ENHANCING THE HAPPO FRAMEWORK WITH RGA

ZHEYU LI [LIZHEYU@SEAS], ZENG LI [LIZENG@SEAS]

HAICHAO ZHAO [HAICHAO@SEAS], YIWEI TANG [TGG123@SEAS]

ABSTRACT. This work enhances the Heterogeneous-Agent Proximal Policy Optimization (HAPPO) framework by integrating a Relation-Aware Global Attention (RGA) module that captures pairwise spatial correlations among agents. The proposed approach improves global contextual awareness and inter-agent coordination. Experimental comparisons with baseline methods demonstrate increased stability, precision, and adaptability in multi-agent reinforcement learning (MARL) environments.

1. INTRODUCTION

Multi–agent reinforcement learning (MARL) has emerged as a powerful framework for solving complex coordination and competition tasks in environments where multiple decision–making entities interact. In particular, on–policy algorithms such as Proximal Policy Optimization (PPO) have been extended to heterogeneous–agent settings (HAPPO) to allow each agent to learn its own policy while accounting for the evolving strategies of its peers. Although HAPPO achieves strong sample efficiency and stability through clipped surrogate objectives, it treats each agent's observation as an independent input and does not explicitly exploit the rich structure present in the interactions among agents. As a result, policies learned by HAPPO may fail to capture global relational patterns that are crucial in settings where the behavior of one agent depends heavily on the collective state of the group.

Attention mechanisms have been widely adopted in computer vision and natural language processing to selectively highlight informative elements of high–dimensional data. In the context of MARL, however, naive attention modules typically operate on local features or fully connected layers, which either neglect long–range dependencies or incur prohibitive computational cost as the number of agents grows. To bridge this gap, Relation–Aware Global Attention (RGA) was recently introduced in the person re–identification literature to mine global structural affinities via compact relation vectors and lightweight convolutional layers. By stacking pairwise correlations and "mining" clustering–like patterns, RGA can effectively infer which features are most salient in a global context, without resorting to expensive non–local operations or deep stacks of convolutions.

In this work, we propose *HAPPO–RGA*, a novel integration of heterogeneous–agent PPO with relation–aware global attention. Our approach injects two cascaded RGA modules (spatial and channel) into both actor and critic networks, allowing each agent to modulate its feature representations based on learned affinities with every other agent. We demonstrate that the RGA–enhanced feature extractor yields a compact embedding which, when followed by standard fully connected heads, leads to more discriminative policies and value estimates. Empirically, HAPPO–RGA outperforms baseline HAPPO in a variety of cooperative and competitive benchmarks, illustrating the benefit of explicitly modeling inter–agent structure.

2. BACKGROUND

Multi–agent reinforcement learning (MARL) enables multiple agents to learn policies through interaction, but scalability and non–stationarity remain major challenges. On–policy methods like Proximal Policy Optimization (PPO) offer stable updates via clipped objectives, and Heterogeneous–Agent PPO (HAPPO) extends this to agents with different observations and actions. However, HAPPO's standard networks process each agent's input independently and overlook the global relationships among agents.

In scenarios such as multi–robot coordination or resource allocation, agents' decisions hinge on both their own states and their interactions with peers. Simple attention or non–local blocks can capture pairwise relations but suffer from high complexity or rigid, position–invariant weights.

Relation–Aware Global Attention (RGA) addresses these issues by compactly encoding all pairwise affinities into relation vectors and learning adaptive attention masks with lightweight convolutions. Integrating RGA into HAPPO promises to inject global structural awareness into both actor and critic, improving coordination and value estimation in heterogeneous–agent settings.

3. Related Work

3.1. **HAPPO.** Proximal Policy Optimization (PPO) **[3]** is a widely used on-policy reinforcement learning algorithm that provides stable and sample-efficient updates via a clipped surrogate objective. It has been successfully applied to single-agent and multi-agent scenarios, but assumes shared policy parameters across agents. MAPPO addresses this limitation by adopting a centralized critic for multiple agents while keeping parameter sharing across agents. However, in heterogeneous-agent environments, parameter sharing can degrade performance due to varying observation and action spaces.

HAPPO, proposed in the HARL framework **[1]**, generalizes PPO to heterogeneous multi-agent systems by introducing a **sequential update scheme**, where each agent updates its policy while keeping others fixed. This strategy reduces non-stationarity and improves optimization stability. HAPPO is theoretically supported by the **multiagent advantage decomposition lemma**, which ensures monotonic joint policy improvement and convergence to a Nash equilibrium. Moreover, HAPPO replaces the trust region constraint with a clipped objective similar to PPO, preserving computational efficiency.

Built upon this solid foundation, our work proposes RGA-HAPPO, which integrates a **Relation-Aware Global Attention (RGA)** module into the actor network to enhance global context perception. By modeling pairwise spatial correlations among agents, RGA-HAPPO improves inter-agent coordination and decision making, leading to improved performance in complex MARL tasks.

3.2. **RGA.** The Relation-Aware Global Attention (RGA) module was originally proposed to improve person re-identification by explicitly modeling global structural dependencies among spatial and channel-wise features [2]. Traditional attention mechanisms often rely on local convolutions or fixed-size receptive fields, limiting their ability to extract meaningful patterns from globally distributed relationships. In contrast, RGA computes pairwise affinities between all feature positions and stacks them to form a compact relation vector that captures clustering-like structural patterns in both spatial and semantic spaces.

The RGA module comprises two branches: spatial RGA (RGA-S) and channel RGA (RGA-C). Each feature node's importance is inferred by jointly considering its original feature and its relation vector through shared convolutional operations. This design not only improves discriminative feature extraction but also supports efficient parameter sharing and scale invariance. Unlike non-local attention, which applies deterministic weighted feature aggregation, RGA treats relations as learnable representations for mining semantics via trainable transformations.

In our work, we adapt RGA to the MARL setting by integrating it into the actor network of HAPPO. The RGA-enhanced actor leveRGAes pairwise spatial relations among agents to improve inter-agent awareness, coordination, and policy expressiveness, especially in decentralized, partially observable environments.

4. Approach

In this section, we describe the architectural integration of the Relation-Aware Global Attention (RGA) module into the Heterogeneous-Agent Proximal Policy Optimization (HAPPO) framework, resulting in the RGA-HAPPO algorithm. The key goal of this design is to enhance the actor network's ability to model inter-agent relations via both spatial and channel attention mechanisms.

4.1. **HAPPO Update Mechanism.** HAPPO performs policy optimization via a sequential update scheme that mitigates non-stationarity by fixing the policies of all other agents when updating a target agent. The training is guided by the **Multi-Agent Advantage Decomposition Lemma**, which expresses the joint advantage function as the sum of conditionally independent advantage terms:

$$A^{1:3}(s, \boldsymbol{a}^{1:3}) = A^1(s, a^1) + A^2(s, a^1, a^2) + A^3(s, a^{1:2}, a^3) \tag{1}$$

This decomposition allows each agent to compute its advantage while conditioning on previously updated actions, ensuring monotonic improvement in joint return.

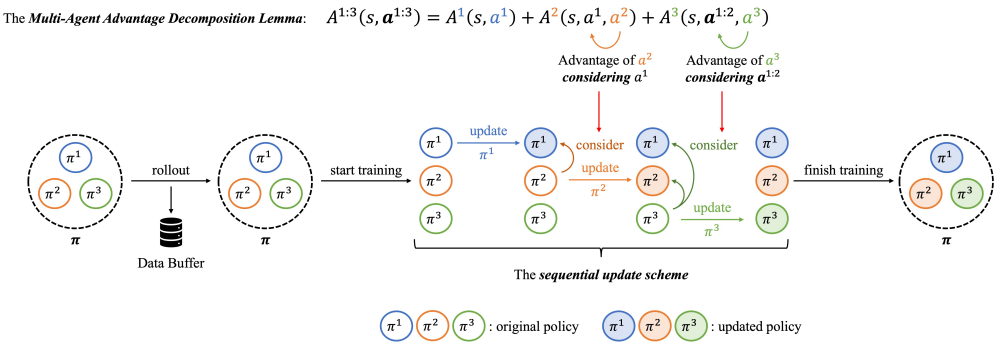

FIGURE 1. Illustration of the Multi-Agent Advantage Decomposition Lemma and HAPPO's sequential policy update scheme. Each agent optimizes its policy while considering previously updated agents, reducing non-stationarity. Figure adapted from **[1]**.

4.2. **Relation-Aware Global Attention Integration.** To improve HAPPO's capacity for capturing inter-agent dependencies, we introduce RGA modules into both the actor and critic networks. RGA modules consist of two components: spatial RGA (RGA-S) and channel RGA (RGA-C), which operate sequentially.

Given intermediate feature maps $X \in \mathbb{R}^{C\times H\times W}$, RGA-S first treats each spatial location i as a node and computes a relation vector $r_{\mathrm{s},i} \in \mathbb{R}^{2N}$, stacking the row and column affinities:

$$r_{\mathrm{s},i} = [X_{i,:}, X_{:,i}] \tag{2}$$

Each relation vector is then embedded and combined with the original feature using a learned attention mask M_i:

$$u_i = \mathrm{ReLU}(W_x x_i + b_x), \tag{3}$$

$$v_i = \mathrm{ReLU}(W_r r_{\mathrm{s},i} + b_r), \tag{4}$$

$$y_i = [u_i, v_i], \tag{5}$$

$$M_i = \sigma(W_2 \mathrm{ReLU}(W_1 y_i + b_1) + b_2), \tag{6}$$

$$x_i' = M_i \odot x_i \tag{7}$$

The output $X^{(1)}$ is then processed by RGA-C to model cross-channel relations:

$$r_{\mathrm{c},c} = [X^{(1)}_{c,:}, X^{(1)}_{:,c}] \in \mathbb{R}^{2C} \tag{8}$$

which undergoes similar embedding and gating to yield the refined representation $X^{(2)}$.

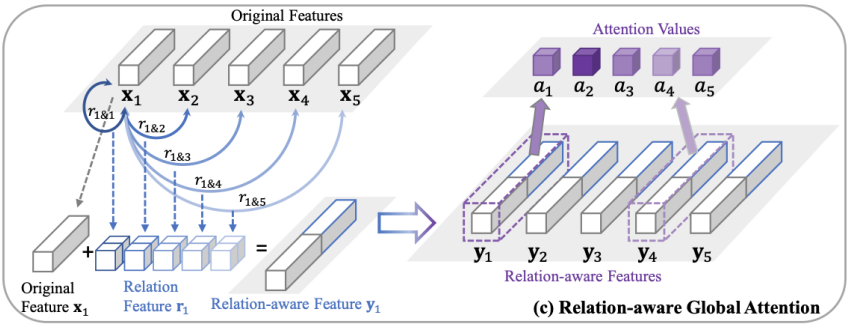

FIGURE 2. Relation-Aware Global Attention (RGA) module. Each feature vector computes relation-aware representations via pairwise correlations, which are modulated with learned attention masks to emphasize salient features. Figure adapted from **[2]**.

After applying global average pooling to obtain $h \in \mathbb{R}^d$, the actor computes its output through fully connected layers and produces the final action distribution:

$$\pi(a \mid s) = \mathrm{softmax}(W^{(3)} \mathrm{ReLU}(W^{(2)} \mathrm{ReLU}(W^{(1)} h + b^{(1)}) + b^{(2)}) + b^{(3)}) \tag{9}$$

The critic shares the same RGA feature extractor and computes the state value via:

$$z_v = \mathrm{ReLU}(V^{(1)}h + c^{(1)}), \tag{10}$$

$$V(s) = V^{(2)}z_v + c^{(2)} \tag{11}$$

The actor is optimized with PPO's clipped surrogate objective:

$$L_{\text{actor}} = -\mathbb{E}_t\left[\min\left(r_t\hat{A}_t, \text{clip}(r_t, 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \quad r_t = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\text{old}}(a_t \mid s_t)} \tag{12}$$

while the critic minimizes the MSE loss:

$$L_{\text{critic}} = \mathbb{E}_t[(V(s_t) - \hat{R}_t)^2] \tag{13}$$

These gradients are jointly backpropagated through the RGA modules and the network, enabling end-to-end optimization.

Algorithm 1 HAPPO with RGA (HAPPO-RGA)

Initialize actor parameters θ and critic parameters ϕ, including all RGA modules
for each iteration **do**
  Collect on-policy trajectories $\{s_t, a_t, r_t, s_{t+1}\}$ by sampling $a_t \sim \pi_\theta(\cdot \mid s_t)$
  Compute advantages $\hat{A}_t$ via GAE and returns $\hat{R}_t$
  for each agent i **do**
    Compute policy ratio $r_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}$
    Actor loss

$$L_{\text{actor}} = -\,\mathbb{E}_t\Big[\min\big(r_t\hat{A}_t,\ \text{clip}(r_t, 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\Big]$$

    Critic loss

$$L_{\text{critic}} = \mathbb{E}_t\big[\big(V_\phi(s_t) - \hat{R}_t\big)^2\big]$$

    Update θ and ϕ by descending $\nabla_{\theta,\phi}\big(L_{\text{actor}} + c\,L_{\text{critic}}\big)$
  end for
end for

5. Experimental Results

We evaluate our proposed **HAPPO-RGA** algorithm against state-of-the-art multi-agent baselines, including **MAPPO** and **HAPPO**, on two benchmarks: the **Multi-Agent Particle Environment (MPE)** and the more complex **Google Research Football (GRF)** scenarios. All models follow the **Centralized Training with Decentralized Execution (CTDE)** paradigm.

5.1. **Environment Setup.** The primary benchmark is the **MPE Spread task**, where N agents must cover N distinct landmarks while avoiding collisions. We adopt the standardized PettingZoo interface to ensure consistency across all evaluations. Following prior work, we modify the reward scheme so that agents receive only a joint reward, i.e., the sum of individual agent rewards, reinforcing collective behavior. All episodes are normalized to a fixed length for training stability.

We additionally test on **GRF Counterattack Easy** and **GRF Counterattack Hard**, which introduce adversarial dynamics, long-horizon coordination, and partial observability. These tasks demand more complex strategic behavior and serve as a test of generalization.

5.2. **Training Performance on MPE Spread.** Figure 3 presents the average episode return during training. MAPPO lags due to shared parameter updates and poor credit assignment in heterogeneous settings. HAPPO improves upon this by updating agents sequentially. Our **HAPPO-RGA** achieves competitive performance, catching up with HAPPO after an initially slower convergence caused by the overhead of attention computation. This supports that RGA enhances representational expressivity without sacrificing asymptotic performance.

5.3. **Evaluation Performance on MPE Spread.** In evaluation (Figure 4), we observe that **HAPPO-RGA** slightly surpasses vanilla HAPPO in final performance. This improvement is attributed to RGA's capacity to model inter-agent dependencies and spatial configurations, enabling more coordinated actions. Even in a simple environment like MPE Spread, relation-aware reasoning contributes measurable benefits during test-time behavior.

FIGURE 3. Training rewards on the MPE Spread task for MAPPO, HAPPO, and HAPPO-RGA.

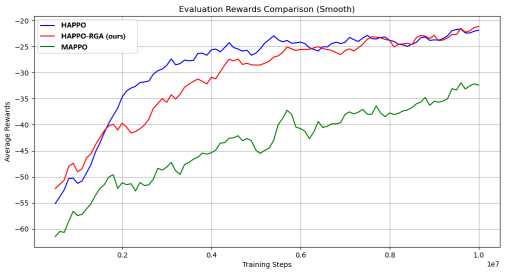

FIGURE 4. Evaluation rewards on the MPE Spread task for MAPPO, HAPPO, and HAPPO-RGA.

5.4. **Generalization to GRF.** To evaluate generalization, we test on GRF's *Counterattack Easy* and *Hard* scenarios. As shown in Figure 5, **HAPPO-RGA** demonstrates consistent gains over HAPPO, especially in the harder setting. This indicates that relation-aware modules like RGA scale well to environments with adversaries and long temporal dependencies. GRF's partial observability and dynamic nature highlight the value of global attention in multi-agent settings.

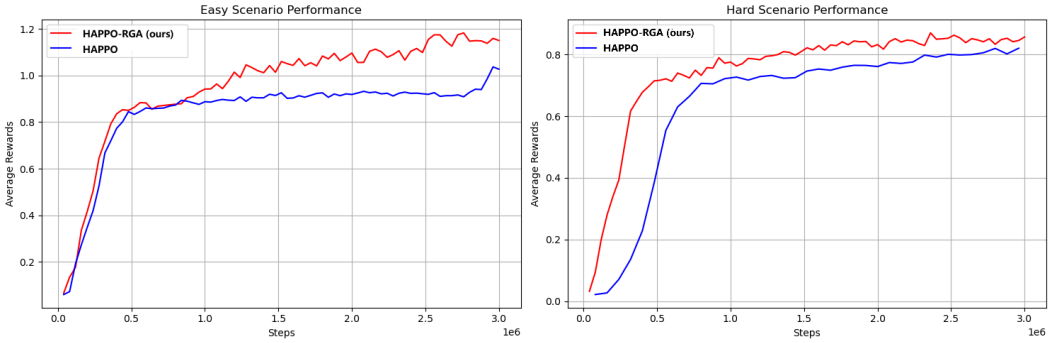

FIGURE 5. Evaluation performance of HAPPO and HAPPO-RGA in GRF Counterattack Easy (left) and Hard (right).

6. DISCUSSION

The experimental results confirm the effectiveness of our proposed HAPPO-RGA method. On the MPE Spread task, it achieves comparable or slightly better final performance than baseline HAPPO, with marginally slower early training due to the additional overhead of RGA computation. The improved final policy quality and coordination justifies this trade-off.

In the more complex GRF environments, HAPPO-RGA significantly outperforms HAPPO, particularly in the Hard scenario. These results indicate that the ability to model spatial and relational structures via RGA becomes increasingly valuable in environments requiring long-horizon planning and coordination under partial observability.

A noted limitation is the increased computational cost during training, and a higher value loss due to the critic's difficulty fitting richer representations. Future work may explore more expressive critic architectures, adaptive attention, or regularization techniques to address these issues.

Overall, HAPPO-RGA enhances multi-agent coordination through attention-based reasoning, scales to challenging tasks, and provides a promising direction for relational MARL frameworks.

REFERENCES

[1] Zhong, Y., Kuba, J. G., Feng, X., Hu, S., Ji, J., & Yang, Y. (2024). Heterogeneous-agent reinforcement learning. Journal of Machine Learning Research, 25:1–67.

[2] Zhang, Z., Lan, C., Zeng, W., Jin, X., & Chen, Z. (2020). Relation-aware global attention for person re-identification. In CVPR. arXiv preprint arXiv:1904.02998.

[3] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

[4] Kuba, J. G., Chen, R., Wen, M., Wen, Y., Sun, F., Wang, J., & Yang, C. (2021). Trust region policy optimisation in multi-agent reinforcement learning. arXiv preprint arXiv:2109.11251.