Reconstruction-Driven Curiosity

Alex Chen*
University of Michigan
lubicz@umich.edu

Chris Hoang*
University of Michigan
choang@umich.edu

Thomas Huang*
University of Michigan
thomaseh@umich.edu

Osama Saeed*
University of Michigan
saeedosa@umich.edu

Jinxin Zhou*
University of Michigan
jinxinz@umich.edu

Abstract

The concept of intrinsic rewards has long been rooted in biological and psychological fields, and has only recently manifested itself in the study of Reinforcement Learning. Previous works have sought to capture the notion of curiosity by crafting intrinsic reward signals. In this paper, we introduce reconstruction-driven curiosity (RC), another form of curiosity based on the reconstruction of states. We define curiosity as the prediction error between the encoded features of the state and the reconstruction state. In theory, this method should be able to recreate states similar to those it has seen before (low intrinsic reward), but struggle with novel states (high intrinsic reward). To validate our idea and understand the intuition, we use the MNIST dataset to conduct toy experiments. We then evaluate RC on two challenging Atari games, Montezuma's Revenge and Venture. We demonstrate that our method is competitive with previous curiosity-based methods.

1. Introduction

Reinforcement learning (RL) methods have experienced success in producing agents that can match or even surpass human-level performance on a variety of tasks such as Atari games [5] and Go [7]. The success of traditional RL algorithms on these tasks depends on the presence of dense reward signals to guide agents to take the right actions towards the completion of the expected tasks. However, the requirement of well-designed reward feedback constrains the range of problems that may be solved by RL. One reason is that it may be expensive and time-consuming for humans to annotate new tasks. In some situations, it is difficult or even impossible to craft well-shaped reward signals due to the complexities of the real world. Thus, researchers have begun to dive into the challenge of designing RL methods that can succeed without these dense extrinsic rewards [1].

Figure 1: Our agent performing a difficult maneuver in Montezuma's Revenge.

One natural way to alleviate this problem is to reward the agent for behavior that is likely to result in achievement of the external task or mastery of skills. For example, babies are very curious and often explore new objects and areas even without their parents' external

[*] Equal contribution.

supervision. This inner motivation helps them explore and understand more about their surrounding environment, which may be useful for more concrete tasks such as walking or finding toys. These types of rewards signals are called intrinsic rewards and previous works have crafted different intrinsic rewards for capturing notions of curiosity.

In this paper, we present reconstruction-driven curiosity (RC), a method which aims to estimate the novelty of a state based on the agent's visual *understanding* of that state. We loosely define visual understanding as the agent's ability to compress a state, given as pixel input, into a low-dimensional latent representation and to then create a pixel reconstruction of the state from its latent representation. The agent is then rewarded for reaching states which it is unable to reconstruct well. We hypothesize that the agent will struggle to reconstruct states that it has not encountered often in its interaction experience and conversely, will be able to reconstruct states that are similar to states it has seen before. Reconstruction intuitively captures understanding since it involves understanding what features are important to remember when compressing the state.

To provide some necessary background, we describe in detail the relevant previous works in curiosity-based learning, and how our proposed methods intend to build on them. We then go on to give a concrete presentation of RC and the framework involved to incorporate the overall method into the training of the agent's behavior policy, as well as expanding on the alternate formulations that we experimented and tested with. Using a variety of frameworks, including the MNIST data set and the Atari test bed, we present the results of our methods and demonstrate the viability of RC as an intrinsic reward mechanism. We find that RC is a valid measure of curiosity and allows us to have promising initial results, comparable to previous works. Additionally, we argue that our reward is more intuitive to understand since it requires understanding of the current state.

2. Related Work

Previous work introduced the concept of surprisal, an exploration reward dependent on the prediction ability of a learned forward dynamics model [1]. With this intrinsic reward, the agent wishes to experience state transitions where the model performs poorly due to the lack of time spent training in those areas of the environment. Their experiments across 54 environments demonstrate that agents trained with only the surprisal reward are able to explore their environment, learn skills, and perform well with regards to the extrinsic reward signal. However, as the authors note, this method may be limited by the *noisy-TV problem*, where the agent instead undesirably seeks out transitions with high entropy despite rather than actually novel transitions. This is mostly due to the reliance on predicting the next state, which is extremely difficult to do if there are random transitions in state. In our work we evaluate the intrinsic reward using only the current state, which decreases random transitions' impact.

Most closely related to our work is random network distillation (RND) [2]. This paper describes novelty as the ability for a predictor network to predict the features of an observation given by a fixed and randomly initialized target network. The prediction error for a particular observation is expected to be higher for states that have been seen less frequently by the predictor network. This method seems to fare especially well in sparse environments, such as Montezuma's Revenge. It also avoid the *noisy-TV problem* because it does not update the policy based on ability to predict the next state, but rather on the ability to predict the current state. However, we find that predicting how a random network's output seems unintuitive to how we as humans evaluate understanding of state.

Our work takes inspiration from this set of related work, and develops a new curiosity-related reward, one focused on being able to compress an image in a way that allows accurate reconstruction. We find that this method is more intuitive with respect to visual understanding, as it demonstrates ability to encode raw images into compact representations that still maintain sufficient information for proper reconstruction.

3. Reconstruction-Driven Curiosity

In this section, we first describe the intuition behind our method and then go into the details of the formulation.

3.1. Intuition

Reconstruction-driven curiosity (RC) is an intrinsic reward based on the performance of an auxiliary reconstruction module that is simultaneously trained as the agent learns a behavior policy. The intuition behind RC is that the agent's reconstruction module will perform better on states similar to those it has visited in the past and conversely, it will perform worse on states dissimilar to those it has visited in the past, which we assume correspond to novel states. The agent is given an intrinsic reward proportional to the reconstruction error and is thereby encouraged to visit novel states. As the agent visits these novel states, the reconstruction module will begin to train on those states until it can reconstruct them relatively well. Those states will no longer be considered as novel and the process will then repeat where the agent will again be encouraged to visit newer states. This formulation captures the idea of understanding of the state, where rewards are given for reaching states that the agent does not understand, and not given when the agent understands the state they are in.

3.2. Formulation

More concretely, the reconstruction module is formulated as an autoencoder system with an encoder network $\phi : S \rightarrow \mathbb{R}^d$ and a decoder network $\psi : \mathbb{R}^d \rightarrow S$, where S is the state space and $\mathbb{R}^d$ is the embedding space [4]. The operation of the autoencoder may be interpreted as the ability to parse and compress visual information from the state into latent features that are most informative for reconstructing that state. Our method is generally independent of the type of autoencoder used, and we explore different possibilities in Section 4.2. We define the RC reward signal as the mean squared error (MSE) between the original state and reconstructed state in the *embedding* space, which is mapped using the same encoder network. This is shown below in equation 1.

$$r_{RC}(s) = ||\phi(s) - \phi(\psi(\phi(s)))||_2^2 \qquad (1)$$

The RC reward signal is computed in the embedding space to estimate state novelty in terms of latent features which capture the important aspects of the state. In this manner, we also avoid noise that may arise from inconsequential errors in reconstructing raw pixels of the state. Similar to [2], we normalize the reward signal by dividing it by the running standard deviation in order to avoid problems with generalizability across time and across different environments.

The parameters of the autoencoder are trained via gradient descent using with the MSE loss between the pixels of the original state and the reconstructed state. Because initial reconstructions will be very poor and not indicative of state novelty, we pretrain the autoencoder for a small number of iterations before conducting policy optimization. Afterwards, the autoencoder will continue to train as the agent learns an action policy through policy optimization, which is achieved via the proximal policy optimization algorithm [6].

The total reward given to the agent is a linear combination of the RC intrinsic reward and the extrinsic reward as formulated likewise in RND [2]. Equation 2 explicitly gives the total reward for a transition at time t, where r_E is the extrinsic reward and α is the coefficient for the RC reward.

$$r(s_t, a_t, s_{t+1}) = \alpha \cdot r_{RC}(s_{t+1}) + r_E(s_t, a_t, s_{t+1}) \qquad (2)$$

The full architecture of the RC module and how it fits into the reinforcement learning pipeline are depicted in Figure 2.

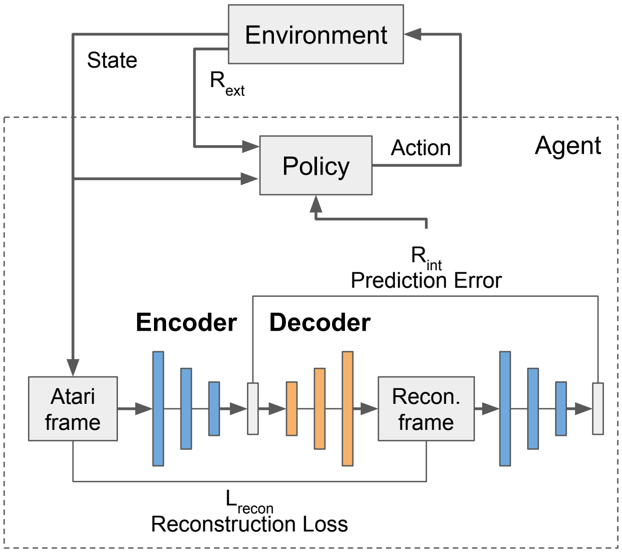

Figure 2: Full pipeline with RC module and policy optimization.

4. Experiments

We conduct several experiments on the MNIST dataset and two Atari games to evaluate the effectiveness of our proposed RC method.

4.1. MNIST Toy Model

To validate our idea and illustrate the intuition behind it, we conducted toy experiments using the MNIST dataset similar to those conducted in RND [2]. We denote 0s to represent states we have seen before, and a target digit class to represent more novel states. We then train a reconstruction module with different proportions of 0s and the target digit while maintaining the same overall number of training samples.

We use a traditional VAE trained on data splits which each contain increasing proportions of the target digit class. We then evaluate the ability of the VAE to reconstruct the target class. The results shown in Figure 3 depict how the reconstruction loss scores decrease when the VAE sees more of the target digit class during training. From these results, we begin validating our hypothesis that completely novel observations will give very high intrinsic rewards, and that previously seen observations will slowly decrease in reward as they are more frequently seen.

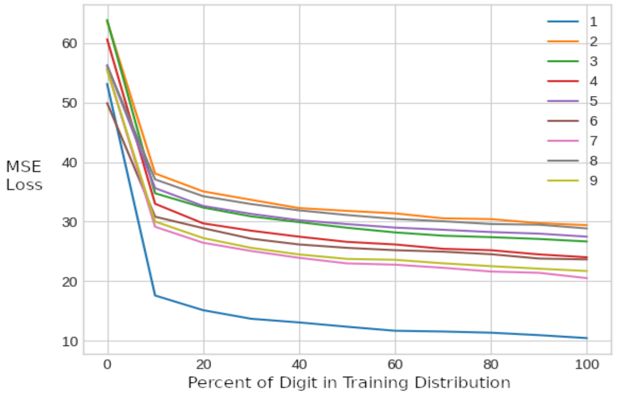

Figure 3: Loss for each digit as a function of percent split between target digit and zero.

4.2. Reconstruction Network Variants

We hypothesize that generative models could learn useful features of the input observations and could lead to more meaningful intrinsic rewards. In order to validate our hypothesis, we use three different kinds of architectures for the reconstruction module: the standard autoencoder, the variational autoencoder (VAE), and a variant of VAE with adversarial loss (VAE-GAN). Since the VAE constrains the latent space to fit a multivariate Gaussian distribution, details in the input frames, such as the player and the enemy, may be ignored, so the reconstruction may not be useful as an intrinsic reward. The VAE-GAN utilizes adversarial loss, which may encourage the model to focus on these details and alleviate the aforementioned issue. The VAE-GAN architecture is shown in Figure 4. We evaluate the reconstruction performance of these architectures to see whether they can successfully overfit to the training data and be used to compute the intrinsic reward.

Qualitative results for this experiment are shown in Figure 5. In the top row of the figure, we see that the standard autoencoder architecture is able to successfully reconstruct the input frame. We can see that the positions of the player and the enemy are correctly captured as well. In the second row, we see that the VAE is not able to reconstruct the fine details of the input frame, and the positions of the player and the enemy are blurred. This is likely due to the issue mentioned previously. In the last row, we see that adding adversarial loss did not solve the issue. The VAE-GAN also cannot capture the precise location of the characters, and the reconstruction result is noisy. Since we want our reconstruction network to overfit to the input frames, we decided to stick to the standard autoencoder architecture.

4.3. Montezuma's Revenge and Venture

Next, we evaluate our proposed RC module on two Atari games, Montezuma's Revenge and Venture. These games are known to be very difficult games due to the sparsity of the rewards, which makes learning using standard RL methods challenging. Our goal is to see whether the reconstruction-based intrinsic rewards can motivate the model to explore more novel states and achieve good performance. To give a sense of the performance against other methods, we compare our results against RND [2]. We also show an ablation of simply using the reconstruction loss in the pixel space, instead of the embedding space.

We use the standard autoencoder architecture for our RC module. For the encoder, we use four convolutional layers and one fully-connected layer to encode the input frame to a 512-dimensional latent vector. The decoder is a symmetric version of the encoder. In both experiments, we train all models for 1K rollouts of length 128 per environment with 128 parallel environments, which give us a total of around 1.6 million frames of training data. For our models, we use the first 50 rollouts to pre-train the reconstruction network.

The results for Montezuma's Revenge and Venture are shown in Figure 6. Our RC module with prediction error is able to achieve performance that is competitive with RND. After 1K rollouts of training, our model is able to achieve around the same max score as RND. Our RC module with reconstruction error, on the other hand, cannot get any rewards. This is likely due to the noise in pixel-level reconstruction error, which makes learning the policy unstable. This performance is also consistent with training the RL agent without any intrinsic rewards, as shown in [2]. Due to the sparsity of the rewards, standard RL algorithms cannot obtain any rewards on these two games.

4.4. Size of Reconstruction Network

Our approach adds additional costs in training and memory due to the usage of an additional reconstruction network. In this section, we investigate whether the full model used in Section 4.3 is necessary, or if a smaller model could work just as well. To reduce the size of the model, we halve the number of channels used in every convolutional layer. For context, the full model has around 19 million parameters, while the smaller model, denoted as the "half model," has around 8 million parameters. We evaluate the performance of the half model on Montezuma's Revenge and Venture, and plot the performance against our full model. We use the same training setup as in Section 4.3, except we train our models for 1.4K rollouts.

The results are shown in Figure 7. On Montezuma's Revenge, the half model is able to achieve a final performance

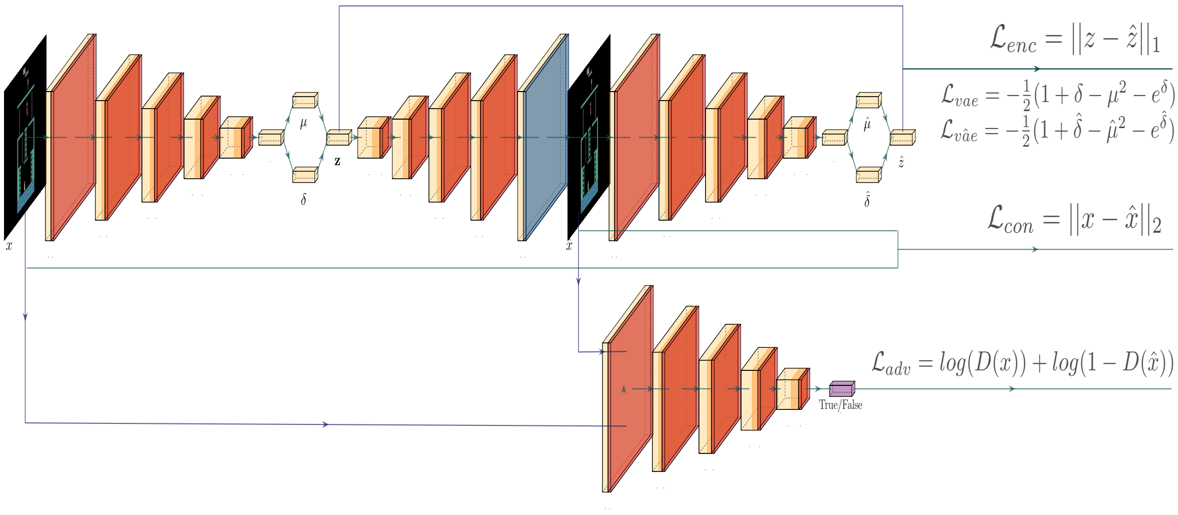
x
μ
δ
z
x
$\hat{\mu}$
$\hat{\delta}$
$\hat{z}$
True/False
$\mathcal{L}_{enc} = ||z - \hat{z}||_1$
$\mathcal{L}_{vae} = -\frac{1}{2}(1 + \delta - \mu^2 - e^{\delta})$
$\mathcal{L}_{v\hat{a}e} = -\frac{1}{2}(1 + \hat{\delta} - \hat{\mu}^2 - e^{\hat{\delta}})$
$\mathcal{L}_{con} = ||x - \hat{x}||_2$
$\mathcal{L}_{adv} = log(D(x)) + log(1 - D(\hat{x}))$

Figure 4: Network architecture of the GAN-augmented VAE and corresponding loss functions.

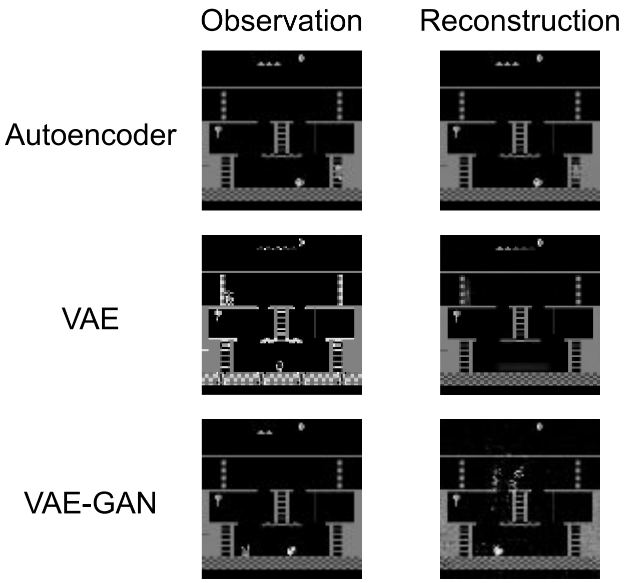

Figure 5: Reconstruction results of different variants of the reconstruction network.

of around 2500, which is the same as the full model. However, the full model is able to achieve the final performance with fewer parameter updates compared to the half model. This shows that the full model is more efficient in utilizing the training data. On Venture, the half model is not able to achieve any rewards at all. Looking at the reconstruction results of the half model, we observed that the half model's reconstructions were missing the details necessary to distinguish between different states. This issue likely caused the intrinsic reward to not have much meaning, thus not helping in exploration of the states. These results show that the half model is not sufficient to achieve the same performance demonstrated by the full model on Montezuma's Revenge and Venture.

5. Discussion

First, we note that, similar to RND, our method only receives state at a single time step as input and therefore cannot abuse stochasticity that stems directly from state *transitions* to receive more intrinsic rewards. From our qualitative evaluations, we observed that our agent trained with RC does not get stuck switching between rooms in Montezuma's revenge while previous prediction-based curiosity rewards exhibit such behavior [1]. Furthermore, we formulated RC in the latent feature space in order to avoid noise that may arise from inconsequential changes in pixel-level details across states that are conceptually similar.

Some environments may require not only local exploration of states, but also exploration of diverse behavior over longer-term trajectories for success. For example, to finish the first level of Montezuma's Revenge, the agent must save keys it obtains to open doors that it later finds rather than immediately using them to gain points. Because RC operates on the per-state level, we expect that its ability to explore on a longer time scale to be limited. Nevertheless, we speculate that RC's use of learned features to capture state novelty may be able to generalize better across states that are temporally further apart. Additionally, it may be fruitful to extend the basic ideas of RC to high-level temporal abstraction frameworks such as options or successor representations which can handle a greater range of time scales and may be more

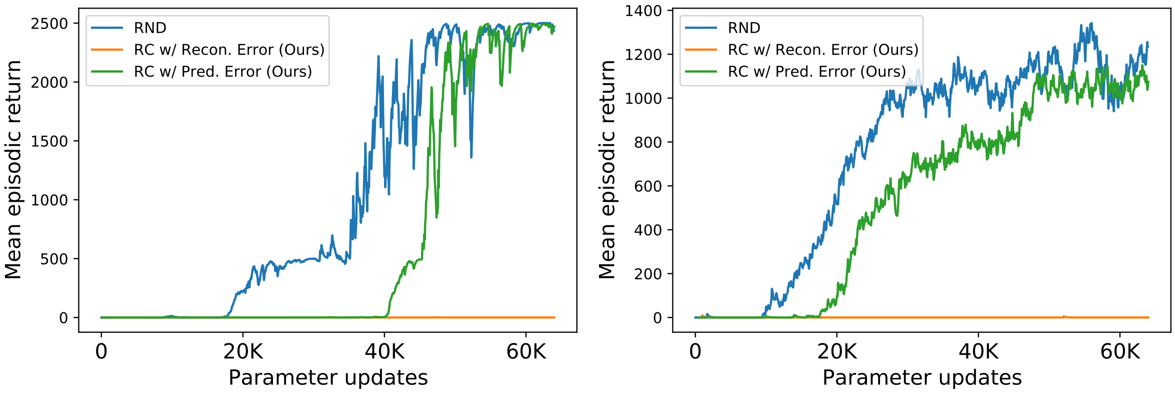

Figure 6: Mean episodic return of different methods on Montezuma's Revenge (left) and Venture (right).

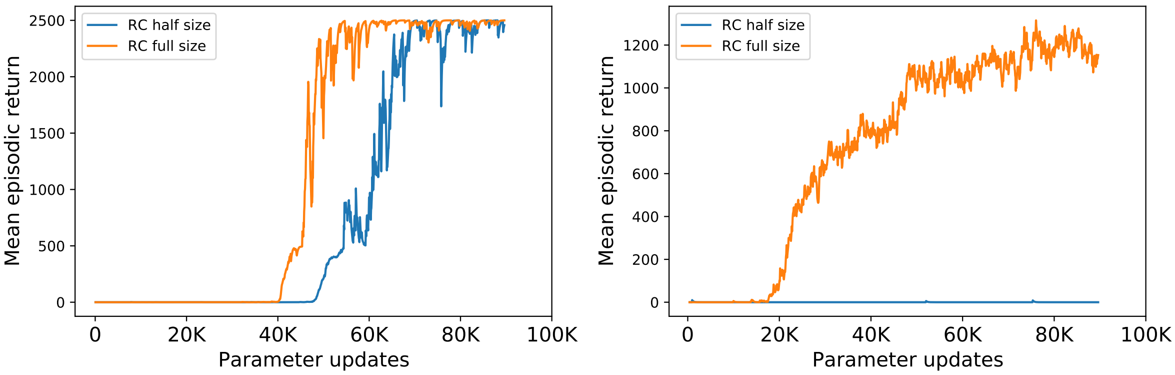

Figure 7: Mean episodic return of networks of different sizes on Montezuma's Revenge (left) and Venture (right).

amendable for long-term exploration [8, 3].

There are also more immediate lines of future work. We would like to do more analysis on what events seem to trigger the intrinsic rewards in the game, and how they differ in different environments such as those in continuous control. New environments may also present opportunities to explore other forms of behavior such as cooperation with multiple agents and how they may interact with curiosity rewards. Additionally, as we continue to develop new forms of intrinsic rewards, there remains work on how to best combine these different reward streams together as investigated in RND [2]. We predict that a better agent may leverage multiple rewards to inform its decision-making.

In general, entirely new forms of curiosity are also worth exploring. Although RND and RC both do not suffer from the noisy TV problem, the way they go about doing so does not seem organic. For example, when changing a TV channel repeatedly, we expect novelty, and soon would grow bored once that form of novelty is consistent. A measurement that captures more meta information, like "surprisal of surprisal", may solve the noisy TV problem in a more natural way. We also have contemplated how methods involving saliency or semantic understanding may also be compelling directions for defining curiosity.

6. Conclusion

In this paper, we presented RC, a new curiosity-based intrinsic reward grounded upon the the ability of a reconstruction network to encode and reconstruct states that an agent encounters.

We successfully verified our hypothesis that error of the reconstruction network in the embedding space can serve as an estimation of state novelty on the MNIST dataset. Our experiments on difficult Atari games with sparse rewards, Montezuma's Revenge and Venture, then demonstrate how RC enables an agent to achieve scores that are competitive with those obtained by RND.

A large motivation for RC is its intuitive, albeit loose formulation as a measure of visual understanding. It remains a question of whether this process and the features learned during its execution can fit more concrete notions of interpretability. We also see potential in integrating local intrinsic reward signals with exploration frameworks that operate on longer time horizons. We hope that future work continues to explore different types of intrinsic rewards and begins to address how to extend these insights to work in higher-order decision making and planning.

References

[1] Yuri Burda, Harrison Edwards, Deepak Pathak, Amos J. Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. *CoRR*, abs/1808.04355, 2018.

[2] Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. *CoRR*, abs/1810.12894, 2018.

[3] Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5:613, 1993.

[4] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 52–59, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, Feb. 2015.

[6] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[7] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR*, abs/1712.01815, 2017.

[8] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181 – 211, 1999.