BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2026
http://excel.fit.vutbr.cz

Reward Design Selects Coordination Regimes in Cooperative Multi-Agent Reinforcement Learning

Martin Ševčík*

Abstract

Reward curves alone do not tell us how a multi-agent team coordinates. We train MAPPO in a JAX port of Knights, Archers, and Zombies under seven reward schemes and score policies on spatial structure, role reliance, and behavior. The environment adds shielded zombies that only knights can break, and an XP/leveling system that rewards fair progression. We measure how rewards and mechanics such as parameter-sharing paradigms and vision radius alter behavior within this structure: Survival policy yields the clearest frontline-backline (knights in front, archers in back) regime and best durability in the primary setup, Coalition policy converges to a compact mixed-role cluster, and Shared Reward policy to a more aggressive separated style; knight survival predicts episode length better than focus fire or behavioral diversity, though less strongly once archers can also break shields. We test reward-scheme compatibility through cross-playing between policies and role dependence through leave-one-out tests.

*xsevcim00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Cooperative multi-agent reinforcement learning (MARL) is usually evaluated by return or episode length, but that does not explain how a team succeeded. In small heterogeneous teams, policies with similar performance may rely on different role structures, spatial organization, and failure modes [1, 2].

This paper studies that issue in *Knights, Archers, and Zombies* (KAZ), a cooperative benchmark from PettingZoo with two knights and two archers [3]. With only four agents, question is whether different incentives produce different *coordination regimes*: who protects whom, how tightly the team moves, and which role becomes the bottleneck.

We build on three established pieces: MAPPO as a cooperative MARL baseline [4], reward shaping as a lever on social behavior [5], and role-diversity metrics for specialization [6]. We keep the task fixed, vary only incentives, and ask whether the resulting policies differ only in performance or also in coordination style.

Our main contributions are:

- a custom JAX-based KAZ environment with bosses, waves, XP leveling, and shielded zombies, still efficient enough for GPU training [7, 8],
- seven reward schemes, a metric suite for coordination structure and role burden, and Replay Browser for qualitative replay inspection,
- evidence that reward design selects distinct coordination regimes, while mechanics and parameter sharing decide which remain effective.

2. Environment and Experimental Setup

To make KAZ demanding enough for coordination analysis, we add bosses and wave progression, XP leveling, and shielded zombies that only knights can break.

The environment is implemented in JAX as pure `reset` and `step` functions, vectorized with `jit`, `vmap`, and `lax.scan`. On one NVIDIA RTX 4070 Super GPU, training sustains roughly 130 000 frames per second. We train MAPPO with a centralized critic and decentralized actors [4]. The default setup uses *type-shared* policies, one network for knights and one for archers. Actors and critic are three-layer MLPs with 256 hidden units.

We compare seven reward schemes with fixed dynamics: *Baseline* (individual kills), *Shared Reward* (half shared among attackers), *Egalitarian* (penalized XP inequality), *Survival* (alive bonus and teammate death penalty), *Coalition* (reward shared with nearby allies), *Zero-Sum* (small competitive transfer), and *Territorial* (role-owned

map halves). Together they encourage local sharing, equalized progression, conservative protection, compact grouping, weak competition, or explicit lane ownership.

The custom metrics cover *spatial structure*: clustering, front–back formation, role separation, and coalition lifetime; *coordination*: focus fire, system neural diversity (SND) [9], and knight close-commit; and *protection*: unsupported archer exposure, survival ratios, knight HP bleed, and XP inequality. Together they track team compactness, whether knights screen for archers, whether agents attack together or split up, and which role carries the load. Primary training uses five seeds, 256 parallel environments, 128 rollout steps, 10 000 updates, and 400 px vision. Final evaluation uses 500 deterministic episodes per seed. We also vary vision, parameter sharing, and shield rules.

3. Results

The seven rewards yield distinct coordination regimes. Table 1 summarizes performance, by reporting mean episode length across all evaluated setups.

Policy	TS	NS	SH	Vision	ABS
Survival	**701.1**	509.8	504.3	583.9	572.9
Shared Reward	608.8	508.2	**585.5**	**666.4**	**668.3**
Coalition	583.3	**538.1**	479.7	539.3	633.8
Baseline	556.3	510.0	379.0	501.6	571.6
Egalitarian	552.8	456.5	439.0	477.5	551.8
Zero-Sum	534.4	428.4	402.3	431.5	539.7
Territorial	431.7	431.2	361.1	474.3	525.2

Table 1. Mean episode length. TS: primary type-shared setup with 400 px vision; NS/SH: non-shared and fully shared parameters; Vision: unlimited vision; ABS: archers-break-shields.

Survival is the most durable policy in the primary setup. It has the clearest frontline–backline structure, the largest role separation, and the lowest unsupported archer exposure. It does *not* win by being the most aggressive: archer attack frequency is only 0.272 and knight HP bleed remains low. In the shield-gated environment, it succeeds by protecting the load-bearing role and taking cleaner fights.

Coalition is the most compact regime. It has the highest clustering, the lowest role separation, and the longest coalition lifetime median (51.7 steps). Coalition prefers dense mixed-role movement where archers stay close enough to benefit from local sharing and knight cover.

Shared Reward is the aggressive hybrid. It keeps a clearer front–back structure than Coalition but behaves more aggressively than Survival, with agents clustering near the top.

Baseline and **Zero-Sum** end up surprisingly close, suggesting that the task absorbs the weak competitive transfer in Zero-Sum.

Egalitarian and **Territorial** induce the intended local behaviors, but both damage performance, showing that a reward can push behavior in the intended direction without producing a robust policy.

The second result is that many intuitive coordination proxies are weaker than they look. Across the primary evaluation, focus fire correlates *negatively* with episode length ($\rho = -0.759$). After demeaning within reward families, spatial metrics lose most predictive power while knight-side burden management stays dominant. Several metrics identify regime, not robustness.

Parameter sharing also shifts the outcome. Type-shared (TS) policies perform best for all seven rewards. Non-shared (NS) keeps some visible structure, but underperforms. And with one shared policy for both knights and archers (SH), agents end up with almost equal XP, suggesting less role-specific behavior.

Vision radius pulls in the same direction. Limited vision (400 px) is equal or better for five of seven rewards. Coalition is the clearest case: with full-map vision, it stops clustering, shifts toward a cleaner front–back formation, and still performs worse. Seeing the whole map washes out the local incentives that made compact play effective.

Finally, the archers-break-shields ablation (ABS), cross-play, and leave-one-out analysis quantify reliance on the built-in knight bottleneck. Once archers can also break shields, Shared Reward becomes best, Coalition improves, Territorial gains the most, and Survival loses its lead. Knight survival remains important but less decisive. Cross-play shows limited transfer between learned conventions: 41 of 42 off-diagonal pairs are below self-play, with only Territorial archers plus Coalition knights slightly above parity (1.019). Leave-one-out shows the same pattern more directly: removing a knight is far more damaging than removing an archer in every tested policy; for Survival the drop is 451.0 vs 37.1 steps.

4. Conclusion

Teams coordinate in KAZ even under the baseline reward. Reward shaping changes the *form* that coordination takes: compact clustering, sharp frontline–backline separation, or an aggressive hybrid. Knights are load-bearing by design; the real question is which reward schemes lean hardest on that bottleneck, and which become competitive once it is relaxed.

References

[1] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*, pages 321–384. Springer, 2021.

[2] Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press, 2024.

[3] Farama Foundation. Knights archers zombies (KAZ) — PettingZoo documentation. https://pettingzoo.farama.org/environments/butterfly/knights_archers_zombies/, 2024. Accessed: 2026-04-01.

[4] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre M. Bayen, Yuantao Gu, and Yi Wu. The surprising effectiveness of MAPPO in cooperative multi-agent reinforcement learning. *arXiv preprint*, arXiv:2103.01955, 2021.

[5] Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 464–473, 2017.

[6] Siyi Hu, Chuanlong Xie, Xiaodan Liang, and Xiaojun Chang. Policy diagnosis via measuring role diversity in cooperative multi-agent RL. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 9041–9071, 2022. PMLR volume 162.

[7] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs. http://github.com/google/jax, 2018. Version 0.3.13.

[8] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX. http://github.com/google/flax, 2020. Version 0.12.0.

[9] Matteo Bettini, Ajay Shankar, and Amanda Prorok. System neural diversity: Measuring behavioral heterogeneity in multi-agent learning. *Journal of Machine Learning Research*, 26(163):1–27, 2025.