

PERCEPTION-DRIVEN CURIOSITY WITH BAYESIAN SURPRISE

Anonymous authors

Paper under double-blind review

ABSTRACT

Intrinsic rewards in reinforcement learning provide a powerful algorithmic capability for agents to learn how to interact with their environment in a task-generic way. However, increased incentives for motivation can come at the cost of increased fragility to stochasticity. We introduce a method for computing an intrinsic reward for curiosity using metrics derived from sampling a latent variable model used to estimate dynamics. Ultimately, an estimate of the conditional probability of observed states is used as our intrinsic reward for curiosity. In our experiments, a video game agent uses our model to autonomously learn how to play Atari games using our curiosity reward in combination with extrinsic rewards from the game to achieve improved performance on games with sparse extrinsic rewards. When stochasticity is introduced in the environment, our method still demonstrates improved performance over the baseline.

1 INTRODUCTION

Methods encouraging agents to explore their environment by rewarding actions that yield unexpected results are commonly referred to as *curiosity* (Schmidhuber (1991; 1990a;b)). Using curiosity as an exploration policy in reinforcement learning has many benefits. In scenarios in which extrinsic rewards are sparse, combining extrinsic and intrinsic curiosity rewards gives a framework for agents to discover how to gain extrinsic rewards (Jaegle et al., 2019). In addition, when agents explore, they can build more robust policies for their environment even if extrinsic rewards are readily available (Forestier & Oudeyer, 2015). These policies learned through exploration can give an agent a more general understanding of the results of their actions so that the agent will have a greater ability to adapt using their existing policy if their environment changes.

Despite these benefits, novelty-driven exploration methods can be distracted by randomness. (Schmidhuber, 1990b; Storck et al., 1995) When stochastic elements are introduced in the environment, agents may try to overfit to noise instead of learning a deterministic model of the effect of their own actions on their world. In particular, Burda et al. (2018a) showed that when a TV with white noise is added to an environment in which an agent is using the intrinsic curiosity module (ICM) developed by Pathak et al. (2017), the agent stops exploring the environment and just moves back and forth in front of the TV.

In this paper, we present a new method for agent curiosity which provides robust performance in sparse reward environments and under stochasticity. We use a conditional variational autoencoder (Sohn et al., 2015) to develop a model of our environment. We choose to develop a conditional variational autoencoder (CVAE) due to the success of this architecture in modeling dynamics shown in the video prediction literature (Denton & Fergus, 2018; Xue et al., 2018). We incorporate additional modeling techniques to regularize for stochastic dynamics in our perception model. We compute our intrinsic reward for curiosity by sampling from the latent space of the CVAE and computing an associated conditional probability which is a more robust metric than the commonly used pixel-level reconstruction error.

The primary contributions of our work are the following.

1. **Perception-driven approach to curiosity.** We develop a perception model which integrates model characteristics proven to work well for deep reinforcement learning with recent architectures for estimating dynamics from pixels. This combination retains robust-

ness guarantees from existing deep reinforcement learning models while improving the ability to capture complex visual dynamics.

2. **Bayesian metric for surprise.** We use the entropy of the current state given the last state as a measurement for computing surprise. This Bayesian approach will down-weight stochastic elements of the environment when learning a model of dynamics. As a result, this formulation is robust to noise.

For our experiments, autonomous agents use our model to learn how to play Atari games. We measure the effectiveness of our surprise metric as a meaningful intrinsic reward by tracking the total achieved extrinsic reward by agents using a combination of our intrinsic reward with extrinsic rewards to learn. We show that the policy learned by a reinforcement learning algorithm using our surprise metric outperforms the policies learned by alternate reward schemes. Furthermore, we introduce stochasticity into the realization of actions in the environment, and we show that our method still demonstrates successful performance beyond that of the baseline method.

2 RELATED WORK

Perception-Driven Curiosity. Several existing models incentivize curious agent behavior through estimating and seeking visual novelty. Bellemare et al. (2016) and Ostrovski et al. (2017) generalize count-based exploration traditionally used in tabular settings for continuous states. Burda et al. (2018b) learns a predictive model on features given by a randomly initialized target network and uses reconstruction error of the random features as intrinsic reward. Jaegle et al. (2019) provides a full review of recent perception-driven methods to encourage curiosity. In this work, we combine a CVAE, an architecture recently used to successfully estimate dynamics from image frames, with methodological approaches from deep reinforcement learning to build a robust perception model in which visual novelty is computed via an estimation of the conditional log-likelihood of observed states.

Prediction-Based Exploration Bonuses. Schmidhuber (1991) proposed an approach to exploration by building prediction models and formulating intrinsic reward as the error of the next state prediction. Recently, this line of work has been shown to explore efficiently in a large number of simulated environments by Pathak et al. (2017) and Burda et al. (2018a). Achiam & Sastry (2017) formalizes the prediction error as Bayesian surprise given a heteroscedastic Gaussian predictive model. This approach is closest to our own. However, in contrast to these reward methods which are built upon simple predictive models, our formulation of Bayesian surprise is computed via importance sampling from our latent variable model. This construction of surprise is significant due to the ability of this variational inference approach to express complex multimodal distributions over images.

Information-Theoretic Measures for Exploration. Several methods rely on maximizing information-theoretic measures of agent behavior. The method by Co-Reyes et al. (2018) maximizes the entropy of agent trajectories between successive states directly. Eysenbach et al. (2018) propose to learn skills without supervision by maximizing the mutual information between a latent skill embedding and the behaviour that the associated skill produces. Pong et al. (2019) introduce a method for unsupervised goal-conditional reinforcement learning which maximizes the entropy of the distribution of possible goals. Houthoofd et al. (2016) uses the KL divergence between the previous and current dynamics models as intrinsic reward as a proxy for information gain. We indirectly measure the entropy of agent trajectories by measuring surprise in terms of the entropy of next state given the previous state, providing an alternative to these existing approaches.

Video Prediction with Latent Variable Models. Chung et al. (2015) introduced a stochastic model for sequential data based on variational inference approaches by Kingma & Welling (2013) and Rezende et al. (2014). This model was adopted for high-dimensional data such as video by Denton & Fergus (2018); Babaeizadeh et al. (2018); Lee et al. (2018) and Rybkin et al. (2019). A similar model based on Sohn et al. (2015) was used for next frame prediction in Xue et al. (2018). We leverage the success of variational inference techniques for high-dimensional data to construct a stochastic model of videos from which surprise can be efficiently estimated.

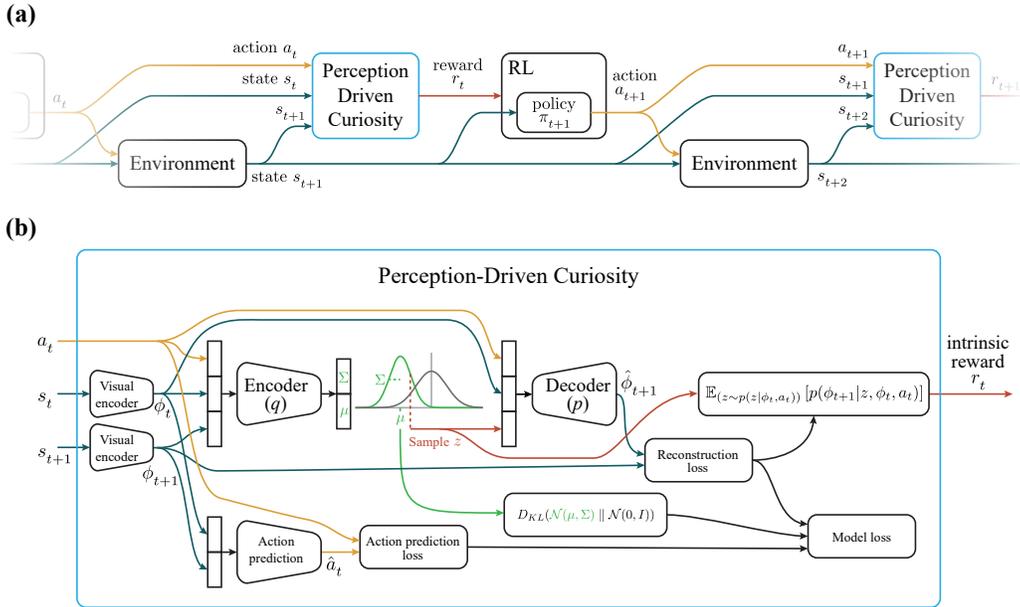


Figure 1: We show how our model integrates with reinforcement learning algorithms to provide intrinsic rewards in (a). The optimization procedure for our perception model as well as our method for computing intrinsic reward are visualized in (b).

3 PERCEPTION MODEL

We construct a perception model for our agents using a conditional variational autoencoder (CVAE) which generates an estimation of an embedded state, $\widehat{\phi}_{t+1}$, given the embedded state itself, ϕ_{t+1} . This generative model is conditioned on the last embedded state, ϕ_t , and action, a_t . Intuitively, this construction gives an agent a visual model of the environment conditioned on the dynamics associated with their interactions with the environment. The state embeddings are encoded by a neural network submodule in our architecture which computes feature vectors ϕ_t and ϕ_{t+1} from states s_t and s_{t+1} respectively. In our experiments, the states we observe from our simulation environment are image frames.

We derive our approach and the following properties of our model from the theoretical properties of conditional variational autoencoders presented by Sohn et al. (2015). We first define $p(\phi_{t+1}|z, \phi_t, a_t)$ as the generative distribution from which we draw the output $\widehat{\phi}_{t+1}$. The prior distribution of the latent space is given by $p(z|\phi_t, a_t)$ which is relaxed in the CVAE formulation to make the latent variable, z , statistically independent of the input variables. Thus, our prior for the latent space distribution is given by $p(z) \sim \mathcal{N}(0, I)$. Through training, our model learns a latent representation. The distribution of this representation, $q(z|\phi_{t+1}, \phi_t, a_t)$, approximates $p(z|\phi_t, a_t)$.

Using the previously defined distributions and the analysis by Sohn et al. (2015), we define the empirical lower bound of the conditional log-likelihood and objective function, f , of our CVAE as

$$f(\phi_t, a_t, \phi_{t+1}) = -D_{KL}(q(z|\phi_t, a_t, \phi_{t+1})||p(z|\phi_t, a_t)) + \frac{1}{N} \sum_{i=1}^N \log p(\phi_{t+1}|z_i, \phi_t, a_t). \quad (1)$$

We recall that the sum of log-probabilities is equal to the reconstruction loss of our model up to a constant and multiplicative offset. Thus, we denote

$$L_{MSE} = \left\| \phi_{t+1} - \widehat{\phi}_{t+1} \right\|_2^2 \approx -\frac{1}{N} \sum_{i=1}^N \log p(\phi_{t+1} | z_i, \phi_t, a_t). \quad (2)$$

We can write the KL-divergence term in Equation 1 as

$$L_{KL} = D_{KL}(q(z|\phi_t, a_t, \phi_{t+1})||p(z|\phi_t, a_t)) = D_{KL}(q(z|\phi_t, a_t, \phi_{t+1})||p(z)). \quad (3)$$

The final component of our perception model is a neural network predicting a_t from ϕ_t and ϕ_{t+1} built off the inverse model presented by Pathak et al. (2017). This component regularizes for dynamics in the environment which do not contribute to agent actions. We note that this component controls for environment stochasticity when learning the weights of our network. The error in action prediction can be formulated in terms of maximum likelihood estimation of the parameters of our network under a multinomial distribution. This error is used as the loss for our action-prediction network and denoted L_A .

We can now use the approximation of our CVAE objective function and the loss from our action-prediction network to formulate the total loss for our perception model as

$$L_{total} = \lambda_1 L_{KL} + \lambda_2 L_{MSE} + \lambda_3 L_A. \quad (4)$$

We tune the hyper-parameters λ_1 , λ_2 , and λ_3 to weight the contribution of each loss in our model. The tuning procedure and hyperparameters used in our experiments are given in Appendix B.

The full architecture of our perception model is shown Figure 1.b.

4 BAYESIAN SURPRISE

We define *Bayesian surprise* as the amount an agent should be curious about an observation derived from a conditional likelihood of the observation occurring given the current world model of the agent. From our definition of curiosity, we want to reward actions more strongly which result in less likely outcomes. Therefore, we use the negative of this conditional probability estimate as a reward for agents. In our approach, this probabilistic reward takes the form $r_t = -\log p(\phi_{t+1}|\phi_t, a_t)$.

Similar objectives were used in prior work which considered simple homoscedastic (Burda et al. (2018a)) or heteroscedastic (Achiam & Sastry (2017)) Gaussian forward models. Due to our use of a base CVAE architecture, our perception model can capture multimodal distributions over images. (Sohn et al., 2015) To retain this improved expressiveness in our derived intrinsic reward, we use importance sampling from the latent space of our CVAE to estimate conditional likelihoods for our formulation of surprise as follows.

$$\log p(\phi_{t+1}|\phi_t, a_t) = \log \mathbb{E}_{(z \sim p(z|\phi_t, a_t))} [p(\phi_{t+1}|z, \phi_t, a_t)] \quad (5)$$

$$= \log \mathbb{E}_{(z \sim q(z|\phi_{t+1}, \phi_t, a_t))} \left[\frac{p(\phi_{t+1}|z, \phi_t, a_t)p(z|\phi_t, a_t)}{q(z|\phi_{t+1}, \phi_t, a_t)} \right] \quad (6)$$

$$= \log \mathbb{E}_{(z \sim q(z|\phi_{t+1}, \phi_t, a_t))} \left[\frac{p(\phi_{t+1}|z, \phi_t, a_t)p(z)}{q(z|\phi_{t+1}, \phi_t, a_t)} \right] \quad (7)$$

$$\geq \mathbb{E}_{(z \sim q(z|\phi_{t+1}, \phi_t, a_t))} \left[\log \frac{p(\phi_{t+1}|z, \phi_t, a_t)p(z)}{q(z|\phi_{t+1}, \phi_t, a_t)} \right] \quad (8)$$

We use the reconstruction loss of our model to compute the conditional probability $\log p(\phi_{t+1}|z, \phi_t, a_t)$.

We recall that the negative logarithm of our conditional probability is equal to Bayesian surprise, so we explicitly define our reward as follows.

$$r_t = -\mathbb{E}_{(z \sim q(z|\phi_{t+1}, \phi_t, a_t))} \left[\log \frac{p(\phi_{t+1}|z, \phi_t, a_t)p(z)}{q(z|\phi_{t+1}, \phi_t, a_t)} \right] \quad (9)$$

We use the Bayesian surprise computed by our perception model as intrinsic reward input to a reinforcement learning algorithm. The interaction of this reward and our perception model with the reinforcement learning procedure is visualized in Figure 1.a.

5 EXPERIMENTS

We evaluate the ability of our model to enable effective and robust exploration. We use Atari video games as simulation environments since they provide reasonably complex visual environments with large variations in both sparsity of extrinsic reward and stochasticity in scenes between different games. As a result, Atari games have been frequently used as a testbed for curiosity approaches. (Pathak et al., 2017; Burda et al., 2018a;b; Mnih et al., 2013) We use our intrinsic reward measurement with the proximal policy optimization (PPO) reinforcement learning algorithm developed by Schulman et al. (2017) due to the ability of PPO to perform well with relatively little hyperparameter tuning. In training, we combine our intrinsic reward with extrinsic rewards provided by the game environments for task-specific success such as knocking blocks out of a wall in Breakout. We compare the ability of agents using this reward combination to learn to play different Atari games against the ability of agents using a leading alternate prediction-based exploration bonus by Pathak et al. (2017) in combination with extrinsic rewards. We also compare our approach to agent behavior derived from policies learned by purely extrinsic rewards.

Note that, though combination rewards are used to train PPO, each method is evaluated by comparing extrinsic reward per episode alone since extrinsic rewards measure the successful accomplishment of tasks in each game. The hyperparameters used in training as well as additional implementation details are given in Appendix B. Furthermore, Appendix A shows details and analysis of the perception model performance throughout training via this active learning procedure.

5.1 CURIOSITY-AIDED GAME PLAY

We first test the impact of our curiosity-reward on learning to play Atari games with varying levels of extrinsic reward sparseness. Gravitar, Beam Rider, Breakout, Space Invaders, and River Raid all have reasonably dense extrinsic rewards. In contrast, Montezuma’s Revenge, Pitfall, and Private Eye all have sparse extrinsic rewards and are thus known as a traditionally challenging games for deep reinforcement learning algorithms to play successfully with only the information provided by game scene observations.

Our results for training 3 seeds for each method over 10 million timesteps in each of these games are plotted in Figure 2. Table 1 summarizes the results of the extrinsic rewards achieved at the end of training. The best performance for each game is bold in the respective row.

For games with dense extrinsic rewards, the best performance is split somewhat equally between each of the 3 reward strategies. Thus, we conclude that we perform comparably to ICM in the case

Table 1: Mean and standard deviation of extrinsic reward over last 1 million time steps in training across 3 independent seeds for each model.

Atari Game	Reward Strategies		
	Extrinsic	ICM+Extrinsic	Ours+Extrinsic
Gravitar	525.82 ± 200.77	601.09 ± 322.02	582.00 ± 269.63
Private Eye	61.92 ± 49.45	77.89 ± 36.28	85.70 ± 22.61
Space Invaders	811.57 ± 189.11	1086.29 ± 260.13	1041.50 ± 175.52
Beam Rider	3250.01 ± 843.39	3131.36 ± 945.44	2755.15 ± 721.71
Breakout	262.49 ± 22.16	249.89 ± 37.11	263.60 ± 45.28
River Raid	7952.27 ± 1167.93	6466.26 ± 2373.94	6428.72 ± 2283.45
Pitfall	-1.20 ± 1.22	-3.04 ± 2.02	-1.63 ± 1.37
Montezuma’s Revenge	0.05 ± 0.36	0.00 ± 0.00	3.19 ± 8.31

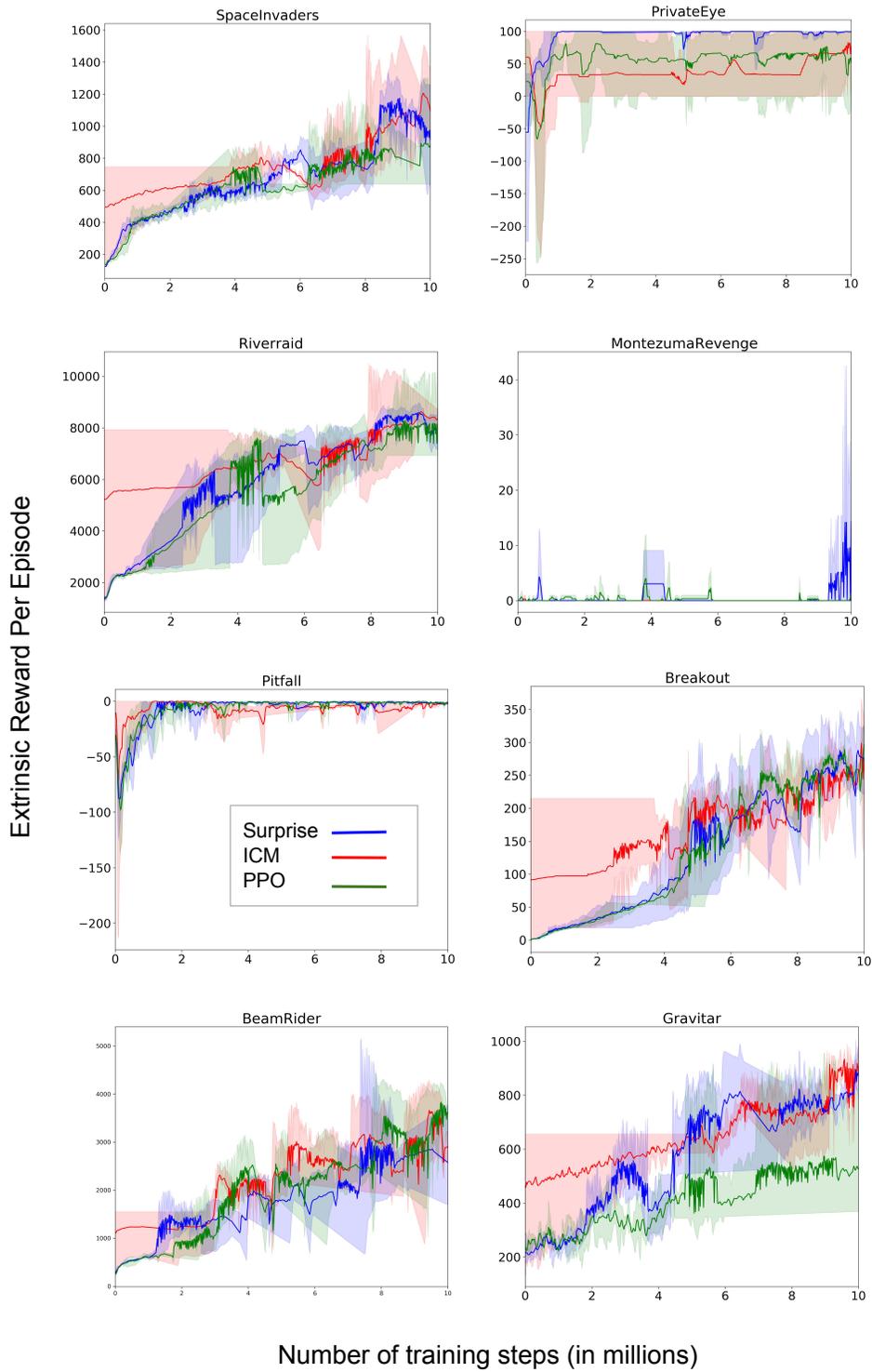


Figure 2: Extrinsic reward per episode achieved in training over 10 million time steps and 3 seeds for the following Atari games: Beam Rider, Breakout, Gravitar, River Raid, Private Eye, Space Invaders, Montezuma’s Revenge, and Pitfall.

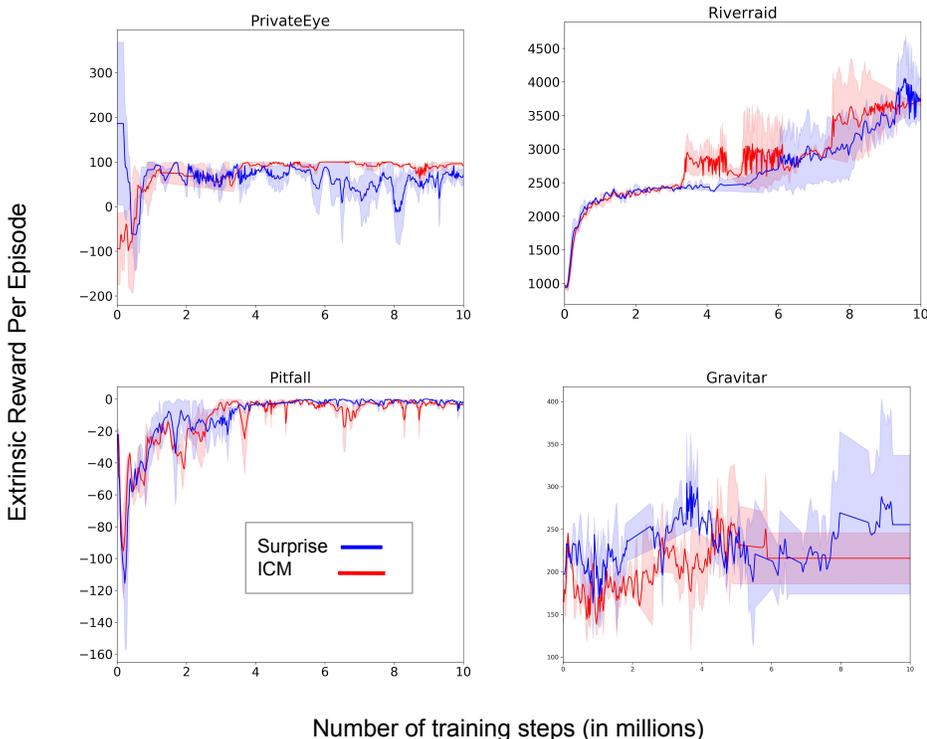


Figure 3: Extrinsic reward per episode achieved with sticky actions in training over 10 million time steps and 2 seeds for the following Atari games: Gravitar, River Raid, Private Eye, and Pitfall.

where extrinsic rewards are readily available. However, we also conclude that the value of using curiosity for the purpose of achieving improved performance in these cases is substantially less significant than when rewards are sparse considering that we also show comparable performance to using extrinsic reward only in these games.

We recall the games with sparse extrinsic rewards are Pitfall, Montezuma’s Revenge, and Private Eye. Though PPO achieved the highest extrinsic reward on Pitfall, no successful game strategy was found. The mean extrinsic reward for each method is always negative in Pitfall. Exploration necessary to ultimately discover a successful strategy may require incurring temporary negative extrinsic rewards, so comparing the average results before any working strategy is learned by any method is premature.

On both Montezuma’s Revenge and Private Eye, our method outperforms the combination of ICM and extrinsic rewards as well as extrinsic rewards only. We further observe that the variance of our method at convergence in Private Eye is very low. These results demonstrate the improved ability of our approach to use information from better models for perceiving complex scene dynamics in order to learn in the absence of extrinsic rewards.

5.2 ROBUSTNESS TO STOCHASTICITY

There is a challenging balance with methods rewarding novelty to encourage exploration of the unknown while avoiding confounding the model by focusing on randomness. We showed that our method explores more effectively than our baseline in cases where extrinsic rewards are sparse. However, we now need to demonstrate that that improved performance did not introduce brittleness to environmental noise.

Following Burda et al. (2018b), we perform a sticky action experiment to demonstrate the robustness of our model to stochasticity. With a 25% probability, the action an agent takes is repeated over 8 consecutive frames in the environment though the agent believes that their new action decisions are

being executing across those frames. We observe that the performance of ICM and our model in the presence of sticky actions is comparable with slightly better performance from our model.

Our results for training 3 seeds for each method over 10 million timesteps in each of these games are plotted in Figure 2. Table 2 summarizes the results of the extrinsic rewards achieved at the end of training. The best performance for each game is bold in the respective row.

Table 2: Mean and standard deviation of extrinsic reward while using sticky actions over last 1 million time steps in training across 2 independent seeds for each model.

Atari Game	Reward Strategies	
	ICM+Extrinsic	Ours+Extrinsic
Gravitar	216.00 \pm 30.00	263.42 \pm 90.79
Private Eye	92.18 \pm 7.85	65.65 \pm 24.56
River Raid	3656.87 \pm 173.70	3692.37 \pm 437.56
Pitfall	-3.09 \pm 1.38	-1.73 \pm 2.05

6 DISCUSSION

In summary, we presented a novel method to compute curiosity through the use of a meaningfully constructed model for perception. We used a conditional variational autoencoder (CVAE) to learn scene dynamics from image and action sequences and computed an intrinsic reward for curiosity via a conditional probability derived from importance sampling from the latent space of our CVAE. In our experiments, we demonstrated that our approach allows agents to learn to accomplish tasks more effectively in environments with sparse extrinsic rewards without compromising robustness to stochasticity.

We show robustness to stochasticity in our action space which we support through the action-prediction network used in our perception model. However, robustness to stochasticity in scenes is a separate challenge which the method we use as our baseline, ICM, cannot handle well. (Burda et al., 2018a) Stochasticity in scenes occurs when there are significant changes between sequential image frames which are random with respect to agent actions. We hypothesize that this stochasticity requires a different approach to handle.

A consideration in comparing models for curiosity and exploration in deep reinforcement learning is that typically both the dynamics model and intrinsic reward metric are constructed and compared as unit as we did in this paper. However, a conditional probability estimation could be derived the dynamics model given by ICM just as reconstruction error could be used as intrinsic reward from our CVAE. Alternately, other metrics measuring novelty and learning such as the KL divergence between sequential latent distributions in our model have been proposed in a general manner by Schmidhuber (2010).

An interesting direction for future work would be to explore the impact of intrinsic reward metrics for curiosity on robustness to stochasticity in scenes independent across different choices of dynamics model.

REFERENCES

- Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *CoRR*, abs/1703.01732, 2017. URL <http://arxiv.org/abs/1703.01732>.
- Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. 2018.
- 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.
- Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

- Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018a.
- Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018b.
- Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *NIPS*, 2015.
- John D Co-Reyes, YuXuan Liu, Abhishek Gupta, Benjamin Eysenbach, Pieter Abbeel, and Sergey Levine. Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. *arXiv preprint arXiv:1806.02813*, 2018.
- Emily Denton and Rob Fergus. Stochastic Video Generation with a Learned Prior. In *ICML*, 2018.
- Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. <https://github.com/openai/baselines>, 2017.
- Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning diverse skills without a reward function. 2018.
- Sébastien Forestier and Pierre-Yves Oudeyer. Towards hierarchical curiosity-driven exploration of sensorimotor models. In *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pp. 234–235. IEEE, 2015.
- Rein Houthoofd, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *NIPS*, 2016.
- Andrew Jaegle, Vahid Mehrpour, and Nicole Rust. Visual novelty, curiosity, and intrinsic reward in machine learning and the brain. *arXiv preprint arXiv:1901.02478*, 2019.
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *arXiv:1804.01523*, abs/1804.01523, 2018.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*. 2013.
- Georg Ostrovski, Marc G Bellemare, Aäron van den Oord, and Rémi Munos. Count-based exploration with neural density models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2721–2730. JMLR. org, 2017.
- Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. *CoRR*, abs/1705.05363, 2017. URL <http://arxiv.org/abs/1705.05363>.
- Vitchyr H Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*, 2019.
- Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.
- Oleh Rybkin, Karl Pertsch, Andrew Jaegle, Konstantinos G. Derpanis, and Kostas Daniilidis. Learning what you can do before doing anything. *International Conference on Learning Representations (ICLR)*, 2019.
- J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Trans. on Auton. Ment. Dev.*, 2(3):230–247, September 2010. ISSN 1943-0604. doi: 10.1109/TAMD.2010.2056368. URL <https://doi.org/10.1109/TAMD.2010.2056368>.

- Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animals*, pp. 222–227, Cambridge, MA, USA, 1990a. MIT Press. ISBN 0-262-63138-5. URL <http://dl.acm.org/citation.cfm?id=116517.116542>.
- Jürgen Schmidhuber. Curious model-building control systems. In *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, pp. 1458–1463. IEEE, 1991.
- Jürgen Schmidhuber. Making the world differentiable: On using self-supervised fully recurrent neural networks for dynamic reinforcement learning and planning in non-stationary environments. Technical report, 1990b.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL <http://arxiv.org/abs/1707.06347>.
- Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pp. 3483–3491, 2015.
- Jan Storck, Sepp Hochreiter, and Jürgen Schmidhuber. Reinforcement driven information acquisition in non-deterministic environments, 1995.
- Tianfan Xue, Jiajun Wu, Katherine L. Bouman, and William T. Freeman. Visual dynamics: Stochastic future generation via layered cross convolutional networks. *CoRR*, abs/1807.09245, 2018. URL <http://arxiv.org/abs/1807.09245>.

A PERCEPTION MODEL ANALYSIS

We analyze the ability of an agent using our model to perceive the environment. We confirm that our perception model generates image embeddings which can successfully construct realistic image frames from our video games. We also show that increasing ability to perceive the environment is associated with decreasing intrinsic rewards.

To set up this experiment, we built a visual decoder to reconstruct images from the embeddings learned by the visual encoder shown in Figure 1.b. We trained our decoder on the game images (s_t) and the image embeddings (ϕ_t). Then, we used our decoder to reconstruct images from our predicted image embeddings ($\hat{\phi}_t$). We chose to execute this experiment on Kung Fu Master since it is one of the more visually complex Atari games.

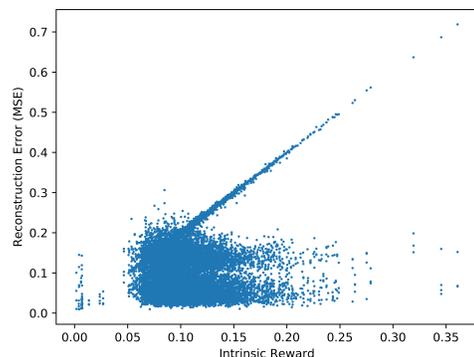


Figure 4: Relationship between reconstruction error and intrinsic reward.

tively early in training. At this time, the agent has learned to only partially perceive the environment. Thus, we can visualize reasonable reconstructed scenes, but we have enough reconstruction

Figure 5 shows the reconstructed images from our perception model next to the image frames which are our states. We note that the observations provided by the OpenAI Gym Atari game simulations are grayscale and rescaled to size 84x84. These images are what we input into our visual encoder. The reconstructions of our predicted embeddings are shown next to these images along with measurements of reconstruction error between the embedded images and intrinsic reward associated with that time step. We observe that the high intrinsic rewards are associated with poor reconstructions of the image frames. Analogously, low intrinsic rewards are associated with good reconstructions of the image frames.

We note that all of the scenes presented in this visualization are intentionally taken from relatively early in training. At this time, the agent has learned to only partially perceive the environment. Thus, we can visualize reasonable reconstructed scenes, but we have enough reconstruction

Table 3: Hyperparameters used in experiments.

CVAE Latent Dimension	64
KL Divergence Loss Weight	0.01
Reconstruction Error Loss Weight	0.39
Action-Prediction Network Loss Weight	0.5
Intrinsic Reward Weight	0.01
Extrinsic Reward Weight	0.99

error such that the perception component of our network will dominate the likelihood measurement we use for surprise. When the perception model improves after additional training, the relationship between intrinsic reward and image reconstruction error becomes less strong since intrinsic reward is also conditioned on the likelihood of the next state prediction which is determined by transition dynamics as well.

We more clearly observe the correlation between reconstruction error and intrinsic reward in Figure 4 for a subset of the training samples in our model. The linear trend becomes weaker and rewards are not as tightly clustered for samples later in training which demonstrates that our model recognizes distinct transition dynamics based on likelihood.

Through our analysis in Figure 5, we observe the success of our CVAE model in perceiving the game environment in Kung Fu Master. In addition, we validate the success of our designed relationship between intrinsic reward and ability to perceive the environment with this analysis in Figure 4.

B IMPLEMENTATION

To execute our experiments, we leveraged the implementation of the PPO algorithm provided by Dhariwal et al. (2017). We used the Atari simulation environment for our video game simulations available in OpenAI Gym and developed by Brockman et al. (2016). We also incorporated infrastructure code from Pathak et al. (2017) along with their inverse model implementation which we use as an action-prediction network in our approach.

We trained each method for 10 million time steps on each environment. Each time step is associated with processing one new frame from the environment. On a 2080 Ti NVIDIA GPU, training for 10 million time steps took approximately 7 hours. This training time is associated with processing about 425 frames per second.

To tune hyperparameters for loss weights in our model, we used intrinsic reward only and swept a range of values between zero and one in a grid search for each weight. Thus, we analyzed how different weighting of the loss values induced agents to explore differently, and we choose the combination which yielded the maximum resulting extrinsic reward realization through intrinsic reward training only.

Once we had optimal loss weights, we then tuned for the intrinsic and extrinsic reward weight combination. We swept a range of values between zero and one this weight combination as well, and we again chose the values which yielded the highest extrinsic reward. We tried 3 different scales of latent space dimension before choosing the one which caused our perception model to perform the best.

The hyperparameters we used in our experiments are listed in Table 3. Note that the intrinsic reward weight and extrinsic reward weight were the same for ICM and our method though we tuned for the weights separately for each model. Also, note we took all the hyperparameters required in ICM other than reward weighting from Pathak et al. (2017). Intrinsic and extrinsic reward weights were not provided for using ICM with PPO.

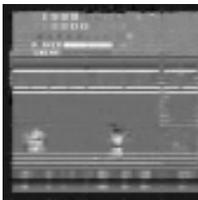
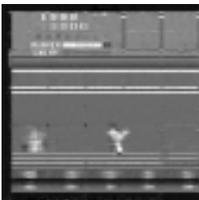
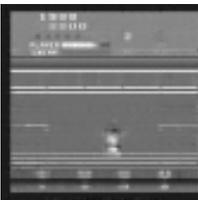
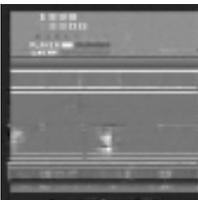
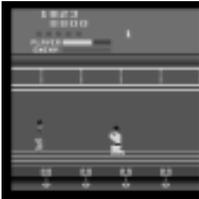
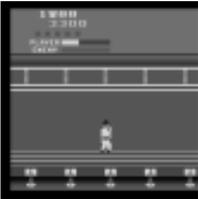
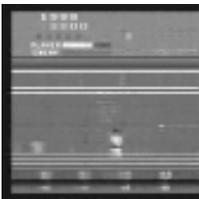
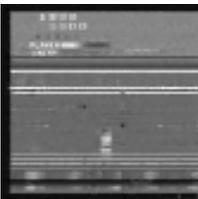
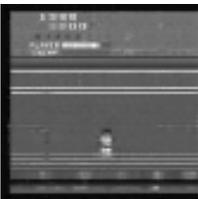
Input image				
Reconstructed prediction				
Intrinsic reward	0.053	0.056	0.076	0.094
Feature reconst. err.	0.121	0.120	0.117	0.163
Input image				
Reconstructed prediction				
Intrinsic reward	0.117	0.132	0.141	0.196
Feature reconst. err.	0.184	0.178	0.219	0.230

Figure 5: The first row shows the game scene images fed into our perception model as states. The second row shows the reconstructed images produced by our perception model. The remaining rows list intrinsic reward and embedded image reconstruction error respectively for each image pair.