Encoding Goals as Graphs: Structured Objectives for Scalable Cooperative Multi-Agent Reinforcement Learning

Extended Abstract

Alessandro Amato*
University of West Florida & IHMC
Pensacola, USA
aamato@ihmc.org

Raffaele Galliera*
University of West Florida & IHMC
Pensacola, USA
rgalliera@ihmc.org

K. Brent Venable
University of West Florida & IHMC
Pensacola, USA
bvenable@uwf.edu

Niranjan Suri
University of West Florida & IHMC
Pensacola, USA
nsuri@ihmc.org

ABSTRACT

Many cooperative multi-agent tasks are naturally defined by graph-structured objectives, where agents must collectively achieve a desired relational configuration or satisfy a set of constraints. However, current goal-conditioned multi-agent reinforcement learning (MARL) methods rarely leverage such symbolic structure to guide learning. To address this challenge, we propose Graph Embeddings for Multi-Agent Coordination (GEMA), which augments any cooperative learner with a State-Graph Encoder (SGE). The SGE is pre-trained contrastively to embed state and goal graphs in a shared metric space. At run time, each agent constructs the state graph, queries the SGE, and computes a similarity score to the goal embedding. This similarity serves as an intrinsic reward, providing dense feedback on task progress, and is also incorporated into each agent's observation. Experiments on cooperative navigation, load balancing, and the StarCraft Multi-Agent Challenge (v2) show that GEMA accelerates convergence and improves team returns.

KEYWORDS

Multi-agent reinforcement learning; Graph neural networks; Goal-conditioned learning; Graph embeddings; Representation learning

ACM Reference Format:
Alessandro Amato, Raffaele Galliera, K. Brent Venable, and Niranjan Suri. 2026. Encoding Goals as Graphs: Structured Objectives for Scalable Cooperative Multi-Agent Reinforcement Learning: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/TDTU5449

1 INTRODUCTION

Numerous real-world tasks—ranging from robot formation control to load balancing in data centers—require teams of learning agents to steer a system towards a **graph-structured goal**: a desired configuration of entities and their relations [1, 2, 5, 6]. Traditional cooperative multi-agent reinforcement learning (MARL) algorithms face significant challenges in these settings. First, progress toward a goal is often only sparsely rewarded by the environment, which can impede learning. Second, traditional algorithms do not fully exploit the relational structure present in many tasks, missing the permutation-invariant inductive biases that graph-based representations naturally provide. We address these challenges with **Graph Embeddings for Multi-Agent Coordination (GEMA)**, a plug-in module that can be paired with any cooperative MARL algorithm. Before policy learning, we contrastively pre-train a **State-Graph Encoder (SGE)** that maps both the current system graph and the objective graph into a metric space calibrated to task progress. At run time, each agent reconstructs the current state graph, feeds both graphs through the frozen SGE, and obtains (i) a similarity feature appended to its private observation, and (ii) an intrinsic reward equal to that similarity. Experiments on cooperative navigation, load balancing, and StarCraft Multi-Agent Challenge (SMACv2) [4] show that GEMA accelerates convergence, improves asymptotic returns, and scales efficiently from three to ten agents.

2 METHOD

The intuition is to learn a latent representation space where the relational similarity of the current configuration and the desired goal can be meaningfully compared, and inform the agents of such (dis)similarities as they train their distributed policy. To this end, prior to policy training, we introduce a contrastive representation learning phase to train a SGE that captures relevant relational properties of the task. Once learned, we employ the SGE to allow each agent to compute an embedding of the current state and measure its similarity to the desired goal. This similarity serves as feedback for our agents and as an intrinsic reward signal that complements the environment reward.

2.1 Learning the State-Graph Encoder (SGE)

We represent each environment configuration as a *State-Graph*: an undirected attributed graph $G = (V, E)$, where nodes V correspond to entities (agents and other relevant objects) and edges encode pairwise relations. Similarly, a *Goal-Graph* encodes the target relational configuration the team should achieve.

*These authors contributed equally to this work.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/TDTU5449

Table 1: Summary of results across benchmarks. Training metrics report final performance from the learning curves; evaluation metrics report mean ± std over the stated number of episodes. Dashes indicate the method/metric was not reported or not applicable for that setting. CN=Cooperative Navigation, LB=Load Balancing. The higher the better.

Task	Metric	Actor	**GEMA**	MAPPO	MASAC	VDN	LAGMA	MASER	MADDPG	QMIX	QMIXRS
CN	Final return (training)	GNN	**0.936**	0.859	0.770	0.795	0.758	0.753	0.320	–	–
CN	Final return (training)	MLP	–	0.864	0.908	0.815	0.753	0.750	0.309	0.814	–
CN	Avg. return (eval), 3 agents	GNN	**93.79 ± 2.58**	86.59 ± 4.00	77.05 ± 7.92	81.18 ± 5.98	73.77 ± 13.51	74.72 ± 11.02	30.62 ± 11.27	–	–
CN	Avg. return (eval), 6 agents	GNN	**93.48 ± 1.47**	88.51 ± 2.20	84.07 ± 4.18	84.81 ± 3.39	79.16 ± 12.45	80.82 ± 9.28	36.70 ± 7.87	–	–
CN	Avg. return (eval), 10 agents	GNN	**93.08 ± 1.04**	89.69 ± 1.49	87.77 ± 2.62	86.65 ± 2.51	82.11 ± 12.16	84.00 ± 8.70	40.78 ± 5.99	–	–
LB	Final return (training)	GNN	**-0.008**	-0.158	–	-0.396	-0.365	-0.352	–	–	–
LB	Avg. return (eval)	GNN	**10.07 ± 59.23**	-4.85 ± 62.81	–	-23.61 ± 51.71	-40.98 ± 41.94	-44.57 ± 46.42	–	–	–
LB	Constraint-satisfying steps (eval)	GNN	**29.5 ± 38.9**	24.0 ± 36.5	–	13.2 ± 25.8	4.07 ± 11.19	4.18 ± 16.02	–	–	–
SMACv2	Win rate (training)	RNN	**0.56**	–	–	–	0.12	0.20	–	0.54	0.44

Training the SGE is conducted purely in an offline process where no policies are learned at this stage, and the environment is queried only to record diverse states. First, we construct a dataset of state–goal graph pairs annotated with coarse progress labels. Then, we embed graphs with a graph neural network (GNN)-based encoder [3] that produces permutation-invariant representations:

$$h_v^{(\ell)} = \phi_{\text{upd}}\Big(h_v^{(\ell-1)}, \bigoplus_{u\in\mathcal{N}(v)} \psi\big(h_v^{(\ell-1)}, h_u^{(\ell-1)}, e_{uv}\big)\Big),$$

$$f_\theta(G) = \mathcal{R}\Big(\Big\{h_i^{(\ell)} \mid v_i \in V\Big\}\Big).$$

Here, $\mathcal{N}(v)$ denotes neighbors of v. Functions ϕ_{upd} and ψ are multilayer perceptrons (MLPs) (parameters θ) shared across nodes and layers, enabling multi-hop relational context [9]. Both $\bigoplus$ and $\mathcal{R}$ are permutation-invariant so $f_\theta(G)$ is index-order independent.

Finally, we learn the latent space with a triplet contrastive loss [13] with adaptive margins, encouraging the cosine distance to reflect task progress:

$$\mathcal{L}_{\text{triplet}} = \Big[d_{\cos}\big(z^a, z^p\big) - d_{\cos}\big(z^a, z^n\big) + \alpha_{pn}\Big]_+.$$

where $d_{\cos}$ is the cosine distance, z^a the anchor, z^p the positive sample, z^n the negative, and α_{pn} the margin.

After this pretraining phase, the encoder parameters are frozen and reused during policy optimization.

2.2 Training the Agents

We integrate the SGE into the online learning loop by providing each agent with a shared task-level context derived from state–goal similarity and an intrinsic reward proportional to that similarity.

At each time step t, every agent constructs the current state graph G_t and the goal graph $G^\star$. The SGE produces embeddings $z_t = f_{\theta_{\text{sge}}}(G_t)$ and $z^\star = f_{\theta_{\text{sge}}}(G^\star)$. We then compute the cosine similarity $c_t = \cos(z_t, z^\star)$, which serves as a compact global signal shared among all agents. Each agent i concatenates c_t with its local observation $o_{t,i}$ and feeds the resulting vector to its policy (and value-function) head.

Finally, we use c_t as an intrinsic shaping reward,

$$r_t^{\text{int}} = c_t, \quad \tilde{r}_t = r_t^{\text{env}} + \beta\, r_t^{\text{int}},$$

where β controls the contribution of the intrinsic reward relative to the environment reward.

3 RESULTS

Table 1 summarizes results on three benchmarks: cooperative navigation (CN), load balancing (LB), and SMACv2. We compare GEMA against five standard multi-agent algorithms—MAPPO [15], MASAC [7], VDN [14], MADDPG [10], and QMIX [12]—and two goal-based baselines, LAGMA [11] and MASER [8]. Across CN and LB, GEMA is integrated into MAPPO, and into QMIX for SMACv2[1].

In cooperative navigation, the team succeeds when each agent reaches and occupies a distinct landmark. GEMA achieves the highest return, regardless of whether the baselines use a GNN- or MLP-based actor. To assess scalability, we also evaluate GEMA by increasing the number of agents (and targets) at test time without retraining. In this setting, GEMA still outperforms all baselines.

In load balancing, agents must maintain a target load across a set of cloud machines by deciding where to execute a stream of incoming jobs. Learning is driven by sparse rewards. Across all metrics, GEMA achieves better performance by leveraging the intrinsic reward induced by the SGE.

Finally, in SMACv2, GEMA operates under partial observability and therefore uses only the intrinsic reward (i.e., without concatenating the similarity score to the agents' observations). Even in this setting, GEMA attains a higher win rate. To highlight the importance of SGE pretraining, we include an ablation (QMIXRS) where the intrinsic reward is computed from raw state features rather than the learned SGE embedding. This variant underperforms, indicating that representation learning is critical to the method.

4 FUTURE WORK

We introduced GEMA, a novel plug-in that enables cooperative MARL algorithms to yield higher returns and maintains its edge as the team size grows. Future research will explore end-to-end online training of the SGE, partial observability, hierarchical subgoal discovery, distributional shift between the data used to train the SGE and on-policy states, and deployment in environments with imperfect state reconstruction.

This material is based on research sponsored by AFRL/RW under agreement number FA8651-25-1-0003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes, notwithstanding any copyright notation thereon. Approved for public release: distribution unlimited.

[1]Code: https://github.com/aamatodev/gema

REFERENCES

[1] Akshat Agarwal, Sumit Kumar, Katia Sycara, and Michael Lewis. 2020. Learning Transferable Cooperative Behavior in Multi-Agent Teams. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems* (Auckland, New Zealand) *(AAMAS '20)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1741–1743.

[2] Alessandro Amato, Alessandro Morelli, Mattia Fogli, Raffaele Galliera, and Niranjan Suri. 2024. Multi-Agent Reinforcement Learning for Distributed Workflow Orchestration at the Tactical Edge. In *MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM)*. 64–69. https://doi.org/10.1109/MILCOM61039.2024.10773787

[3] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. 2018. Relational inductive biases, deep learning, and graph networks. *CoRR* abs/1806.01261 (2018). arXiv:1806.01261 http://arxiv.org/abs/1806.01261

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

[5] Raffaele Galliera, Thies Mohlenhof, Alessandro Amato, Daniel Duran, Kristen Brent Venable, and Niranjan Suri. 2024. Distributed autonomous swarm formation for dynamic network bridging. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 1–6.

[6] Raffaele Galliera, Kristen Brent Venable, Matteo Bassani, and Niranjan Suri. 2025. Collaborative Information Dissemination with Graph-Based Multi-Agent Reinforcement Learning. In *Algorithmic Decision Theory*, Rupert Freeman and Nicholas Mattei (Eds.). Springer Nature Switzerland, Cham, 160–173.

[7] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. Pmlr, 1861–1870.

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

[9] Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. 2020. Graph Convolutional Reinforcement Learning. arXiv:1810.09202 [cs.LG] https://arxiv.org/abs/1810.09202

[10] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6382–6393.

[11] Hyungho Na and Il-Chul Moon. 2024. LAGMA: latent goal-guided multi-agent reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning* (Vienna, Austria) *(ICML'24)*. JMLR.org, Article 1506, 19 pages.

[12] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research* 21, 178 (2020), 1–51.

[13] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 815–823. https://doi.org/10.1109/cvpr.2015.7298682

[14] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2018. 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* (Stockholm, Sweden) *(AAMAS '18)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2085–2087.

[15] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2021. The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games. *arXiv preprint arXiv:2103.01955* (2021).