Extended Abstract

MiniHedgemony: Asymmetric Reward Structures in Self-Play Wargames

Alex Wang and Dario Soatto

Motivation. Strategic wargames are a natural but underexplored testbed for reinforcement learning because they combine sparse, adversarial outcomes with long-horizon planning, partial observability, simultaneous actions, and role asymmetries. We built MiniHedgemony, a simplified reinforcement-learning version of RAND's *Hedgemony*, a strategic resource-allocation wargame originally developed to help defense planners reason about force structure, posture, modernization, readiness, and hedging under uncertainty (1). Rather than asking only whether self-play can optimize for winning, our central question is whether the structure of the reward function changes how agents play. In particular, we test whether reward asymmetries such as loss aversion and action cost lead self-play agents to discover qualitatively different and repeatable strategic behaviors while holding the action space and initial conditions fixed.

Method. MiniHedgemony is a two-player, 20-turn, simultaneous-move game with action masking, conflict resolution, role-specific objectives, and a 23-dimensional observation for each agent. We implemented the environment as a fully JAX-native pure function inspired by the PureJaxRL design philosophy (6). This keeps the environment, policy inference, and learning loop inside JAX, allowing the step function to be JIT compiled and vectorized over many parallel game instances. Training uses recurrent PPO to handle partial observability and a self-play league inspired by AlphaStar (7). Before entering league training, each side is cold-started against a heuristic rule-based opponent for 150 million environment steps. This gives the league a stable initial policy pool rather than beginning from two random policies that can reinforce degenerate behavior. After this warm start, we maintain frozen snapshot pools for each side; the active learner plays opponents sampled from the opposing pool, snapshots are added at fixed intervals, and the learning side alternates periodically.

Experiments and Results. We evaluated policies through generation-wise round-robin tournaments in which every snapshot from one side plays every snapshot from the other side. We record win rates, timeout rates, and Elo over training generations. The symmetric reward regime, which encodes the game's native scoring rules without added behavioral incentives, produced the clearest evidence of stable improvement: later generations consistently beat earlier ones over roughly 50 generations, and the Elo chart shows the same upward trend over time. Training also revealed a strong role asymmetry. Fitting Elo against main-training timesteps over the stable training window, the offensive side improves about $1.8\times$ faster per unit of compute than the defensive side ($+139.7$ versus $+76.9$ Elo per 100M timesteps) and far more reliably (attack $R^2 = 0.983$ versus defense $R^2 = 0.880$). Defense starts from a higher baseline but its returns to additional training are smaller and noisier. We therefore allocated a larger training budget to defense rather than forcing attack and defense to receive equal compute. We interpret the slower, noisier defensive improvement as evidence that defense depends more on opponent modeling and historical context than offense in MiniHedgemony.

Findings. Our main negative result is that added asymmetric reward shaping did not improve strategic behavior. Loss-averse and action-cost variants suppressed exploration, increased timeouts, and encouraged passive or indecisive play. In several failed regimes, both sides learned behaviors that jointly preserved score rather than forcing decisive competition. This suggests that MiniHedgemony already rewards hedging through its native game mechanics; additional reward terms can double-count this incentive and push policies toward overly conservative equilibria. The result supports a broader lesson: in strategically rich games, reward shaping is not a neutral training convenience but an intervention that can alter the strategic equilibrium.

Conclusion. MiniHedgemony provides a compact accelerator-friendly benchmark for studying self-play under role asymmetry and reward design. Our league and recurrent PPO pipeline yields stable improvement under symmetric rewards, but asymmetric shaping terms destabilize or distort training. Future work should stabilize the failed regimes, cluster trajectory-level behavioral features across seeds and reward settings, compare alternative league structures such as larger-scale OpenAI Five-style self-play (8), and evaluate whether longer training can close the remaining gap between learned policies and human play.

MiniHedgemony: Asymmetric Reward Structures in Self-Play Wargames

Alex Wang
Department of Computer Science
Stanford University

Dario Soatto
Department of Computer Science
Stanford University

Abstract

We introduce MiniHedgemony, a simplified reinforcement-learning environment based on RAND's strategic wargame *Hedgemony*. The environment is a two-player, simultaneous-move, partially observable game in which agents allocate finite resources over a 20-turn horizon while responding to a dynamic opponent. We use MiniHedgemony to study whether asymmetric reward structures alter the qualitative strategies discovered through self-play. Our implementation is fully JAX-native and supports accelerator-resident, vectorized training. We train recurrent PPO agents in a snapshot-based self-play league after a 150 million step cold-start phase against heuristic rule-based opponents, and evaluate policies with generation-wise round-robin tournaments, win rates, timeout rates, and Elo over time. Symmetric self-play, where rewards directly encode the game's native rules, produces stable generation-over-generation improvement over roughly 50 generations. We further find a pronounced role asymmetry: offensive policies gain Elo roughly $1.8\times$ faster per unit of compute than defensive policies and with much lower variance, so we budget additional training to defense. By contrast, loss-averse and action-cost shaping regimes suppress exploration and lead to passive, high-timeout play. These findings suggest that in strategically rich environments, reward shaping can distort rather than sharpen strategic behavior, especially when the base game already embeds incentives such as hedging and risk management.

1 Introduction

Deep reinforcement learning has achieved impressive results in competitive games, including StarCraft II and Dota 2, largely through large-scale self-play (7; 8). These systems typically optimize a natural competitive objective: win the game. However, many real strategic domains are not merely about maximizing a terminal win indicator. They include asymmetric roles, long-term resource constraints, partial information, hidden tradeoffs, and incentives that may reward patience, deterrence, hedging, or selective escalation. In such domains, the reward function is not only a scalar learning signal; it is also a design choice that can shape the type of strategy an agent discovers.

This project studies that issue in the context of strategic wargaming. We build MiniHedgemony, a simplified reinforcement-learning version of RAND's *Hedgemony*, a strategic choices game designed to help players reason about finite defense resources, force posture, readiness, modernization, and planning under uncertainty (1). The original game is compelling for reinforcement learning because it is not a short-horizon tactical simulator. It asks players to manage competing strategic commitments over time while responding to an opponent whose goals and constraints are not identical to their own.

Our central research question is: *holding the game dynamics fixed, do asymmetric reward structures lead self-play agents to learn qualitatively different strategic behaviors?* We focus on reward asymmetries such as loss aversion and action cost. The goal is not only to improve final performance but to test whether reward design can be used as a behavior-steering tool in a strategic environment.

We make three contributions. First, we implement MiniHedgemony as a JAX-native environment suitable for high-throughput self-play. Second, we design a recurrent PPO self-play league with frozen opponent snapshots to reduce cycling and improve training stability. Third, we compare symmetric and asymmetric reward regimes and find that the symmetric game-rule reward produces the strongest and most stable learning, while asymmetric shaping often degrades behavior by encouraging passive, timeout-heavy equilibria. Because the shaping regimes destabilized before passing our monotonicity gate, our central finding is this destabilization itself rather than a behavioral comparison across stable regimes.

2 Related Work

Strategic wargaming and Hedgemony. RAND's *Hedgemony* was developed as a game of strategic choices in which players allocate constrained resources across force structure, posture, modernization, and readiness while hedging against uncertain futures (1). Our work does not attempt to reproduce the full RAND game. Instead, MiniHedgemony abstracts the core reinforcement-learning-relevant features (limited resources, role asymmetry, sequential interaction, simultaneous moves, and strategic tradeoffs over a multi-turn horizon) just as SMAX abstracts the StarCraft Multi-Agent Challenge into a fast, self-play-friendly environment by removing the underlying game engine (2).

Reward shaping. Reward shaping is a standard tool for accelerating reinforcement learning, but it is well established that an arbitrary added reward term can change the optimal policy. Ng et al. show that only potential-based shaping preserves the optimal policy of the underlying MDP (3). Our shaping terms are deliberately not potential-based: a loss-aversion penalty and an action cost are functions of outcomes and actions rather than differences of a state potential, so we should expect them to move the equilibrium. Furthermore, in multi-agent settings, even potential-based shaping that preserves Nash equilibria can change which equilibrium is reached and can affect performance in either direction depending on the heuristic (4), and risk-sensitive objectives such as loss aversion are known to resist faithful translation into a reward term: Greenberg et al. argue that a naive encoding of risk aversion into the reward typically yields over-conservative policies, and that in general no reshaped reward recovers the intended risk-sensitive optimum (5). Our finding that loss-averse shaping produces passive, timeout-heavy play is a concrete instance of this failure mode in a competitive self-play setting.

Self-play in competitive games. Self-play has produced strong results in complex multi-agent games, including StarCraft II and Dota 2 (7; 8). A recurring difficulty is that training against only the latest opponent is unstable and cyclic because the learner overfits to the current policy and becomes exploitable by older styles it no longer sees (9). Classical fictitious play addresses this by best-responding to the average of past opponents and converges to Nash equilibria in two-player zero-sum games; neural fictitious self-play extends this idea to partially observable, function-approximation settings (9). AlphaStar's league of historical opponents and exploiters is a practical realization of the same principle at scale (7). Our snapshot pool is a deliberately small instance of this family: rather than building a full population-based training system, we keep enough frozen historical opponents to prevent immediate overfitting and to support generation-wise evaluation. This is important in MiniHedgemony because, as our v2/v3 experiments show, direct head-to-head training can collapse as each update shifts the opponent distribution faced by the other side.

JAX-native reinforcement learning. PureJaxRL demonstrates the benefit of keeping the full reinforcement-learning loop inside JAX, including the environment, rollout, and optimization (6). We follow this design principle because MiniHedgemony requires many game instances and repeated self-play evaluations. A conventional CPU environment with GPU policy inference would require frequent CPU-GPU transfers; our pure functional step function avoids that bottleneck and can be JIT compiled and vectorized.

3 MiniHedgemony Environment

MiniHedgemony is a compact two-player strategic game intended to preserve the core learning challenges of Hedgemony while remaining simple enough for repeated deep RL experimentation. Each episode lasts 20 turns. On every turn, both players choose actions simultaneously. The

environment applies action masks, resolves conflicts, updates resources and strategic state, and produces role-specific observations and rewards. Each agent observes a 23-dimensional vector that summarizes its local state, available resources, selected public information, and game progress.

The two sides are intentionally asymmetric. They do not face identical strategic problems, and their learning curves differ. The offensive side can often improve by discovering pressure patterns that exploit available opportunities. The defensive side must respond to a changing opponent, protect against multiple possibilities, and manage resources across a broader set of contingencies. This distinction later appears clearly in the training results: defense improves more slowly and less reliably, which we quantify in Section 6.2.

The environment is implemented as a pure function

$$(s_{t+1}, o_{t+1}, r_t, d_t, m_{t+1}) = f(s_t, a_t, k_t),$$

where s_t is the full game state, a_t is the simultaneous joint action, k_t is the random key, o_{t+1} is the observation, r_t is the reward, d_t is the done flag, and m_{t+1} is the next action mask. This formulation is compatible with JAX transformations such as `jit` and `vmap`. In practice, this allows many games to be rolled out in parallel on the accelerator.

4 Method

4.1 Policy Optimization

We train agents with Proximal Policy Optimization (PPO) (10). Early experiments with feed-forward PPO were unstable and often failed to maintain meaningful strategic behavior. The environment is partially observable: a single observation does not reveal the full strategic context or the opponent's latent plan. As a result, the policy needs memory. We therefore use recurrent PPO so that each agent can integrate information over multiple turns.

We also found that policy drift had to be constrained. In one training variant, vanilla PPO initially improved but later collapsed before partially recovering. A later variant added a KL penalty against a reference snapshot, which stabilized learning throughout training. This result suggests that Mini-Hedgemony is sensitive to rapid changes in policy behavior, likely because the opponent distribution is itself nonstationary in self-play.

4.2 Self-Play League

Training two policies only against each other can lead to cycling. A policy may overfit to the latest opponent, then become vulnerable to an older style of play that it no longer encounters. To reduce this effect, we train with a snapshot-based self-play league. For each side, we maintain a pool of frozen policy snapshots. At any point, one side is the active learner. Its opponent is sampled from the frozen snapshot pool of the other side. At fixed intervals, the learner is saved into its side's pool, and the active training side alternates periodically.

This league structure is inspired by AlphaStar's use of historical opponents (7), though our implementation is much smaller. The goal is not to build a full population-based training system but to provide enough opponent diversity to prevent immediate overfitting and to enable generation-wise evaluation.

4.3 Reward Regimes

We compare a symmetric reward regime against asymmetric shaping variants. In the symmetric regime, rewards directly encode the game's native rules. This is the closest analogue to optimizing the actual game objective. In the asymmetric variants, we add terms intended to induce behavioral differences. The loss-averse variant penalizes unfavorable outcomes more strongly, while the action-cost variant discourages unnecessary or expensive actions. Conceptually, these variants test whether reward shaping can create more cautious, efficient, or strategically disciplined policies.

The key methodological point is that reward is the independent variable. The action space, environment dynamics, initial conditions, and training pipeline remain fixed as much as possible. This lets us attribute behavioral differences to the reward regime rather than to changes in game mechanics.

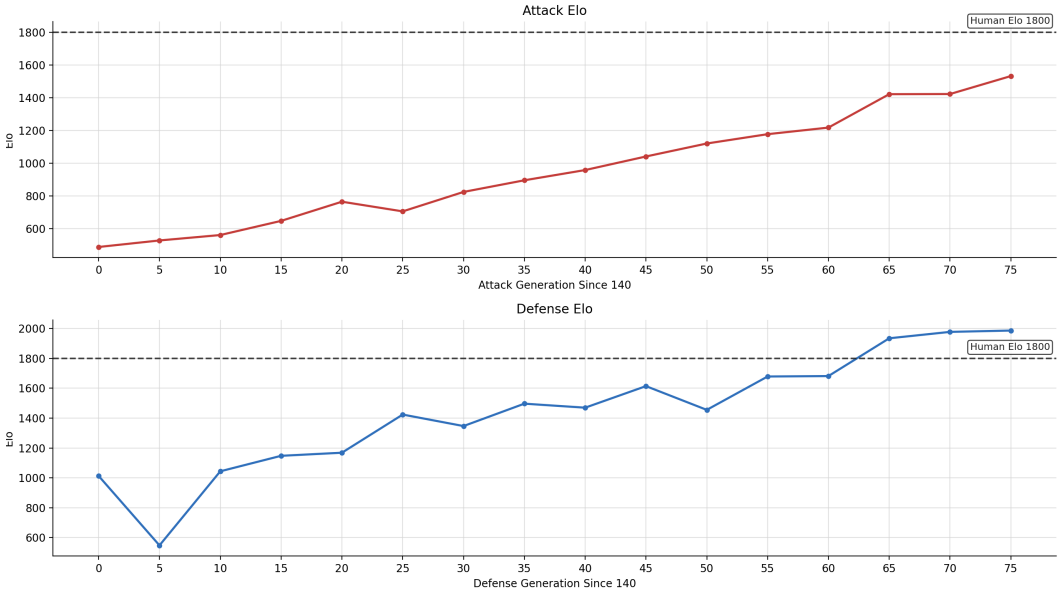

Figure 1: Elo over training generations: Elo rises for later generations under symmetric self-play.

5 Experimental Setup

5.1 Training

We train separate league runs for the symmetric reward and for the asymmetric variants. Each run produces a sequence of policy snapshots for each side. Because the two roles differ in how efficiently they convert compute into strength, we do not split training equally between them. Elo-versus-timestep fits over the stable training window show that defense improves roughly half as fast as offense per unit of compute ($+76.9$ versus $+139.7$ Elo per 100M timesteps) and with substantially higher residual variance. Equal compute would therefore leave defense both weaker and less stable at evaluation time. To compensate, we give defense a larger training budget: attack-side runs used a 10 million-step budget once the cold-start phase produced a stable baseline, while defense-side runs used a 20 million-step budget before snapshots were admitted into league evaluation. The asymmetry in budgets is thus a consequence of the measured asymmetry in learning efficiency, not an independent assumption.

5.2 Evaluation

Our primary evaluation is a generation-wise round-robin tournament. Every snapshot from one side is evaluated against every snapshot from the other side. We record win rates, timeout rates, and Elo ratings over time. A reward regime is considered eligible for cross-regime behavioral comparison only after it passes a monotonicity gate: later generations should reliably outperform earlier generations. This gate prevents us from overinterpreting trajectory-level behavior from policies that have not learned a stable strategic baseline.

5.3 Metrics

Win rate measures competitive performance. Timeout rate measures whether policies are producing decisive games or drifting into passive play. Elo provides a compact longitudinal measure of relative strength across generations. Together, these metrics distinguish policies that merely avoid losing from policies that improve strategically.

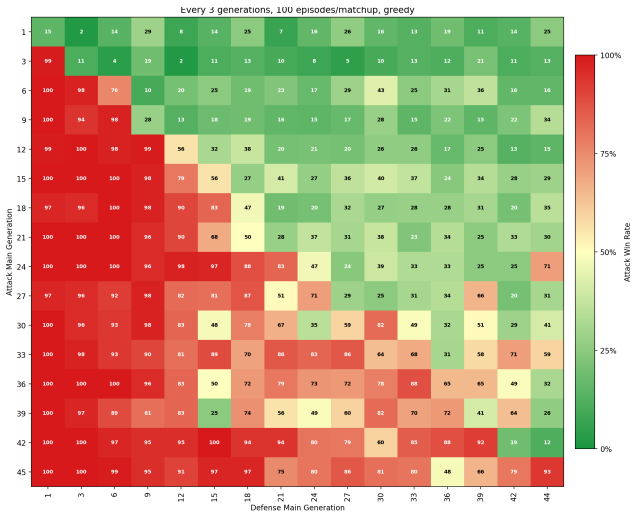

Figure 2: Symmetric self-play league. Later generations consistently beat earlier generations, indicating close to monotonic improvement under the native game-rule reward.

6 Results

6.1 Symmetric Self-Play Improves Steadily

The symmetric reward regime produced the clearest successful result. Over roughly 50 generations, later policies consistently beat earlier policies in round-robin evaluation (Figure 2; the corresponding Elo trend is shown in Figure 1). Over the full run, offense starts much lower (below 300) and climbs to nearly 1600, while defense starts higher (near 1000) and reaches nearly 2000; offense thus covers more ground from a weaker base. (These spans are read from the full league in Figure 2; the regression fits in Section 6.2 cover generation 140 onward, hence their higher intercepts.) This indicates that the league produces generation-over-generation refinement rather than only short-term adaptation to the latest opponent.

This result is important because it validates the training and evaluation pipeline. Before asking whether asymmetric rewards produce distinct strategic styles, we first need a regime in which agents can learn the base game. The symmetric reward regime satisfies this requirement and gives us a stable reference point.

6.2 Defense Improves More Slowly and Less Reliably Than Offense

Training exposed a clear role asymmetry, but not the simple one of raw budget. To quantify it, we fit Elo against main-training timesteps for each role over the stable training window (generations 140 onward), with C measured in millions of timesteps:

$$\text{Elo}_{\text{attack}} = 433.34 + 1.397\,C, \qquad \text{Elo}_{\text{defense}} = 859.67 + 0.769\,C.$$

The two roles differ in both level and slope. Defense operates from a much higher baseline (intercept 859.67 versus 433.34 Elo), consistent with the cold-start phase leaving defense in a stronger initial position. But attack improves roughly $1.8\times$ faster per unit of compute: $+139.7$ Elo per 100M timesteps versus $+76.9$ for defense. In other words, additional training buys offense more than it buys defense. Figure 3 shows these fits, and Figure 4 shows the corresponding timestep requirements for each role.

The fit quality reinforces this asymmetry. The attack trend is tight and nearly linear ($R^2 = 0.983$, RMSE $= 42.0$ Elo), whereas the defense trend is substantially noisier ($R^2 = 0.880$, RMSE $= 131.0$ Elo)—the residual scatter for defense is roughly three times larger. Defense therefore improves both more slowly and less predictably with compute.

We interpret this as evidence that defense is the harder learning problem despite its higher absolute Elo. An offensive policy can often improve by sharpening a relatively narrow set of pressure tactics,

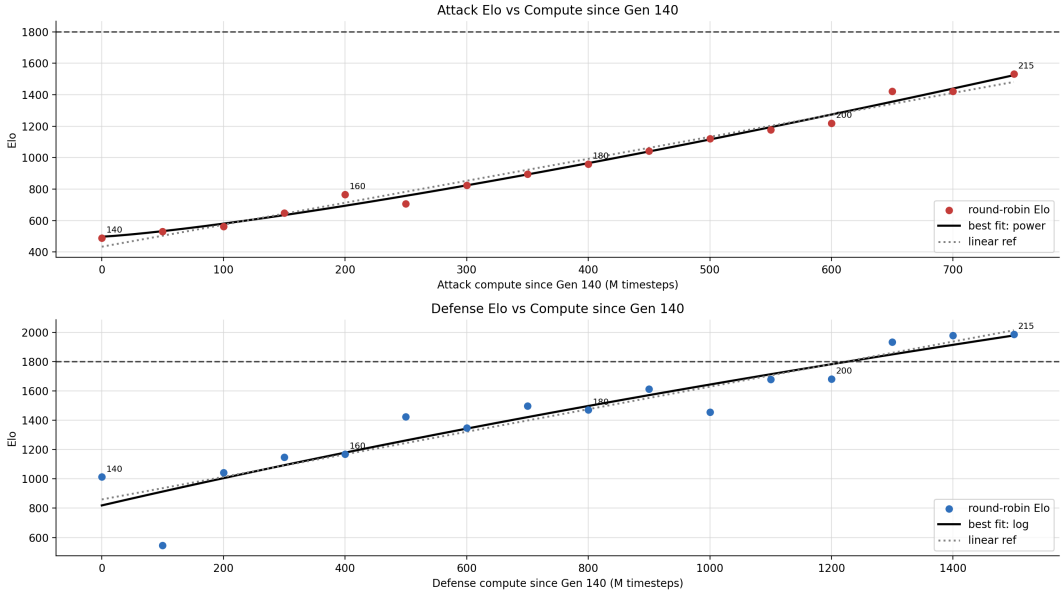

Figure 3: Attack and defense Elo scaling as a function of cumulative main-training compute.

Figure 4: Role asymmetry in training. Defense improves more slowly and less reliably than offense per unit of compute, consistent with greater dependence on opponent modeling and history.

which shows up as steady, low-variance Elo gains. A defensive policy must hedge across multiple possible offensive plans and depends more heavily on opponent strategy and history; this contingency makes its returns to additional training both smaller and noisier. The high RMSE for defense suggests its performance depends on factors not captured by timestep count alone, such as the particular mix of offensive snapshots it faces in a given generation.

This asymmetry shaped our experimental design. Our goal here is fair evaluation—comparable competence between the two roles—rather than maximizing aggregate Elo per step. Equal compute would compare a fast-improving offense against a slower-improving defense and risk evaluating an undertrained defender; we therefore budgeted additional training to defense before admitting its snapshots into league evaluation. It also tempers the headline result: because defense gains less per step, closing the gap to human-level defensive play likely requires disproportionately more compute than the offensive side.

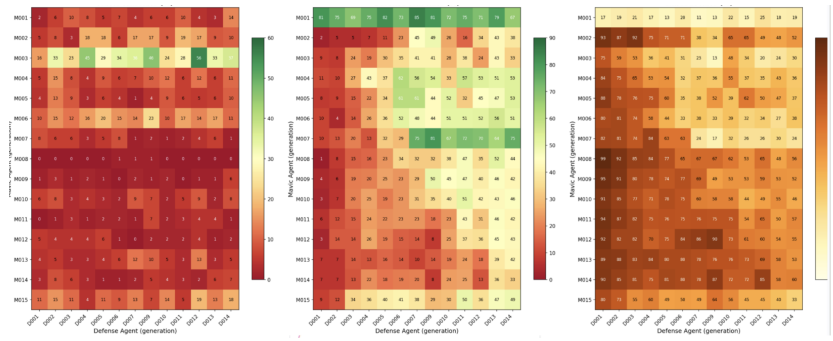

Figure 5: Risk-averse asymmetric variant. Added risk-aversion produced degenerate, indecisive play with high timeout rates, consistent with overemphasizing a hedging incentive already present in the game.

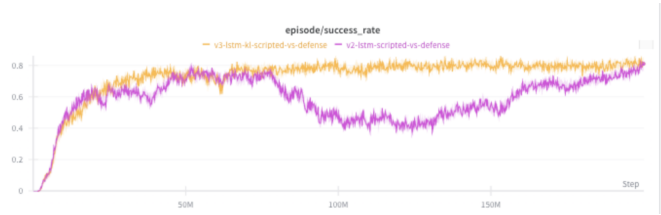

Figure 6: Training stability under policy-drift control. The v3 configuration (KL penalty against a reference snapshot) maintains stable Elo, whereas vanilla v2 collapses before partially recovering.

6.3 Asymmetric Reward Shaping Degrades Training

The main negative result is that asymmetric reward shaping did not improve strategic behavior. The symmetric regime outperformed the loss-averse and action-cost variants. In failed asymmetric runs, policies often drifted toward high timeout rates and passive play. Instead of competing decisively, both sides sometimes learned to preserve score or avoid costly engagement (Figure 5). The policies converged to a degenerate, non-improving regime (high timeout, passive play), so generation-wise Elo comparison was not meaningful.

This failure mode is especially informative. Hedgemony is already a game about hedging. Its native rules reward strategic caution, resource management, and preparation for uncertain futures. Adding explicit loss-aversion or action-cost terms can therefore double-count caution. Rather than producing more sophisticated strategic restraint, the additional shaping pushes agents toward indecision.

6.4 Constraining Policy Drift Improves Stability

Training stability depended heavily on controlling policy drift. We refer to successive training configurations as v2 (vanilla recurrent PPO) and v3 (the same setup with an added KL penalty). In the v2 training setup, vanilla PPO improved initially but later collapsed before partially recovering. In v3, a KL penalty against a reference snapshot kept training more stable (Figure 6). This suggests that the interaction between PPO updates and self-play nonstationarity is a central practical challenge in MiniHedgemony. Because each policy update changes the opponent distribution faced by the other side, large updates can destabilize the league even if they improve short-term returns.

7 Discussion

The project began with the hypothesis that reward asymmetries might produce qualitatively distinct strategic styles. The results complicate that hypothesis. We did find that rewards strongly affect behavior, but not in the straightforward positive sense that shaped rewards yield better or more interpretable strategies. Instead, the most successful agents were trained on the native game rules, and additional shaping terms often distorted play.

This has two implications. First, reward shaping should be treated as a strategic modeling decision, not only an optimization trick. In a game where the native objective already captures caution and hedging, adding another caution-inducing term can move the equilibrium away from competitive play. Second, evaluating only final score is insufficient. The asymmetric agents did not merely score worse; they failed in a qualitatively different way by producing passive, timeout-heavy trajectories. Behavioral diagnostics such as timeout rate, trajectory clustering, and action-distribution analysis are therefore necessary for understanding self-play outcomes.

The need for recurrent policies and KL-constrained updates also indicates that MiniHedgemony has real learning complexity despite being a simplified environment. Feed-forward PPO was not enough, and unconstrained policy updates were unstable. These failures are useful: they suggest that the environment captures meaningful partial observability and nonstationarity rather than reducing to a trivial Markov game.

8 Limitations and Future Work

The trained agents are not yet human-level. Performance appears to improve with additional training, so longer runs may narrow the gap, but it remains unclear whether the current environment and algorithm are sufficient for human-level strategic play. We also have not yet completed the full behavioral clustering analysis originally envisioned. Such analysis would extract trajectory-level features, cluster behaviors across seeds and reward regimes, and test whether reward structures produce repeatable strategic styles even when they do not improve win rate.

Future work should stabilize the loss-averse and action-cost regimes enough for them to pass the monotonicity gate, then compare their learned behaviors under controlled conditions. Additional league designs should also be tested. Our current snapshot pool follows the spirit of AlphaStar, but larger-scale population training closer to OpenAI Five may provide stronger opponent diversity and reduce overfitting. Finally, richer evaluation against human players or scripted strategic baselines would help calibrate whether league improvement corresponds to strategically meaningful play.

9 Conclusion

We built MiniHedgemony, a JAX-native self-play environment for studying asymmetric reward structures in strategic wargames. The environment supports fast accelerator-resident training and captures key challenges of strategic interaction: role asymmetry, simultaneous actions, partial observability, and long-horizon resource allocation. A recurrent PPO snapshot league produced stable improvement under the symmetric native game-rule reward. However, asymmetric reward shaping through loss-aversion and action-cost terms degraded training, increasing passive play and timeout rates. Our findings suggest that in strategic games, reward design can steer behavior, but it can also distort the equilibrium when it duplicates incentives already present in the environment. The safest and strongest baseline in MiniHedgemony was to let the game rules speak for themselves.

10 Team Contributions

- **Alex Wang:** Led the JAX implementation, including the pure-functional step function, action masking, conflict resolution, observation representation, and integration with the training loop. Jointly prepared the poster and final analysis of symmetric versus asymmetric reward regimes.
- **Dario Soatto:** Led the MiniHedgemony environment design and managed evaluations of the self-play league experiments, round-robin evaluation, and reward-regime comparisons. Jointly prepared the poster and final analysis of symmetric versus asymmetric reward regimes.

Changes from Proposal. The project shifted from emphasizing broad behavioral comparison across many asymmetric reward functions to first establishing a stable self-play league and monotonic symmetric baseline. A major part of this shift was adding the 150 million-step heuristic cold start and adopting unequal role-specific budgets, with attack trained for 10 million environment steps per replicate and defense trained for 20 million environment steps per replicate. This adjustment was necessary because early feed-forward and vanilla PPO variants were unstable, and the asymmetric regimes did not reliably pass the monotonicity gate. As a result, more effort was allocated to

recurrence, KL-stabilized training, role-specific training budgets, and league evaluation before drawing conclusions about reward-induced behavioral clusters. Responsibility allocation did not change significantly.

11 AI Tools Disclosure

Core project ideas, environment design decisions, reinforcement-learning implementation, experiment execution, and analysis of results were developed independently by the project team. AI assistance was not used as a substitute for implementing the main reinforcement-learning algorithms or for generating experimental results. AI assistance from Claude was used for research and code debugging.

References

[1] Michael E. Linick, Jason Yurchak, Michael Spirtas, Stephen Dalzell, Yuna Huh Wong, and Yvonne K. Crane. *Hedgemony: A Game of Strategic Choices*. RAND Corporation, TL-301-OSD, 2020.

[2] Alexander Rutherford, Benjamin Ellis, Matteo Gallici, and others. JaxMARL: Multi-agent RL environments and algorithms in JAX. In *Advances in Neural Information Processing Systems*, 2024.

[3] Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 278–287, 1999.

[4] Sam Devlin and Daniel Kudenko. Theoretical considerations of potential-based reward shaping for multi-agent systems. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 225–232, 2011.

[5] Ido Greenberg, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Efficient risk-averse reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

[6] Chris Lu, Jakub Grudzien Kuba, Alistair Letcher, Luke Metz, Christian Schroeder de Witt, and Jakob Foerster. Discovered policy optimisation. In *Advances in Neural Information Processing Systems*, 2022.

[7] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, and others. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[8] Christopher Berner, Greg Brockman, Brooke Chan, and others. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[9] Johannes Heinrich and David Silver. Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121*, 2016.

[10] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

A Additional Implementation Details

The environment stores all game variables in a single JAX-compatible state structure. Before policies enter the self-play league, each role is trained for 150 million environment steps against a heuristic rule-based opponent to create a stable initial policy and avoid random-policy bootstrapping failures. Rewards and transitions are deterministic functions of the current state, joint action, and random key. Action masks are recomputed after every transition so that invalid actions are removed before policy sampling. The same step function is used during training and evaluation, reducing the chance of mismatch between rollout and tournament behavior.