Combining Selfish Individual Rewards With Intrinsic Motivation for Synergistic Tasks

Gunjan Bhattarai
University of Texas at Austin
gunjan.bhattarai@utexas.edu

Amit Joshi
University of Texas at Austin
amitjoshi24@utexas.edu

Abstract—**In this work, we explore adding individualized, selfish extrinsic reward signals to see how they would impact agents in a multi-agent, sparse reward synergistic task with an intrinsic motivation reward. While prior work has mainly focused on getting agents to work together in multi-agent tasks where working together is essential to achieving a goal, scant, if any, attention has been placed upon the situation where an agent may have additional motives external to the synergistic task. For example, in soccer, agents have to work together in a team to score goals (the positive reward), but each agent can get a yellow (warning) or red (sending off) card for individually committed fouls, which may lead them to pursue actions that may not optimize the team's performance. To study this situation, we explore adding both an intrinsic, synergistic reward using a dynamics model as well as an individual extrinsic signal (either reward or penalty) on the Two-Arm Handover task. While time and computational limits only allowed us to run 1000 updates per experiment and thus prevented the robots from being able to complete the task, we hope that our work will prove useful for future researchers to explore this problem in the future.**

I. Introduction

Traditional reinforcement learning methods have been shown to be very effective in having individual agents learn a task both where it acts individually to maximize performance in an environment (Mnih et al., 2013) as well where it learns to maximize its performance against an adversary (Silver et al., 2016, Silver et al., 2017). However, such strategies do not translate very well in environments where multiple agents must engage in synergistic behavior, especially when positive rewards come by rarely (for example, having a team of robots work together to play baseball or soccer). This is partially because combining sparse rewards with multiple agents in a synergistic environment makes the action space enormous and difficult to explore effectively.

To solve this problem, the reinforcement learning community has generally employed intrinsic motivation (Schmidhuber 1991). One approach to intrinsic reward functions is curiosity, or having the environment provide agents a reward whenever they explore a state/action pair that is deemed worthwhile to explore. For example, "unseen" state/action pairs are worth exploring because they allow the agent to learn a policy that has breadth (Stadie et al., 2015). The hopes of such a strategy are that the agent will put greater emphasis on exploring the action space and thus have a higher probability of formulating policies emphasizing greater synergies with other agents. Conceptually, this can be seen as a more sophisticated version of epsilon-greedy exploration, where a random action is taken with ϵ probability, and the learned policy chooses the action with the other $1 - \epsilon$ probability. In this case, our goal is to increase ϵ to enable sample-efficient learning and maximize exploration of discovery of synergistic policies.

In general, using intrinsic motivation improves performance in synergistic environments, including robotic control tasks (Oudeyer et al., 2007) and Atari games (Bellemare et al., 2016, Pathak et al., 2017). Further enhancements have been proposed to intrinsic motivation, including using the discrepancy of predictions between a joint and a compositional prediction model (Chitnis et al., 2020).

However, to our knowledge, there have not been any experiments combining a positive, intrinsic reward framework with extrinsic, individual-based penalties. For example, while a game of soccer provides positive rewards for the whole team in the form of scored goals, there are also individual penalties for fouls (including the agent potentially being sent off from the game due to excessive fouling). Alternatively, in a corporate environment where team members must work together to finish a project, the team as a whole may receive positive reinforcement for the completion of the task, but each member could be held individually responsible and terminated from employment for their own shortcomings. In the latter situation, there is even a possibility of the extrinsic penalty being adversarial (i.e., in the form of selfish actions) itself, with individuals also competing for limited promotion spots to advance their careers. Ultimately, most prior work exploring synergistic behavior have generally abstracted away these situations, leaving open the question of how providing extrinsic rewards for selfish or adversarial actions affect the ability of intrinsic motivation to encourage synergistic behavior.

In this paper, we employ Robosuite (Zhu et al., 2020) to simulate the Two-Arm Handover task, using a two-layer deep Actor-Critic (Konda and Tsitsiklis, 1999) reinforcement learning algorithm, a two-layer deep dynamics model and a curiosity generator to provide intrinsic rewards, and an extrinsic penalty and reward function either penalizing or rewarding an agent for dropping the ball during the task. Our goal was to see how these individualized extrinsic signals would ultimately impact the model's overall performance. While heavy constraints on computational time and resources resulted in none of our agents being able to successfully complete the task and thus receive a reward (making the effectiveness of our proposed method undetermined), we nevertheless hope that our work will help future researchers with greater computational resources explore the problems we have outlined with our proposed methods.

II. Related Work

A. Modifying Intrinsic Motivation to Encourage Synergy

One alternative proposal to guide exploration and boost synergy has been to employ social motivation (Jaques et al., 2019). Specifically, this proposed reward function seeks to encourage agents to select actions that have the largest impact on the actions other agents select. LOLA (Foerster et al., 2018) takes a similar approach, although it does away with the emphasis on exploration and focuses entirely on having agents' policies being influenced by their impact on other agents.

Chitnis et al. (2020) questions whether exploration is even a good predictor of synergy in the first place, preferring to view synergistic actions are those where agents simultaneously acting together change the environment differently than if they acted sequentially. Based on this, they tested out two new intrinsic motivation functions: first, using compositional prediction error, and second, comparing prediction disparity between the result of actions of multiple agents taken jointly and sequentially (the latter having the advantage that the intrinsic reward function is now also analytically differentiable to the action, allowing for more informative gradient updates). Abe et al. (2021) also ends up doing away with the exploration focus entirely in optimizing the performance of agents in soccer, getting surprisingly strong results from choosing a simple intrinsic reward of +1 whenever each agent is involved in the ball for the first time. Our work builds upon these by including additional extrinsic reward signals that would theoretically impede intrinsic performance and seeks to evaluate the extent that this actually occurs.

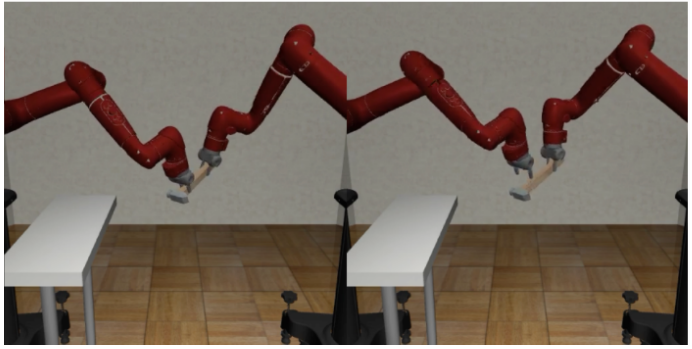
B. Multi-Agent Adversarial Reinforcement Learning

Numerous reinforcement learning problems can be expressed as tasks that are adversarial to some degree. A key example of this is games, where models such as DeepMind's AlphaZero (Silver et al., 2016) end up playing games of AlphaGo against itself to maximize its performance against its opponents in test time. BiCNet (Peng et al., 2017) takes this a step further by training multiple agents in a task where two teams attempt to beat each other in Starcraft combat games, thus needing both synergistic behavior with each member of the team and adversarial behavior with each member of the opposing team. Multi-agent adversarial learning can also be done in inverse reinforcement learning (Yu et al., 2019). Ultimately, while our selfish extrinsic reward signal can be fully adversarial like in these situations, we also consider extrinsic rewards that would still in theory leave some room for intrinsic synergistic learning to occur.

III. Data

For our experiments, we employed Robosuite's Two-Arm Handover simulation to collect data. In this environment, two arms attempt to move an object from one table to the other, and hand-off the object to the other arm in the air. This problem lends itself well to our research, as synergistic behavior is required for success (both arms meeting, and the second arm setting the object down on its table), and extrinsic selfish individual rewards which correspond to one arm dropping the object. The unselfish extrinsic reward is setting the hammer on the final table.

Fig. 1. Robosuite Two Arm Handover Problem (Zhu et al., 2020)



To form our observation space, we took the proprioception observations, arm joint positions (sine and cosine), arm joint velocities, effector pose, gripper finger positions, and gripper finger velocities of each robot as well as the hammer and object states. We concatenated these together to make two observation vectors (with each robot getting its specific observations as well as the common hammer and object states). While we did not make use of the environment-provided image data, we strongly recommend future work to do so. To update our Actor-Critic, dynamics, and rollout models, we had to postprocess the observation data to get a clipped version of our observation state, the mean of the observations, and the variance of the observations. In contrast, we simply used the action spec property of the environment without any modifications to provide each robot with its possible action space.

Due to time and computational constraints, we were only able to run 1000 simulation runs of 2 steps each (1 for each robot), although given the robots' inability to find a policy that completed the task in any of our experiments, we strongly recommend that future practitioners dramatically increase this to something more reasonable (e.g. the 1 million training steps used by OpenAI on their MuJoCo benchmarks or the 100,000-200,000 training steps that Chtinis et al., 2020 ran their experiments on).

IV. Methods

A. Base Reinforcement Learning Algorithm

We employed a deep Actor-Critic algorithm with 2-layer neural networks being used for both our policy and value functions. In this algorithm, the actor proposes an action and takes it, and the critic informs the actor how good the action was by computing the value function.

Algorithm 1 is a temporal difference learning policy-based reinforcement learning algorithm. The procedure will repeat until convergence (Karunakaran 2020).

Algorithm 1 actor-critic

```
procedure ACTOR-CRITIC($s_t$)    ▷ Takes in current state
    $a_t \leftarrow$ sample from actor's policy $\pi_\theta$
    $A_{\pi_\theta}(s_t, a_t) = r(s_t, a_t) + V_{\pi_\theta}(s_{t+1}) - V_{\pi_\theta}(s_t)$    ▷ advantage function
    $\nabla J(\theta) \approx \nabla_\theta log \pi_\theta(a_t, s_t) A_{\pi_\theta}(s_t, a_t)$
    $\theta = \theta + \alpha \nabla J(\theta)$    ▷ update policy parameters
    $w = w + \alpha A_{\pi_\theta}(s_t, a_t)$    ▷ update critic weights
end procedure
```

Our actor critic algorithm was comprised of a policy and value function, both of which were comprised of two-layer neural networks. Each of these employed ReLU for the first activation function and Tanh for the second activation function, with the only difference being that the output dimension of the policy function was 64 while it was 1 for the value function. All neural networks were trained with the Adam optimizer (Kingma and Ba, 2014) with linear decay. Our policy learning rate was set to 1e-4, our value function learning rate was set to 3e-4, and dynamics learning rate was set to 3e-4.

Other hyperparameter choices for our policy gradient algorithm include our selection of the clipping parameter to be 0.2, the entropy coefficient to be 0.01, and the dynamics coefficient to be 0.5. Finally, we employed rollout in order to simulate possible simulations from our current state due to the need to model uncertainty in a robotics environment, although we did not employ a supervised learning method to do so due to concerns over computational cost.

Our intrinsic reward function was similar to Chitnis et al. (2020) introduced - each agent had a 3 layer deep neural network (activation functions ReLU, ReLU, and Tanh, respectively) dynamics model that takes in the current environment state ($\in \mathcal{S}^{\text{env}}$), the current agent state ($\in \mathcal{S}^{\text{agent}}$), and an action ($\in \mathcal{A}^{\text{agent}}$). If we call one agent A and the other agent B, we denote the dynamics model as $f^A : \mathcal{S}^{\text{env}} \times \mathcal{S}^{\text{A}} \times \mathcal{A}^{\text{A}} \rightarrow \mathcal{S}^{\text{env}}$ (resp. f^B). Consider the composition of f^A and f^B:

$$f^{\text{composed}}(s, a) = f^B(f^A(s^{\text{env}}, s^A, a^A), s^B, a^B)$$

We now give our intrinsic reward function, where s is initial state, $a = (a^A, b^B)$ is the tuple of actions taken by both agents, and s' is the next state. Intuitively, for synergistic actions $r^{\text{intrinsic}}$ is likely to be high, as f^{composed} will predict the environment if the actions are taken sequentially, which would be different from the environment resulting from the actions taken jointly, s'.

$$r^{\text{intrinsic}}(s, a, s') = ||s' - f^{\text{composed}}(s, a)||_2^2$$

Alternative methods we considered to compute intrinsic reward included calculating the squared distance between the dynamics model's predicted output p^t an actual environment y^t at each step, but we decided against this due to Chitnis et al. (2020)'s empirical demonstration that this was a suboptimal formulation of intrinsic reward.

Our individualized selfish extrinsic reward was a constant value (either +1 or -1) provided for an action under certain conditions. These conditions are if an agent was the one to perform the action (out of all other agents), and making sure the action did not lead to a unselfish extrinsic reward. Specifically for the Two Arm Handover problem, we defined a selfish reward for one arm dropping the object (hammer) on the floor. The unselfish extrinsic reward was setting the hammer on the final table. This reward would be added to the unselfish extrinsic reward for the agent that received it, with the combined extrinsic reward being backpropagated through the Actor-Critic neural network.

B. Code Implementation

We based a significant portion of our code on Michaux and Qing (2018)'s intrinsic motivation implementation. However, because their experiments involved training an individual agent to learn OpenAI's FetchPushv1 simulation task with the Gym library (Brockman et al., 2016), we still had to make significant modifcations to conduct our experimentation. We expanded their code to include support for two agents, rewriting the intrinsic motivation function to more closely follow Chitnis et al. (2020)'s work, being able to take as input Robosuite's data formats, directly included relevant source code from OpenAI's Baselines library (Dhariwal et al., 2017) to postprocess our observation data due to the methods the repository's authors used being deprecated, added our extrinsic reward/penalty from dropping the object, and myriad additional modifications that were necessary to get the base code to compile and run successfully. We made use of PyTorch (Paszke et al., 2019) to write our code and used Google Colab to finalize and train our models. Because the deep neural networks we employed were relatively shallow, the primary bottleneck of our implementation was waiting for the Robosuite environment to return the specific environment observations and rewards. As such, we ended up running our experiments on CPU, using a 2.2 GhZ single core Intel Xeon CPU on Google Colab.

V. Experiments

We ran eight experiments in total, with four experiments having an episode length of 2000 steps (1000 updates, 1 step per agent) and the other four having an episode length of 500 steps (thereby resulting in us having four different episodes of training). The four experiments for each of these cases included using only the unselfish extrinsic reward; using the unselfish extrinsic reward and the intrinsic reward; using the unselfish extrinsic reward, the intrinsic reward, and a selfish extrinsic reward of -1 (where the agent is penalized for dropping the object); and using the unselfish extrinsic reward, the intrinsic reward, and a selfish extrinsic reward of 1 (where the agent is rewarded for dropping the object).

The reason for the two selfish extrinsic reward cases is to explore a situation where one agent is motivated to act directly adversarial to the task itself, whereas in the other, the robot merely has the chance of being discouraged to move the object it all in hopes of not getting a negative reward. The latter reward strategy still can work with synergy, though, because not dropping the object is essential to completing the actual task. We used two Panda robots for all experimentation.

Of the four different types of experiments across a given episode framework, two of them (no intrinsic or selfish extrinsic, intrinsic but no selfish extrinsic) were designed to be baselines to compare the results of the experiments with agents motivated by selfish extrinsic reward signals with.

Unfortunately, due to the heavy computational limitations of our experiments, our agents were neither able to complete the task during any episode nor were even able to find the hammer in the first place (we are also open to the possibility that some undetected error in our code formulation may have hampered results as well, although we did not see any evidence to support this conclusion in our training runs). Thus, neither the positive nor the negative selfish extrinsic reward was able to make an impact on the behavior of either agent, leaving the questions we had formulated regarding selfish extrinsic reward signals with an indeterminate answer. The intrinsic rewards ended up being the only way that we could influence the robots' behavior, although the value function loss in robots with intrinsic motivation only ended up exponentially increasing across the 1000 updates due to the robots' inability to discover a goal-based, extrinsic reward. In contrast, the value function loss steadily decreased when no intrinsic motivation was present.

Reward Type	1	2	3	4
Extrinsic/Maximum	0	0	0	0
Extrinsic/Mean	0	0	0	0
Extrinsic/Median	0	0	0	0
Extrinsic/Minimum	0	0	0	0
Intrinsic/Maximum	0	0.0264	0.0388	0.0426
Intrinsic/Mean	0	0.0257	0.0343	0.0397
Intrinsic/Median	0	0.0257	0.0343	0.0397
Intrinsic/Minimum	0	0.025	0.0298	0.0368
Selfish Extrinsic/Maximum	0	0	0	0
Selfish Extrinsic/Mean	0	0	0	0
Selfish Extrinsic/Median	0	0	0	0
Selfish Extrinsic/Minimum	0	0	0	0

Table 1: Unselfish extrinsic, intrinsic, and extrinsic reward data at the 1000th update for each experiment with episode length of 1000 updates of 2 steps each. Experiment 1 used only the unselfish extrinsic reward; Experiment 2 used only the unselfish extrinsic reward and the intrinsic reward; Experiment 3 used the unselfish extrinsic reward, the intrinsic reward, and a selfish extrinsic reward of -1; and Experiment 4 used the unselfish extrinsic reward, the intrinsic reward, and a selfish extrinsic reward of 1.

Reward Type	1	2	3	4
Extrinsic/Maximum	0	0	0	0
Extrinsic/Mean	0	0	0	0
Extrinsic/Median	0	0	0	0
Extrinsic/Minimum	0	0	0	0
Intrinsic/Maximum	0	0.0243	0.0245	0.0393
Intrinsic/Mean	0	0.0241	0.0227	0.0329
Intrinsic/Median	0	0.0241	0.0227	0.0329
Intrinsic/Minimum	0	0.0239	0.0208	0.0266
Selfish Extrinsic/Maximum	0	0	0	0
Selfish Extrinsic/Mean	0	0	0	0
Selfish Extrinsic/Median	0	0	0	0
Selfish Extrinsic/Minimum	0	0	0	0

Table 2: Unselfish extrinsic, intrinsic, and extrinsic reward data at the 1000th update for each experiment with episode length of 250 updates of 2 steps each. Experiment 1 used only the unselfish extrinsic reward; Experiment 2 used only the unselfish extrinsic reward and the intrinsic reward; Experiment 3 used the unselfish extrinsic reward, the intrinsic reward, and a selfish extrinsic reward of -1; and Experiment 4 used the unselfish extrinsic reward, the intrinsic reward, and a selfish extrinsic reward of 1.

Loss	1	2	3	4
Loss/DeltaPi	-2.8e-5	6.54e-5	-2.5e-4	5.4e-5
Loss/DeltaV	-9.3e-10	-0.00977	-0.0244	-0.084
Loss/Dynamics	0	3.9e-5	1.9e-5	7.5e-05
Loss/Entropy	464	23.2	23.6	23.8
Loss/KL	228	8.97	22.2	10
Loss/Policy	0.024	-0.124	-0.113	-0.0446

Table 3: Data relating to the value of various loss functions

at the 1000th update for each experiment with episode length of 1000 updates of 2 steps each. Experiment 1 used only the unselfish extrinsic reward; Experiment 2 used only the unselfish extrinsic reward and the intrinsic reward; Experiment 3 used the unselfish extrinsic reward, the intrinsic reward, and a selfish extrinsic reward of -1; and Experiment 4 used the unselfish extrinsic reward, the intrinsic reward, and a selfish extrinsic reward of 1.

Loss	1	2	3	4
Loss/DeltaPi	-3.6e-5	1.34e-3	-1.8e-4	6.3e-4
Loss/DeltaV	2.2e-10	-0.00391	-0.00488	-0.0112
Loss/Dynamics	0	9.3e-5	8.97e-5	1.1e-4
Loss/Entropy	464	22.8	23	24.7
Loss/KL	228	6.85	10.7	11.9
Loss/Policy	0.0261	-0.116	-0.104	-0.129

Table 4: Data relating to the value of various loss functions at the 1000th update for each experiment with episode length of 250 updates of 2 steps each. Experiment 1 used only the unselfish extrinsic reward; Experiment 2 used only the unselfish extrinsic reward and the intrinsic reward; Experiment 3 used the unselfish extrinsic reward, the intrinsic reward, and a selfish extrinsic reward of -1; and Experiment 4 used the unselfish extrinsic reward, the intrinsic reward, and a selfish extrinsic reward of 1.

VI. Discussion and Future Work

The first and most obvious step to expand upon our work would be to run our experiments for significantly longer. For perspective, Chitnis et al. (2020) did not end up getting a task completion rate greater than 0 until it ran for at least 10,000 steps on Mujoco's bottle task and 20,000 steps on Mujoco's ant push task with a 4-layer neural network for the policy, value, and dynamics functions. However, the authors also hand designed a set of specific features to aid the system in completing the desired tasks, something that we were unable to do. We recommend that future researchers look into better hand design of features or by using demonstration to teach the robots the features themselves.

After that step, there are numerous avenues to take this project further. We can take advantage of the image data and concatenate a representation of it to the other observations we made us of. This can either be done by making use of a convolutional or Vision Transformer (Dosovitskiy et al., 2020) feature encoder or by making use of an implicit neural representation. In the latter case, SIREN (Sitzmann et al., 2020) is likely a good starting point to encode robotic perception data. In addition, this can also be set up as a multitask learning problem similar to GIGA (Jiang et al., 2021). Please note, however, that taking such steps would make the use of a GPU essential to have reasonable training speeds, especially considering the relatively large model sizes of modern robot perception and computer vision models.

In addition, given that our reinforcement learning neural networks were relatively shallow (2-3 layers), using more sophisticated (for example, deeper) architectures might be a good avenue to explore. However, it should be noted that these steps should be taken in tandem with increasing computational power and access to data, something we were largely unable to do.

We trained each agent's dynamics model by using the agent's action and the observed states before and after. As each agent had its own dynamics model, we do not want the other agents to interfere in this training. Based on this, we could improve upon our current training of the dynamics model by simply ignoring synergistic datapoints, where the other agent had a vital part in influencing the ”after” state. This could be accomplished by looking at the intrinsic reward of a particular step.

Finally, there are numerous avenues for additional extrinsic reward signals designed to impede synergistic intrinsic learning. For example, we can penalize an agent if it interferes with another agent (e.g. if B is in A's way and thus blocks A from completing its task). This could be accomplished by having a network that predicts environments assuming other agents do not exist. If the true environment is different than what A predicted, and A would have received a reward, then penalize B. Alternatively, it would be of interest to explore using intrinsic rewards (both with and without the adversarial extrinsic reward) in a situation where synergy would actually hurt overall performance. Lastly, it would be of interest to see how our methods would perform on other robotic domains (e.g. Two-Arm Peg In Hole), with other robots (e.g. Sawyer, Jaco), or with greater than 2 agents.

VII. Conclusion

In this work, we studied the impact of adding a selfish extrinsic reward signal to a synergistic intrinsic motivation reward in the Two Arm Handover problem. While heavy computational limitations made our hypothesis and evaluation of the effectiveness of our proposed methods indeterminate, we are optimistic that future work can take off from where we have left off and provide a definitive answer.

VIII. Acknowledgements

We express our thanks to our professor Dr. Yuke Zhu and our teaching assistant Zhenyu Jiang for introducing us to much of the relevant literature needed to complete this project and for his wonderful lectures that we enjoyed and learned from throughout the semester as well as to the ARISE Initiative and the UT Austin Robot Perception and Learning Lab for building the Robosuite library. We would also like to thank our classmates for the student presentations that we enjoyed and learned from throughout the semester. We also express our thanks to our friend Jeffrey Wang, who helped

us visualize our results using Tensorboard. Lastly, for those interested in running our experiments in the computational environment needed to better evaluate our methods, we have attached our code here: https://github.com/amitjoshi24/selfish-rewards-with-intrinsic-motivation

IX. Bibliography

[1] Takashi Abe, Ryohei Orihara, Yuichi Sei, Yasuyuki Tahara, and Akihiko Ohsuga. Acquisition of Cooperative Behavior in a Soccer Task Using Reward Shaping. In *ACM Digital Library*, 2021.

[2] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang and Wojciech Zaremba. OpenAI Gym *arXiv preprint arXiv:1606.01540*, pp. 1471–1479, 2016.

[4] Rohan Chitnis, Shubham Tulsiani, Saurabh Gupta, Abhinav Gupta. Intrinsic Motivation for Encouraging Synergistic Behavior *ICLR*, 2021.

[5] Dhariwal, Prafulla and Hesse, Christopher and Klimov, Oleg and Nichol, Alex and Plappert, Matthias and Radford, Alec and Schulman, John and Sidor, Szymon and Wu, Yuhuai and Zhokhov, and Peter. OpenAI Baselines. In *Github*, 2017.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, M. Dehghani, Matthias Minderer, Georg Heigold, S. Gelly, Jakob Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[7] Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 122–130. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[8] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro A Ortega, DJ Strouse, Joel Z Leibo, and Nando de Freitas. Intrinsic social motivation via causal influence in multi-agent RL. In *ICML*, 2019.

[9] Jon Michaux and Yifeng Qing Intrinsic Motivation for Robotic Exploration. In *Github*, 2018.

[10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller Playing Atari with Deep Reinforcement Learning. In *arXiv preprint arXiv:1312.5602*, 2013.

[11] Vijay R Konda, John Tsitsiklis. Actor-critic algorithms. In *Solla et al. (1999)*, pages 1008–1014.

[12] Zhenyu Jiang, Yifeng Zhu, Maxwell Svetlik, Kuan Fang, Yuke Zhu. Synergies between affordance and geometry: 6-dof grasp detection via implicit representations. *arXiv preprint arXiv:2104.01542*, 2021.

[13] Dhanoop Karunakaran. The Actor-Critic Reinforcement Learning algorithm *Medium*, 2020.

[14] Diederik P. Kingma, Jimmy Ba Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library *arXiv preprint arXiv:1912.01703*, 2017.

[17] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.

[18] Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, Jun Wang. Multiagent Bidirectionally-Coordinated Nets: Emergence of Human-level Coordination in Learning to Play StarCraft Combat Games *arXiv preprint arXiv:1703.10069*, 2017.

[19] Jurgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proceedings of the international conference on simulation of adaptive behavior: From animals to animats*, pp. 222–227, 1991.

[20] Vincent Sitzmann, Julien NP Martel, Alexander W Bergman, David B Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *arXiv preprint arXiv:2006.09661*, 2020.

[21] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel and Demis Hassabis Mastering the game of Go with deep neural networks and tree search. In *Nature*, 2016.

[22] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering Chess and Shogi by Self-Play with

a General Reinforcement Learning Algorithm *arXiv preprint arXiv:1712.01815*, 2017.

[23] Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

[24] Yuke Zhu, Josiah Wong, Ajay Mandlekar, and Roberto Martín-Martín. robosuite: A modular simulation framework and benchmark for robot learning. *https://arxiv.org/pdf/2009.12293.pdf*, 2020.

X. Graphs

We present the value function loss graphs for 3 experiments, each run for 1000 steps. We also thought it would be of interest to show the mean intrinsic reward graphs.

Fig. 2. Baseline: No Intrinsic Motivation and No Selfish Extrinsic Rewards

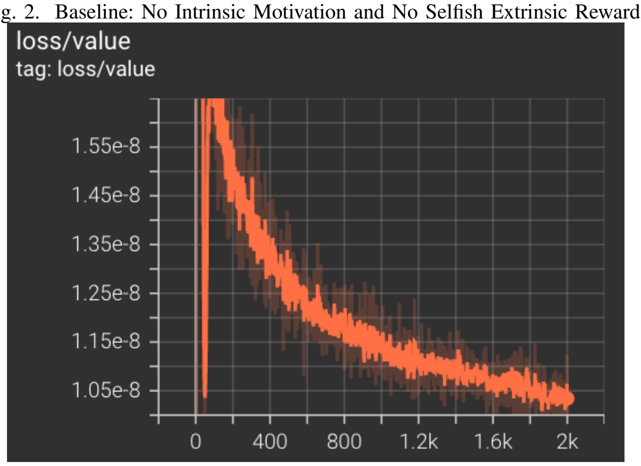


Fig. 3. Baseline: Intrinsic Motivation but No Selfish Extrinsic Rewards

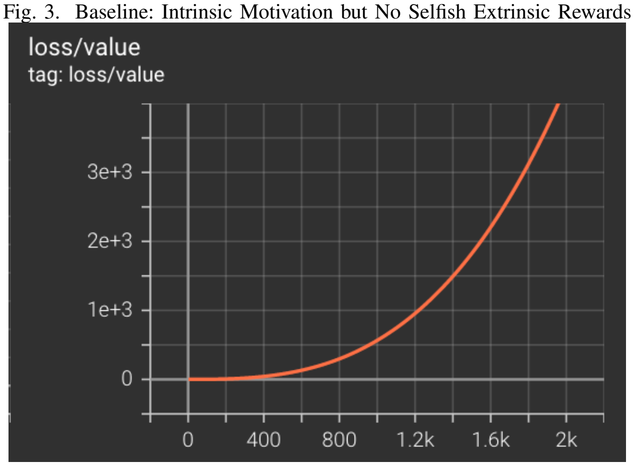


Fig. 4. Intrinsic Motivation and Selfish Extrinsic Rewards (+1)

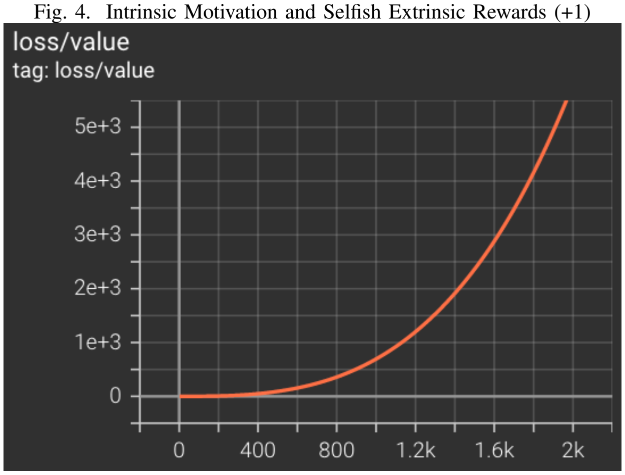


Fig. 5. Mean Intrinsic Reward. Experiment: Intrinsic Motivation but No Selfish Extrinsic Rewards

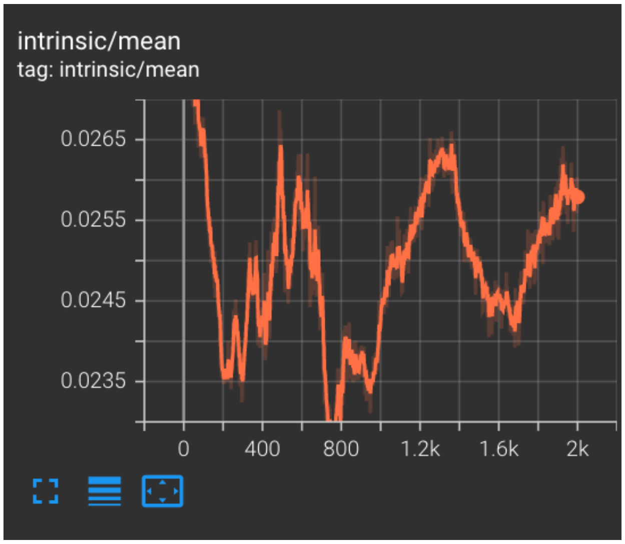
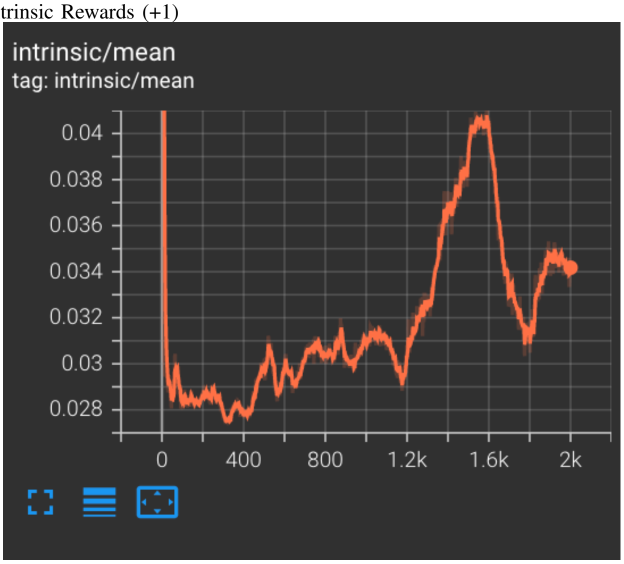


Fig. 6. Mean Intrinsic Reward. Experiment: Intrinsic Motivation and Selfish Extrinsic Rewards (+1)


intrinsic/mean
tag: intrinsic/mean
1.2k
1.6k
2k