An Adversarial Multi-Agent Pursuit Game for Locomotor Agility Pretraining

Nathan A. C. Hogg Matt Hyatt Neil Klingensmith George K. Thiruvathukal
Department of Computer Science, Loyola University Chicago
{nhogg, mhyatt, nklingensmith, gthiruvathukal}@luc.edu

Abstract: Agile humanoid locomotion requires controllers that can adapt to rapidly changing motion demands while remaining stable and robust. Recent locomotion controllers are typically designed with either a dense reward that is carefully shaped to encourage steps and penalize falling, or by using a reference tracking reward which teaches the controller to follow predetermined motions. Once a behavior that maximizes the reward is discovered, the controller ceases to learn. While generally stable, reference-tracking methods focus on motor control separately from motion generation. Self-play has been proven as an effective technique for superhuman performance on several multiagent games like Go, Starcraft, hide-and-seek, and capture-the-flag. We train neural networks to play the game of multiagent tag in a physics simulator to study if new motion generation and control strategies can emerge through sparse feedback and competitive self-play.

We present **MJTag**, a task suite for role-conditioned locomotion during competitive pursuit-evasion. Our model is a role-conditioned policy for a low-level locomotion controller and optional planar velocity commands. MJTag's league-based-play with sparse feedback enables a locomotion controller to adopt new strategies, including dodging, sprinting, and tackling.

Keywords: Locomotion, Multi-agent, Reinforcement Learning, Humanoid

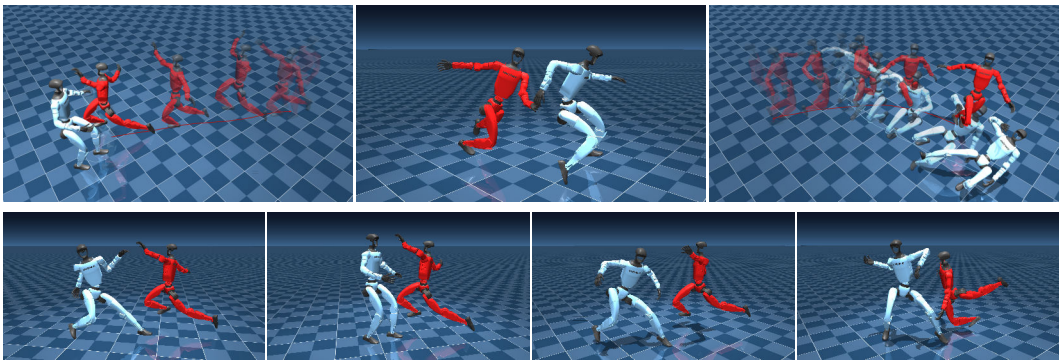


Figure 1: Simulated humanoid agents play tag: demonstrating lunging, diving, dodging and chasing.

1 Introduction

Agile robotic locomotion is essential for robots to be widely useful in real-world problems such as delivery, exploration, search and rescue, and healthcare. Current approaches often rely on supervised imitation of reference trajectories and reinforcement learning (RL) for high frequency control. While imitation learning provides stable training from prior demonstrations [1] , it depends on high-quality motion data. Such data is limited in scale, and it is highly dependent on the capture environment [2].

Often, motion capture must be post-processed to become physically plausible. A generative locomotion controller is limited to move only in ways that fall inside the training distribution.

In contrast, RL allows agents to learn from experience but often stagnates once a reward is maximized. Standard curriculum learning is frequently used to make the environment more difficult but the curriculum is seldom aware of the capabilities of the policy leading to impossible or trivial tasks [3]. This eventually dilutes the quality of training data. Advanced environment tuning results in a difficult high-dimensional optimization problem [4].

We hypothesize that the shortcomings of locomotor RL and motion generation are not due to any poor quality of locomotor skills but rather due to the eventual plateau of behavior when maximizing the reward function under the training environment. Additionally, the manual effort required to tune both the environment difficulty and reward shape are undesirable traits of RL.

Neither supervised learning nor dense RL offer a simple recipe for policy improvement that is resistant to plateau. This differs from RL on Atari-style games or manipulation tasks, where the reward being maximized is itself the desired behavior [5]. Conversely, terrain traversal is open-ended and any dense reward imposes a bias that is not guaranteed to be aligned with locomotion.

In competitive games, self-play is often used to enforce an adaptive curriculum. A weak model can begin to learn by dominating a weak opponent, and successive training rounds will prepare the policy to learn counter-strategies in order to receive a scalar reward [6, 7, 8, 9]. In this case, even though the earned reward is unchanged, the model still adapts since a skilled opponent will force the player into undesirable state-space which it will need to overcome.

Recently, self-play was used as a strategy in drone pursuit [10], demonstrating impressive low-level flight control at high frequency. Previously, tag has been used in quadrupedal motion [11] although their method has some notable differences to ours, with the primary difference being that their evader is a 2D dot coordinate rather than another identical robot entity. This makes their training protocol more similar to goal following than self-play. Moreover, they report that their model cannot learn from RL and they opt for evolutionary algorithms. Han et al. also train quadrupeds using a game of tag [12], but they freeze the locomotion controller and train a goal-proposal policy. This means that the game does not teach locomotor strategy. Instead, it teaches the high-level selection of planar coordinate goals. Similar to us, Han et al. are inspired by the parkour game World Chase Tag. Finally, Prioritized Fictitious Self-Play (PFSP) [6] has been used in the learning of character controllers [13] for boxing matches.

To address limitations in environment difficulty and reward modeling, we propose competitive tag as a surrogate objective that is well aligned with terrain-traversal. We hypothesize that the policy should continue to improve even on simple terrains since PFSP enforces adaptive difficulty.

Our contributions include (1) the development of a multiagent competitive league environment for humanoids, (2) a data sampling strategy to balance role-difficulty imbalances, and (3) a comparative analysis of state exploration under self-play and dense reward shaping. We demonstrate that competitive league-based play with sparse feedback can train a locomotion controller to develop strategic skills like lunging, tackling and ducking.

2 Multi-Agent Pursuit

This section outlines environment design, data sampling, task specification, and rewards. For all experiments, we train an MLP using PPO [14]. Contrary to previous work [11], our model is conditioned on its role, which allows the same model to perform pursuit and evasion.

The environment uses MuJoCo Warp via the `mjlab` [15] library. Physics runs at 200 Hz with 4 substeps and policy control at 50 Hz. The environment contains two Unitree G1 humanoids. Each humanoid has 29 actuators, for 58 total actuators in each simulated environment.

Figure 2: PFSP league training.

Opponent Sampling: We select opponents with Prioritized Fictitious Self-Play (PFSP) [6, 10]. Every 2,500 PPO epochs we add a frozen copy of the online model to a first-in, first-out league of 32 checkpoints, which also always contains a scripted policy that outputs zero actions, and serves as a passive baseline; Fig. 2. Following [6], each opponent is sampled with weight $f(p) = (1 - p)^2$, where p is the learner's empirical win rate against it, so opponent difficulty is measured from outcomes rather than assumed from training age.

Role Sampling: We use an online adaptive sampling mechanism to address the instability associated with role-difficulty imbalance: it is overwhelmingly easier for the models to play as the pursuer rather than the evader. Naively, this leads to training collapse and a lack of meaningful policy feedback. This imbalance is less present in other multiagent games where the objective is the same for both agents. We initialize the role sampler at $\rho = 0.5$ at the beginning of training and adjust the sampling weight after each round. If the evader win-rate (timeout rate) is less than 50%, we increase the role sampling weight by $\delta_\rho = 0.05$ and reduce the learner evader role probability if at least 50% of data rollouts time out. This gives the policy more evasion experience and keeps play competitive. Ablations show that without this sampling mechanism training is unstable. We present a side-by-side comparison that demonstrates how role sampling improves the balance of skill between the pursuer and evader in Appendix A.

Rewards: The policy receives a semi-sparse reward with weak dense shaping. Falls and arena exits terminate the episode and apply a -10 penalty to the offending agent. The environment considers an agent fallen if torso tilt is greater than 70° or pelvis height is below $0.5\mathrm{m}$, while leaving the arena triggers the out-of-bounds condition. Equation (1) shows this formulation:

$$r^i = w_{\text{fall}} * r^i_{\text{fall}} + r^i_{\text{lim}} + \sigma_i * \big(w_{\text{tag}} * r_{\text{tag}} + r_{\text{gap}}\big), \qquad \sigma_i = \begin{cases} +1 & i = \text{pursuer} \\ -1 & i = \text{evader} \end{cases} \tag{1}$$

Here $r^i_{\text{fall}}{=}{-1}$ if agent i falls or leaves the arena, ending the episode, and 0 otherwise; r^i_{lim} is a small joint-limit penalty; $r_{\text{tag}}{=}1$ when the agents come within $0.5\,\text{m}$ in the (x, y) plane, which also ends the episode; and $r_{\text{gap}}{=}{-}w_{\text{gap}}(d_t{-}d_{t-1})$, where d_t is their planar distance. σ_i ensures that role-based competition terms are treated as rewards or penalties for the respective roles. We use $w_{\text{fall}}{=}10$, $w_{\text{tag}}{=}5$, and $w_{\text{gap}}{=}0.001$.

3 Experimental Results

In our evaluation, the locomotion controller adopts interesting behaviors as it attempts to win the game of tag against competitors in the league. We also observe that Elo rises sharply in early training, and the league continues to produce new strategies after the rating levels off. Figure 3 shows the color-coded win rate of the locomotion controller against other competitors in the league. In the topmost row, the results show 100% success, indicating that all subsequent models learn to win against the untrained pursuer. The dark blue color indicates evader success. Similarly, all pursuers win against the first evader (red). In subsequent rounds, the online learner competes against other checkpointed competitors from the league. At $y = 19000$, the map becomes darker red, indicating

Figure 3: Tag win-rate between league matches: x-axis is the evader checkpoint, y-axis is the pursuer checkpoint.

Figure 4: Elo score of the online learner over 40,000 rounds of play (3.9 Billion env steps).

that the pursuer learns a new strategy that improves wins against previous evaders. Similarly from $x = [12000, 16000]$ and $[39000, 43000]$ the evader improves against previous pursuers.

We evaluate progress with Elo ratings [16, 17], computed purely from game outcomes with no human judgment. Each saved checkpoint plays 4096 deterministic games, one episode per parallel world from a fixed reset seed, split evenly among all earlier checkpoints, with roles assigned uniformly at random in each game (win $= 1$, draw $= 0.5$, loss $= 0$). A pair of checkpoints thus plays $4096/k$ games, where k is the number of earlier checkpoints. We fit a Bradley–Terry model to all outcomes jointly, adding one drawn pseudo-game per pair so that ratings stay finite under a clean sweep. Maximum-likelihood ratings are reported on the Elo scale ($400/\ln 10$ per logit), anchored at the untrained policy (0 Elo); lines show the mean across environments and the shaded band spans minimum to maximum.

Learned Agility: Figure 4 shows Elo rising sharply over the first $\sim$5,000 iterations and then holding near 1200. Because ratings are fit only against the league's own checkpoints, a flat curve means later checkpoints are not uniformly stronger than earlier ones; it does not mean behavior has stopped changing. Figure 3 shows distinct strategy shifts after this point: the pursuer at y=19k and the evader at x=39–43k each beat earlier opponents of the other role. We read this as the two roles trading advantages rather than settling into one fixed behavior. Qualitatively, the controller learns to lunge, dodge, and dive. We do not develop a formal agility measure; Elo measures strength relative to prior checkpoints only.

4 Conclusion

In our experiments, the controllers frequently exhibit unusual motions. Because the policy is not rewarded for tracking reference motions or staying near a behavioral prior [18], some of these may be effective exploits of the simulator's contact model rather than poor play. Areas for future work include adversarial terrain traversal subject to natural motion penalties and using self-play to explore efficient motions which still fall in-distribution of human motion generators.

MJTag applies competitive league-based training for RL in robotics, which incrementally incentivizes controllers to adopt new behaviors to win in a tournament. We show that tournament-style training continues to produce new locomotor strategies, such as lunging, dodging, and diving, from a sparse game outcome. Limitations are discussed in Appendix B.

Acknowledgments

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-23-F-0014 in the amount of $153,600.

This research was supported in part by a Research Support Grant administered through the Office of Research Services at Loyola University Chicago in the amount of $5000.

References

[1] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

[2] M. Xu, Y. Shi, K. Yin, and X. B. Peng. PARC: Physics-based Augmentation with Reinforcement Learning for Character Controllers, May 2025. URL http://arxiv.org/abs/2505.04002. arXiv:2505.04002 [cs].

[3] H. Kim, H. Oh, J. Park, Y. Kim, D. Youm, M. Jung, M. Lee, and J. Hwangbo. High-speed control and navigation for quadrupedal robots on complex and discrete terrain. *Science Robotics*, 10(102):eads6192, 2025.

[4] Z. Xie, H. Y. Ling, N. H. Kim, and M. van de Panne. Allsteps: Curriculum-driven learning of stepping stone skills. *Computer Graphics Forum*, 39(8):213–224, 2020. doi:https://doi.org/10.1111/cgf.14115. URL https://arxiv.org/abs/2005.04323.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning, 2013. URL https://arxiv.org/abs/1312.5602.

[6] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019. doi:10.1038/s41586-019-1724-z.

[7] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[8] B. Chen, S. Song, H. Lipson, and C. Vondrick. Visual hide and seek, 2019. URL https://arxiv.org/abs/1910.07882.

[9] Open Ended Learning Team, A. Stooke, A. Mahajan, C. Barros, C. Deck, J. Bauer, J. Sygnowski, M. Trebacz, M. Jaderberg, M. Mathieu, N. McAleese, N. Bradley-Schmieg, N. Wong, N. Porcel, R. Raileanu, S. Hughes-Fitt, V. Dalibard, and W. M. Czarnecki. Open-ended learning leads to generally capable agents. 2021. URL https://arxiv.org/abs/2107.12808.

[10] W. Tang, T. Chen, Z. Cui, B. An, J. Chen, R. Zhang, H. Liu, T. Wu, Q. Liao, F. Gao, Y. Wang, and C. Yu. Agilepe: Autonomous uav pursuit-evasion via self-play reinforcement learning, 2026. URL https://arxiv.org/abs/2608.14135.

[11] Y. Tang, J. Tan, and T. Harada. Learning agile locomotion via adversarial training, 2020. URL https://arxiv.org/abs/2008.00603.

[12] L. Han, Q. Zhu, J. Sheng, C. Zhang, T. Li, Y. Zhang, H. Zhang, Y. Liu, C. Zhou, R. Zhao, J. Li, Y. Zhang, R. Wang, W. Chi, X. Li, Y. Zhu, L. Xiang, X. Teng, and Z. Zhang. Lifelike agility and play in quadrupedal robots using reinforcement learning and generative pre-trained models. *Nature Machine Intelligence*, 6(7):787–798, 2024. ISSN 2522-5839. doi:10.1038/s42256-024-00861-3. URL http://dx.doi.org/10.1038/s42256-024-00861-3.

[13] Q. Zhu, H. Zhang, M. Lan, and L. Han. Neural categorical priors for physics-based character control. *ACM Trans. Graph.*, 42(6), dec 2023. ISSN 0730-0301. doi:10.1145/3618397. URL https://doi.org/10.1145/3618397.

[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[15] K. Zakka, Q. Liao, B. Yi, L. L. Lay, K. Sreenath, and P. Abbeel. mjlab: A lightweight framework for gpu-accelerated robot learning, 2026. URL https://arxiv.org/abs/2601.22074.

[16] A. E. Elo. The new USCF rating system. *Chess Life*, 16:160–161, 1961.

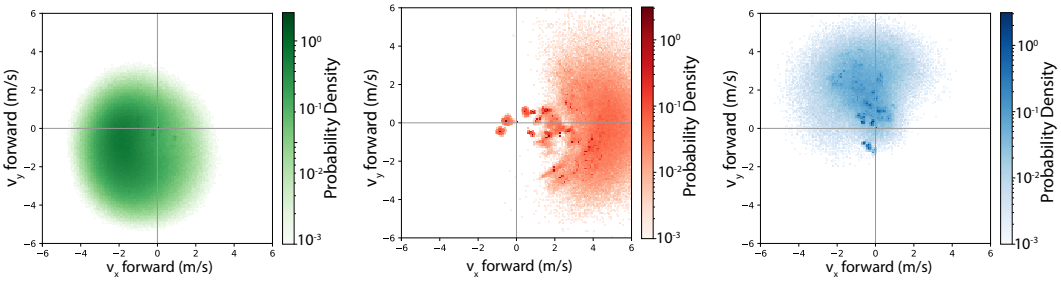
[17] S. Tang, Y. Wang, and C. Jin. Is elo rating reliable? a study under model misspecification, 2025. URL https://arxiv.org/abs/2502.10985.

[18] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang, L. Fan, and Y. Zhu. Hover: Versatile neural whole-body controller for humanoid robots, 2025. URL https://arxiv.org/abs/2410.21229.

A Appendix

Here we present additional evaluation to demonstrate the performance of adversarial play.

Learned Movement Characteristics: Figure 5a shows the probability density of planar velocity of a standard model trained with goal-tracking dense reward. This model did not participate in league play. Overall, the model does not initially exhibit a strong preference for moving in any particular direction. Figures 5b and 5c show the probability density function of the velocity vectors of the pursuer and the evader after participating in 43,000 rounds of league play. After training, the controller clearly has learned a strategy that is different as a result of conditioning the model on the pursuer role and evader role. Moreover, the policy continually seeks to push the boundary of its capability and changes its exploration strategy over the course of training.



(a) Dense Goal-following (b) Pursuer (c) Evader

Figure 5: Probability density of the velocity vector (a) for a naive dense reward without league play, and for the (b) pursuer and (c) evader after 40,000 rounds of tag.

Role Sampling: Figures 6 and 7 demonstrate how role sampling improves the balance of performance between pursuer and evader. Without role sampling (Figure 6), the pursuer's skill increases much faster than that of the evader, and after several hundred rounds wins consistently.

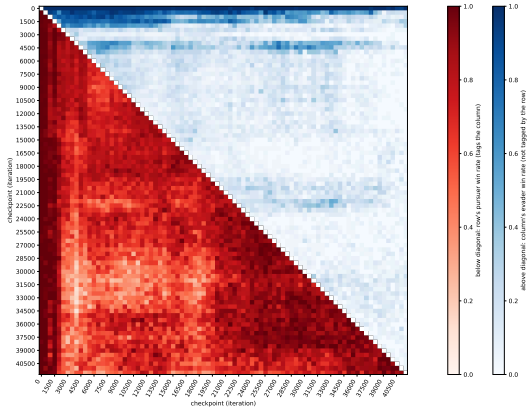


Figure 6: Win rate **without** role sampling.

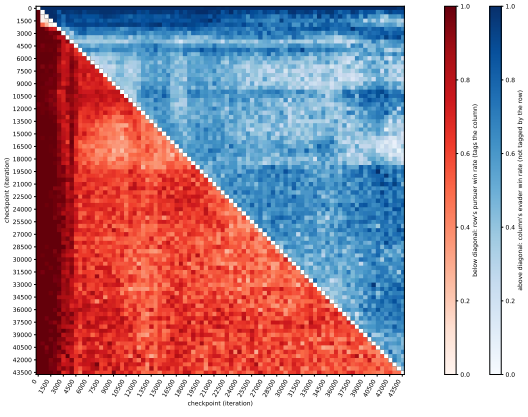


Figure 7: Win rate **with** role sampling.

B Limitations

Scope of evaluation. All results are in simulation on flat terrain, from a single training seed. We measure progress with Elo relative to the league's own checkpoints and do not yet compare against imitation-learning or dense-reward RL baselines on a common metric, nor do we define a formal measure of agility.

Task definition. A tag is defined by planar proximity (0.5 m) rather than physical contact. This may make the task easier than true tag and limits the role of deliberate contact skills such as tackling.

Scope of learned skills. Self-play rewards only skills that help win the game. Behaviors outside it, such as stair climbing, carrying loads, or a natural gait style, still require reward shaping or imitation. Because the policy is not regularized toward reference motions, it frequently produces unusual motions; some may be effective exploits of the simulator's contact model rather than poor play, and we have not yet separated the two.

Sim-to-real transfer. We do not deploy on hardware. Transfer would face safety risks from two robots colliding at speed, hardware wear from repeated falls, contact and friction mismatch during lunges and dives, and the need to estimate the opponent's state from onboard sensing.

Code and the MJTag environment are available at https://github.com/LUC-Robotics-Lab/tag