22]	2023	SA	Y	N
MADT [21]	2021	MA	N	N
SO2-MADT (ours)	2024	MA	Y	Y

transformers for explicit credit assignment, providing clear and direct feedback to optimize agent policies.

III. PRELIMINARIES

In this section, we delineate foundational concepts and knowledge by initially formulating MARL as a Markov Game. Next, we delve into the transformer architecture, emphasizing DT and its attention mechanisms. Finally, we discuss potential strategies for infusing safety mechanisms into the DT model, emphasizing the necessity of balancing performance goals with robust safety guarantees.

A. MARL and Markov Games

A common framework for addressing cooperative MARL challenges is through Markov games, which extend the single-agent Markov Decision Process (MDP) to a multi-agent setting. It can be defined using the tuple:

$$\mathcal{G} := \langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \{\mathcal{R}^i\}_{i\in\mathcal{N}}, P, \gamma \rangle$$

Here, $\mathcal{N} = [N]$ denotes the set of N agents; $\mathcal{S}$ is the state space shared by all agents; and $\mathcal{A}^i$ denotes the action space of agent $i \in \mathcal{N}$. The reward function $\mathcal{R}^i : S \times A^1 \times \ldots \times A^N \to \mathbb{R}$ captures the reward received by agent i, which is contingent upon the current state and the joint action of all agents. Furthermore, $P : S \times A^1 \times \ldots \times A^N \to \Delta(S)$ represents the state transition probability, mapping the current state and joint action to a distribution over the state space. Each agent strives to maximize its long-term reward $\sum_t \gamma^t r_t^i$, where $r_t^i \in R_i$ denotes the reward of agent i at time t, and $\gamma \in [0,1]$ serves as the discount factor.

B. Transformers Attention Mechanism

Transformers [12] have shown diverse applications in language processing [30] and vision tasks [31]. Given a sequence of input tokens $X = \{x_1, x_2, \ldots, x_n\}$, where $x_i \in \mathbb{R}^d$, a Transformer model, consisting of multiple layers, maps it to an output sequence of tokens $Z = \{z_1, z_2, \ldots, z_n\}$, where $z_i \in \mathbb{R}^d$. One of the most essential components in Transformer is the scaled dot-product attention, which captures the interrelationship of input sequences. In an attention mechanism, an input token representation X is linearly mapped into query, key, and value representations, i.e., $\{Q, K, V\} \in \mathbb{R}^{n\times d}$ respectively, which are learnable during training to compute self-attention as follows:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V. \tag{1}$$

C. Decision Transformer

The Decision Transformer (DT) [14] treats offline RL as a sequence modeling task. It trains autoregressive models on pre-collected offline data in a purely supervised manner, eliminating the requirement for computing cumulative rewards via dynamic programming. This enables DT to predict future actions based on desired returns, past states, and current actions. Leveraging a causally masked Transformer, DT can straightforwardly generate optimal actions.

Given a trajectory τ of length T, the reward return at timestep t is computed by $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$. DT utilizes three types of tokens: reward returns $R = \{R_1, \ldots, R_T\}$, states $s = \{s_1, \ldots, s_T\}$, and actions $a = \{a_1, \ldots, a_T\}$. The policy for DT can be trained by minimizing the loss between the predicted actions and the ground-truth actions in a sampled batch of data. Typically, DT uses the cross-entropy loss for discrete action spaces and the ℓ_2 loss for continuous action spaces. In [14], the authors discuss how the Return-to-Go (RTG) token affects the eventual return. However, in offline safe RL, directly manipulating RTG can compromise safety constraints. To address this, DT must incorporate both RTGs for performance and additional safety constraint tokens to ensure safe actions. Finding the optimal balance between these tokens is crucial for safe and effective policy learning.

IV. METHODOLOGY

In this section, we introduce SO2-MADT, a simple yet effective sequence model that tackles MARL challenges by combining safety constraints within the offline pre-training and online fine-tuning at the macro level using the STG token. While at the micro level, we implement explicit credit assignment using the Agent Prioritization Module. An overview of SO2-MADT is shown in Figure 2.

A. The STG Token

To address the challenges of offline safe RL, we introduce the concept of a Safety-To-Go (STG) token. The STG token represents the remaining allowable risk before violating safety thresholds. We formally define the STG token at time step t as $STG_t = \mathcal{V} - v_t$, where v_t is the accumulated safety violation up to that timestep, and $\mathcal{V}$ is the maximal safety limit for a trajectory.

Predicting STGs during training helps the model learn risk-aware representations, leading to proactive decision-making that prioritizes safety within defined limits. In contrast to the dynamic STG token, we define a constant safety limit $\mathcal{V}$ as the maximum allowable risk throughout a trajectory. It acts as a crucial safety constraint, directing agent actions to ensure that the cumulative risk during an episode remains within predefined safety limits $\mathcal{V}$. The STG token is embedded and concatenated with the state and action embeddings at each timestep to form the augmented input $\mathcal{X}_{\text{aug},t}$ for DT. This augmented input plays a crucial role in updating the DT parameters, enabling the agent to make decisions within safe operational bounds. The procedure for inserting the STG token is shown in Algorithm 1.

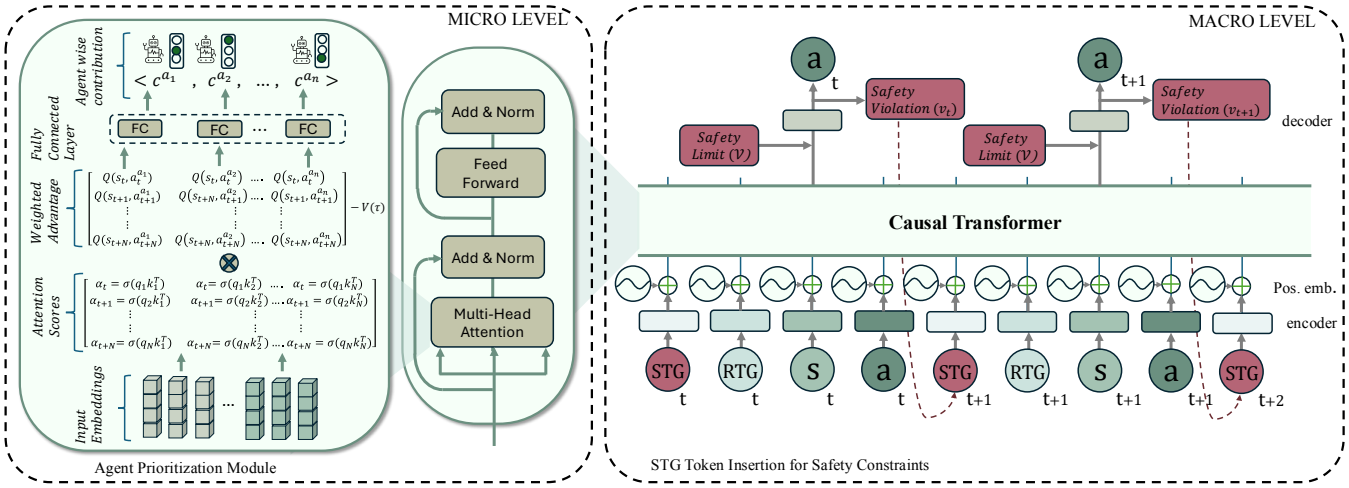

Fig. 2: Overview of SO2-MADT with STG Token for Safety Constraints at the macro level and Agent Prioritization Module at the micro level

Algorithm 1 STG Token Integration for DT

Input: Batch of offline trajectories $\{\tau_i\}$, Maximum safety limit $\mathcal{V}$
for each trajectory τ in $\{\tau_i\}$ **do**
  for each timestep t in τ **do**
    Calculate Safety Violation: $v_t = \sum_{k=1}^{t} v_k$
    Compute STG: $STG_t = \mathcal{V} - v_t$
    Embed STG Token: $e_{STG_t} = \text{Embed}(STG_t)$
    Retrieve RTG and Embeddings for State and Action:
    $e_{RTG_t} = \text{Embed}(RTG_t)$
    $e_{s_t} = \text{Embed}(s_t)$
    $e_{a_t} = \text{Embed}(a_t)$
    Augment Input: $\mathcal{X}_{\text{aug},t} = [e_{s_t}, e_{a_t}, e_{RTG_t}, e_{STG_t}]$
    Update Parameters: Use $\mathcal{X}_{\text{aug},t}$ to update DT parameters
  end for
end for

B. Agent Prioritization Module

As discussed in Section II-C, discerning the individual contribution of each agent is pivotal for effective learning and coordination. To address this challenge, our model implements an explicit credit assignment method by utilizing the attention mechanism. Following the CTDE approach, we incorporate a centralized critic within each agent's network to enhance learning. By combining these two designs, we introduce the Agent Prioritization Module (APM), which dynamically identifies the most influential agents and actions to maximize overall performance. Operating at the micro level, this module facilitates granular performance improvements within the multi-agent system and seamlessly integrates into both the offline and online phases of our model.

1) Offline Pre-training: During offline pre-training, we prioritize learning from actions that have the most significant impact on outcomes within the historical trajectories. To accomplish this, we integrate the attention mechanism into the calculation of our advantage function A, as shown in Algorithm 2. This mechanism dynamically identifies past actions that contribute significantly to achieving long-term rewards. We compute the attention weights α_t by applying a softmax function (σ) to the dot product scores obtained from the query and key representations of the agents' actions and states, thereby normalizing these scores across all actions in the trajectory. We then formulate our attention-weighted advantage function for influential actions as follows:

$$A(\tau, a_t) = \alpha_t * (Q(s_t, a_t) - V(\tau)) \tag{2}$$

where τ represents the trajectory of past states, actions, and rewards; α_t is the attention weight for action a_t; $Q(s_t, a_t)$ is the Q-value representing the expected future rewards; and $V(\tau)$ is the attention weighted value function.

At timestep t, the attention mechanism assigns weights α_t to each agent a_n's action a_t, reflecting its relative importance within the trajectory. These attention weights are then used for calculating the contribution value c for the n^{th} agent a:

$$c^{a_n} = \sum_{t=1}^{N} A(\tau, a_t^{a_n}) \tag{3}$$

Here $A(\tau, a_t^{a_n})$ represents the attention-weighted advantage function, which quantifies the impact of agent a_n's action a_t in state s_t relative to the overall trajectory value $V(\tau)$, accumulated over all time steps from $t = 1$ to N. We employ contribution values c^{a_n} within the APM to augment the decision-making process. The APM utilizes contribution values, represented by c^{a_n} scores, to rank agents and influence the decision-making process. Agents with higher contribution scores, indicate a stronger impact on team outcomes and are prioritized in the action hierarchy. Agents with lower scores take action later in the sequence, ensuring that all agents remain actively engaged in the decision-making process. This prioritization shapes the collective strategy, favoring actions from agents that contribute most to team success, thus enhancing collaborative performance.

To effectively balance the exploration-exploitation trade-off and prioritize actions based on each agent's contribution, we utilize cross-entropy (CE) loss as follows:

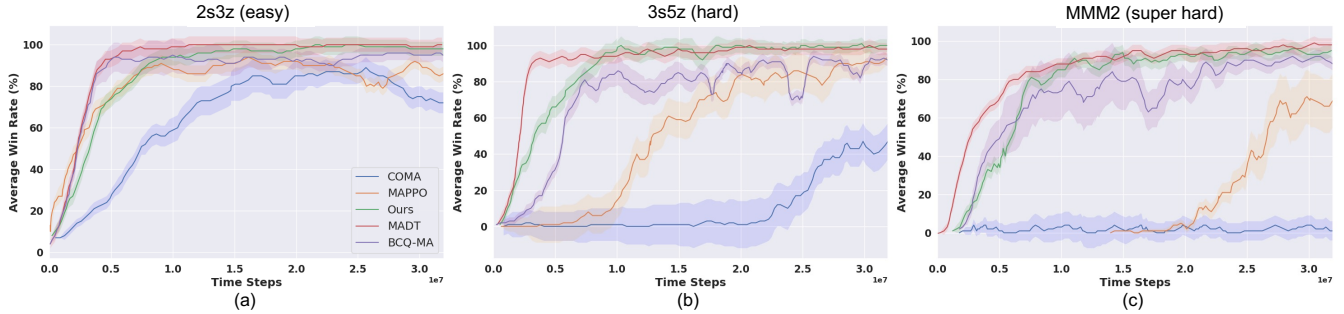

Fig. 3: Performance of SO2-MAD on the SMAC dataset compared to baselines on easy, hard, and super-hard maps. The x-axis is the number of timesteps during training. The y-axis represents the average test win rate in percentage.

Algorithm 2 Offline Pre-training

Input: Offline dataset $\mathcal{D}$
Initialize: Parameters θ for the Decision Transformer and n number of agents, $\mathcal{V}$ as the safety threshold.
for each trajectory τ in $\mathcal{D}$ **do**
  Integrate STG tokens into input sequence for explicit safety.
  Compute safety function v_t
  for each timestep t in τ **do**
    Compute advantage:
      $A(\tau, a_t) = \alpha_t \cdot (Q(s_t, a_t) - V(\tau))$
    Assign attention weights to each agent $a_n{}'s$ action
    Compute contribution value:
      $c^{a_n} = \sum_{t=1}^{N} A(\tau, a_t^{a_n})$
  end for
  Predict next action a_{t+1}
  Update θ using cross-entropy loss $L_{\text{CE}}(\theta)$
end for
return θ

Algorithm 3 Online Fine-tuning

Input: Offline dataloader D, Pretrained policy with parameters θ
Initialize: Actor π_θ, critic V_ϕ, and safety threshold $\mathcal{V}$
for each episode received from the environment **do**
  Initialize STG for the episode as STG
  for each timestep t **do**
    Execute a_t from π_θ,
    observe r_t, s_{t+1}, and safety violation v_t
    Update STG based on v_t and STG_t
    Calculate advantage $A(\tau, a_t^a)$
    Assign attention weight α_t^a
    Update θ using advantage and attention-weight:
      $\theta \leftarrow \theta + \eta \nabla_\theta \log \pi_\theta(a_t^{a_n} \mid s_t) A(\tau, a_t^{a_n})$
    Minimize MSE loss L_ϕ to update ϕ, ensuring safety:
      $L_\phi = \frac{1}{2} \sum_t (R_t - V_\phi(s_t, STG_t))^2$
  end for
end for
return θ, ϕ

$$L_{\text{CE}}(\theta) = \frac{1}{C_L} \left[\sum_{t=1}^{C_L} P(a_t) \log P(\hat{a}_t | \tau_t; \theta) + \lambda \sum_{i=1}^{N} \max(0, \mathcal{V} - v_i) \right] \quad (4)$$

where C_L represents the context length, a_t and $\hat{a}_t$ denote the ground truth action and predicted action respectively; λ is the regularization parameter, while $\max(0, \mathcal{V} - v_i)$ represents the penalty. This CE Loss guides the model toward accurate predictions and enforces predefined safety constraints.

2) Online Fine-tuning: Transitioning from offline training to online fine-tuning often leads to performance degradation. To combat this, our approach focuses on improvements at the micro level during the online phase. At the onset of online fine-tuning, we initialize actors (π_θ) and critics (V_ϕ) for each agent. The actors are responsible for decision-making, while the critics evaluate the collective consequences of these decisions, providing a holistic view of the environment. As the agents interact with the environment, any instances of safety violations v_t are meticulously recorded and utilized to update the STG token. This iterative adjustment serves as a crucial feedback loop for each episode, allowing the model to re-calibrate its actions in real time.

The core of the online fine-tuning process involves updating the policy network. We employ our stable and efficient PPO algorithm, outlined in Algorithm 3. The policy network π is updated through a gradient ascent step to maximize the expected rewards, attentively weighted by the past impact of actions. The mathematical expression for the policy update is as follows:

$$\theta \leftarrow \theta + \eta \nabla_\theta \sum_{t=1}^{N} \alpha_t^{\log \pi_\theta(a_t^{a_n} | s_t) A(\tau, a_t^{a_n})} \quad (5)$$

Here, η represents the learning rate; ∇_θ denotes the gradient to the policy parameters θ; α_t is the attention weight for agent a_n's action a_t at time t; and $A(\tau, a_t^{a_n})$ is the advantage function, evaluating the quality of the action taken in the given state. The MSE loss function enhances the agent's policy by selecting actions that not only maximize expected rewards but also adhere to safety constraints.

V. Experiments and Results

To evaluate the proposed SO2-MADT, we conduct experiments on two widely used MARL benchmark datasets: the StarCraftII Multi-agent Challenge (SMAC) [16] and the Multi-agent MuJoCo (MAMuJoCo) [15]. The proposed model is implemented with the Pytorch 1.5.0 and tested on 4 NVIDIA Tesla P100 GPUs. We use the original hyperparameters from baseline publications to ensure a fair comparison.

A. StarCraftII Multi-agent Challenge (SMAC) Dataset

The SMAC is a benchmark designed specifically to evaluate cooperative MARL algorithms. Leveraging the intricate dynamics of the real-time strategy game StarCraft II,

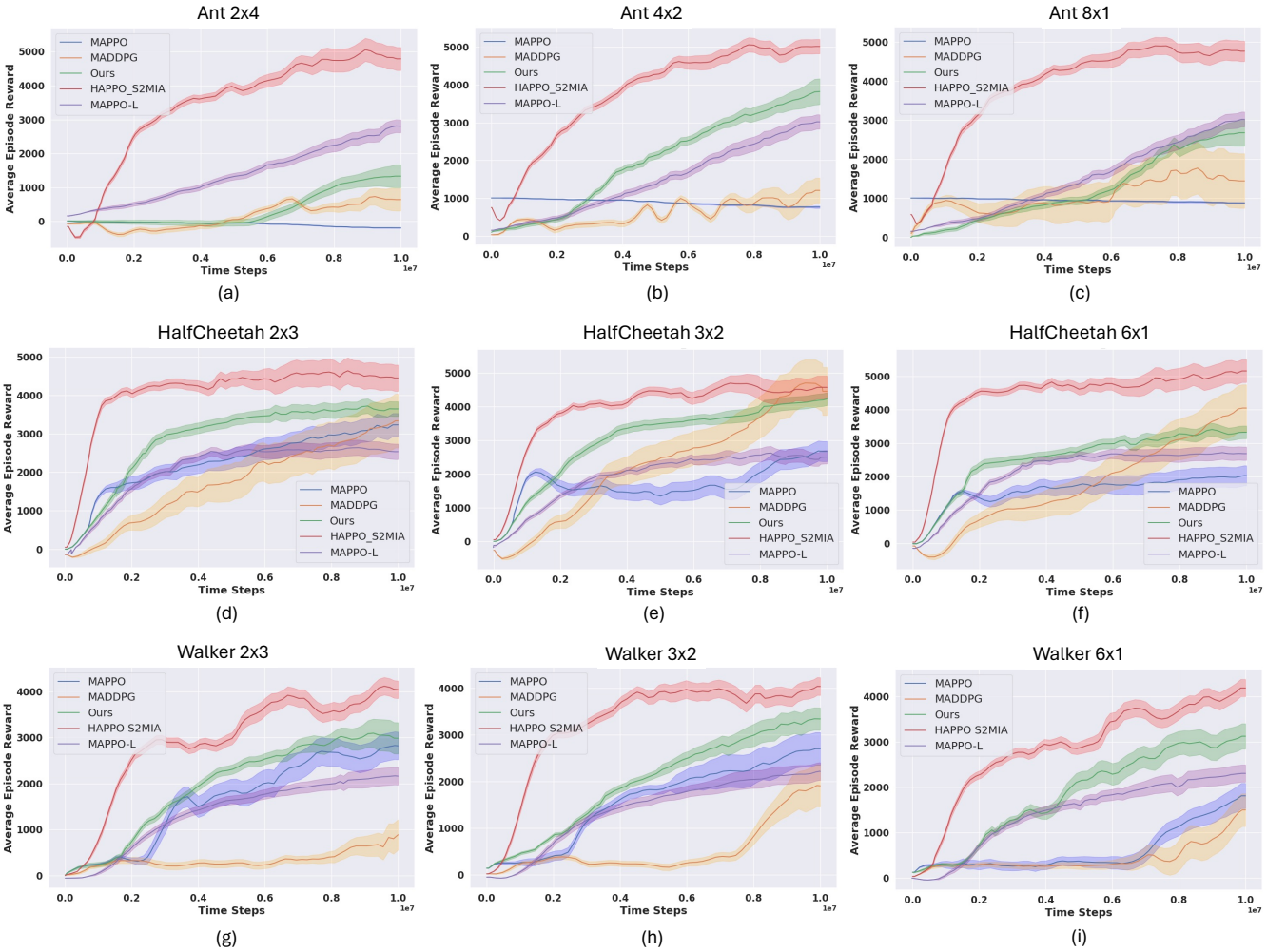

Fig. 4: Mean performance of SO2-MADT on Mutli-agent MuJoCo dataset. The shaded region shows the interquartile range. The x-axis is the number of environmental time steps while The y-axis represents average episode rewards.

SMAC comprises a series of maps, tailored to rigorously test MARL algorithms. Within SMAC, a MARL algorithm commands one group of units, while a built-in heuristic AI controls the opposing group. The scenarios included in SMAC are configured with diverse initial unit placements, types, numbers, and terrain obstacles, in which each agent is endowed with a set of actions. Additionally, a shared reward system is implemented, where the reward of each agent is calculated based on the cumulative damage dealt weighed against the damage received. In the SMAC benchmark, safety constraints require agents to avoid friendly fire, manage resources effectively such as focus fire, and maintain safe positions on the battlefield to prevent unnecessary losses. For comparative performance analysis, we conduct experiments with MADT [21] an offline MARL approach pre-trained and integrated with DT for online evaluation, COMA [27] a counterfactual credit assignment method for multi-agent cooperation, MAPPO [38] the pioneering approach applying PPO directly in MARL, and BCQ-MA [39] a multi-agent extension of the Batch Constrained Q-learning algorithm.

B. Multi-agent MuJoCo Dataset

We selected the MAMuJoCo benchmark suite to assess our method in complex multi-agent control environments thoroughly. MAMuJoCo's tasks involve controlling distinct components of simulated robots, underscoring the necessity for tailored policies for each agent's role. Our method, which combines sequential modeling with explicit credit assignment, is well-suited for this challenge. The attention-based credit assignment mechanism precisely identifies influential actions, enabling the isolation of individual reward signals for accurate assessment, and ensuring focused policy updates. In the MAMuJoCo benchmark, safety constraints include unsafe states and actions, such as agent-specific velocity thresholds and coordinated joint movement to avoid collisions. For experiments on MAMuJoCo, we compare our approach with MAPPO [38]; the same baseline as in SMAC, MAPPO-L [40] which incorporates Lagrangian-based constraints to promote adherence to specified limits, and MADDPG [41] an approach employing a centralized critic with decentralized actor-learners to enhance learning stability. Additionally, we include HAPPO-S2MIA [42], the

current state-of-the-art method based on the HAPPO algorithm, showcasing superior performance in MARL environments using the MAMuJoCo dataset.

C. Results

We evaluated our model on the SMAC dataset across four different battle maps: an easy map (2s3z), a hard map (3s5z, 2c_vs_64zg), and a super-hard map (MMM2). The agents underwent training for 30 million environmental steps, and the results are depicted in Figure 3. Our approach consistently demonstrated robust performance across various difficulty levels, ranging from easy to super-hard maps. Our method showcased superior performance compared to established algorithms such as COMA, MAPPO, and BCQ-L in the hard and super-hard categories, while remaining comparable to MADT. Unlike MAPPO, which relies on parameter-sharing as a primary mechanism, our approach achieves these results by leveraging stringent safety constraints while still maintaining or surpassing comparative performance levels. It is noteworthy that our safety-constrained model initially exhibits a slower learning curve, particularly in 'hard' and 'super hard' scenarios. This cautious approach arises from its careful navigation of safety constraints and prioritization of safe strategies. However, as training progresses, our explicit credit assignment mechanism continually enhances decision-making, resulting in a significant performance boost and higher win rates compared to alternative approaches.

On the MAMuJoCo dataset, we conducted extensive testing across nine tasks within the Walker, HalfCheetah, and Ant scenarios. Agents undergo training for 10 million environmental steps, with the results illustrated in Figure 4. The plots demonstrate that our model outperforms nearly all state-of-the-art (SOTA) works except for HAPPO-S2MIA. This discrepancy primarily arises from HAPPO-S2MIA's absence of explicit safety constraints, which enables more aggressive early learning. Conversely, our safety-constrained model demonstrates a slower initial learning phase. However, as learning progresses, our approach achieves superior average episode rewards compared to most state-of-the-art (SOTA) methods, closely approaching the performance of HAPPO-S2MIA. Particularly noteworthy is our model's surpassing performance in the Ant 8x1 and Walker 6x1 scenarios, presumed to be more complex due to the increased dimensions of the task. Additionally, we observed that the inclusion of safety constraints does not impede the model from achieving high average episode rewards. This is evident in scenarios such as Walker 3x2 and HalfCheetah 2x3, where our model performs comparably with HAPPO-S2MIA.

Overall, our SO2-MADT demonstrates a strong ability to balance safety considerations with the capacity to learn high-reward policies effectively. The results highlight our model's robust generalization across various scenarios in the dataset.

VI. ABLATION STUDY

We conduct ablation studies to assess the effectiveness of safety constraints (SC) and explicit credit assignment (ECA) in our model. We begin by pre-training our model on a 5m_vs_6m (hard map) and then fine-tuning it online on the same map. Figure 5 demonstrates that our model achieves the best performance when incorporating both safety constraints and explicit credit assignment. Conversely, the absence of either safety constraints or explicit credit assignment results in inferior performance and unstable convergence, validating the necessity of each component.

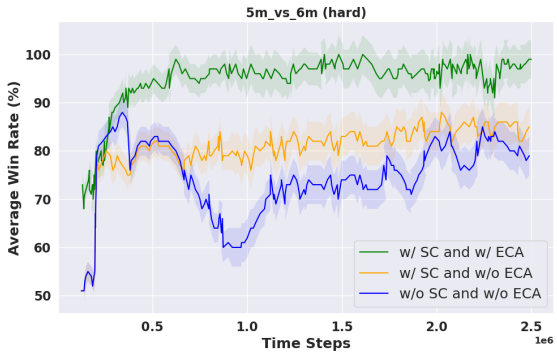

Fig. 5: Ablation results on *5m_vs_6m* hard map for demonstrating the effectiveness of SC and ECA in MARL.

VII. LIMITATIONS

While SO2-MADT represents a significant advancement in safety-constrained MARL, several areas warrant further exploration to enhance its application and efficacy. Firstly, more experiments are needed to assess SO2-MADT's scalability, especially in scenarios with an exceptionally large number of agents. Secondly, despite its proven practicality, additional experiments would help validate SO2-MADT's adaptability to extreme environments. Furthermore, conducting tests in the absence of direct inter-agent interaction may provide additional validation in challenging communication environments. Addressing these uncommon challenges can strengthen the robustness and applicability of SO2-MADT, leading to a more comprehensive and effective methodology.

VIII. CONCLUSIONS

SO2-MADT significantly advances safety-constrained multi-agent reinforcement learning (MARL), establishing a new standard by integrating safety constraints at both macro and micro levels, along with a robust explicit credit assignment mechanism. This innovative approach not only ensures that agents operate within predefined safety parameters but also enhances their overall performance. The implications of this research are profound, particularly in safety-critical domains such as autonomous vehicle navigation and advanced robotics. Moreover, the proposed SO2-MADT can handle future complex scenarios by incorporating novel context-aware safety constraints and diverse agent architectures, demonstrating great extensibility.

ACKNOWLEDGMENT

This research is partially supported by NSF grants CCF-2130688, and CNS-2107057.

REFERENCES

[1] Y. Yang and J. Wang, "An overview of multi-agent reinforcement learning from game theoretical perspective," arXiv preprint arXiv:2011.00583, 2020.

[2] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," in IEEE Transactions on Industrial Informatics, vol. 9, no. 1, pp. 427-438, 2012.

[3] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, et al., "Dota 2 with large scale deep reinforcement learning," arXiv preprint arXiv:1912.06680, 2019.

[4] J. Orr and A. Dutta, "Multi-agent deep reinforcement learning for multi-robot applications: a survey," in Sensors, vol. 23, no. 7, pp. 3625, 2023.

[5] K. Ren and J. Yu, "Research status of bionic amphibious robots: A review," in Ocean Engineering, vol. 227, 108862, 2021.

[6] M. Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Pérolat, D. Silver, and T. Graepel, "A unified game-theoretic approach to multiagent reinforcement learning," in Advances in Neural Information Processing Systems 30, 2017.

[7] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey," in Artificial Intelligence Review, pp. 1-49, 2022.

[8] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers, vol. 16, pp. 66-83, Springer International Publishing, 2017.

[9] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in Advances in Neural Information Processing Systems 29, 2016.

[10] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, OpenAI P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in Advances in Neural Information Processing Systems 30, 2017.

[11] Y.-H. Chang, T. Ho, and L. Kaelbling, "All learning is local: Multi-agent learning in global reward games," in Advances in Neural Information Processing Systems 16, 2003.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems 30, 2017.

[13] J. Shang, K. Kahatapitiya, X. Li, and M. S. Ryoo, "Starformer: Transformer with state-action-reward representations for visual reinforcement learning," in European Conference on Computer Vision, pp. 462-479, Cham: Springer Nature Switzerland, 2022.

[14] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," in Advances in Neural Information Processing Systems 34, pp. 15084-15097, 2021.

[15] B. Peng, T. Rashid, C. Schroeder de Witt, P.-A. Kamienny, P. Torr, W. Böhmer, and S. Whiteson, "Facmac: Factored multi-agent centralised policy gradients," in Advances in Neural Information Processing Systems 34, pp. 12208-12221, 2021.

[16] Samvelyan, Mikayel, 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).

[17] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in International Conference on Machine Learning, PMLR, 2019, pp. 2052–2062.

[18] A. Kumar, J. Hong, A. Singh, and S. Levine, "When should we prefer offline reinforcement learning over behavioral cloning?" arXiv preprint arXiv:2204.05618, 2022.

[19] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," arXiv preprint arXiv:2006.04779, 2020.

[20] M. Janner, Q. Li, and S. Levine, "Offline reinforcement learning as one big sequence modeling problem," in Advances in Neural Information Processing Systems (NeurIPS), 2021.

[21] L. Meng, M. Wen, Y. Yang, C. Le, X. Li, W. Zhang, Y. Wen, H. Zhang, J. Wang, and B. Xu, "Offline pre-trained multi-agent decision transformer: One big sequence model tackles all smac tasks," arXiv preprint arXiv:2112.02845, 2021.

[22] Q. Zhang, L. Zhang, H. Xu, L. Shen, B. Wang, Y. Chang, X. Wang, B. Yuan, and D. Tao, "Saformer: A conditional sequence modeling approach to offline safe reinforcement learning," arXiv preprint arXiv:2301.12203, 2023.

[23] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, et al., "Value-decomposition networks for cooperative multi-agent learning based on team reward," in International Conference on Autonomous Agents and Multi-Agent Systems, pp. 2085–2087, 2018.

[24] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in International Conference on Machine Learning, pp. 4295–4304, 2018.

[25] J. Su, S. Adams, and P. Beling, "Value-decomposition multiagent actor-critics," in AAAI Conference on Artificial Intelligence, pp. 11352–11360, 2021.

[26] Q. Zheng, A. Zhang, and A. Grover, "Online decision transformer," in Proc. International Conference on Machine Learning, PMLR, 2022, pp. 27042-27059.

[27] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in AAAI Conference on Artificial Intelligence, pp. 2974–2982, 2018.

[28] J. Wang, Y. Zhang, T.-K. Kim, and Y. Gu, "Shapley Q-value: A local reward approach to solve global reward games," in AAAI Conference on Artificial Intelligence, pp. 7285–7292, 2020.

[29] J. Li, K. Kuang, B. Wang, F. Liu, L. Chen, F. Wu, and J. Xiao, "Shapley counterfactual credits for multi-agent reinforcement learning," in ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 934–942, 2021.

[30] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2019.

[31] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012-10022, 2021.

[32] W. Li, W. Liu, S. Shao, S. Huang, and A. Song, "Attention-based Intrinsic Reward Mixing Network for Credit Assignment in Multi-Agent Reinforcement Learning," in *IEEE Transactions on Games*, 2023.

[33] Z. Liu, Z. Guo, Y. Yao, Z. Cen, W. Yu, T. Zhang, and D. Zhao, "Constrained decision transformer for offline safe reinforcement learning," arXiv preprint arXiv:2302.07351, 2023.

[34] A. Sootla, A. I. Cowen-Rivers, T. Jafferjee, Z. Wang, D. H. Mguni, J. Wang, and H. Ammar, "Sauté RL: Almost surely safe reinforcement learning using state augmentation," in International Conference on Machine Learning, pp. 20423-20443, PMLR, 2022.

[35] A. Stooke, J. Achiam, and P. Abbeel, "Responsive safety in reinforcement learning by PID Lagrangian methods," in International Conference on Machine Learning, pp. 9133–9143, PMLR, 2020.

[36] Y. Chen, J. Dong, and Z. Wang, "A primal-dual approach to constrained Markov decision processes," arXiv preprint arXiv:2101.10895, 2021.

[37] W. Zhao, T. He, and C. Liu, "Model-free safe control for zero-violation reinforcement learning," in 5th Annual Conference on Robot Learning, 2021.

[38] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu, "The Surprising Effectiveness of MAPPO in Cooperative, Multi-Agent Games," ArXiv, abs/2103.01955, 2021.

[39] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in Proc. International Conference on Machine Learning, 2019, pp. 2052-2062.

[40] S. Gu, J. G. Kuba, M. Wen, R. Chen, Z. Wang, Z. Tian, J. Wang, A. Knoll, and Y. Yang, "Multi-agent constrained policy optimisation," arXiv preprint arXiv:2110.02793, 2021.

[41] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, OpenAI P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in Advances in Neural Information Processing Systems 30, 2017.

[42] M. Sun, Y. Hou, J. Kang, H. Piao, Y. Zeng, H. Ge, and Q. Zhang, "Improving Cooperative Multi-Agent Exploration via Surprise Minimization and Social Influence Maximization," in Proc. of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23), International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2023, pp. 2607-2609.