Curiosity-driven Exploration by Self-supervised Prediction

Deepak Pathak [1] **Pulkit Agrawal** [1] **Alexei A. Efros** [1] **Trevor Darrell** [1]

Abstract

In many real-world scenarios, rewards extrinsic to the agent are extremely sparse, or absent altogether. In such cases, curiosity can serve as an intrinsic reward signal to enable the agent to explore its environment and learn skills that might be useful later in its life. We formulate curiosity as the error in an agent's ability to predict the consequence of its own actions in a visual feature space learned by a self-supervised inverse dynamics model. Our formulation scales to high-dimensional continuous state spaces like images, bypasses the difficulties of directly predicting pixels, and, critically, ignores the aspects of the environment that cannot affect the agent. The proposed approach is evaluated in two environments: *VizDoom* and *Super Mario Bros*. Three broad settings are investigated: 1) sparse extrinsic reward, where curiosity allows for far fewer interactions with the environment to reach the goal; 2) exploration with no extrinsic reward, where curiosity pushes the agent to explore more efficiently; and 3) generalization to unseen scenarios (e.g. new levels of the same game) where the knowledge gained from earlier experience helps the agent explore new places much faster than starting from scratch.

1. Introduction

Reinforcement learning algorithms aim at learning policies for achieving target tasks by maximizing rewards provided by the environment. In some scenarios, these rewards are supplied to the agent continuously, e.g. the running score in an Atari game (Mnih et al., 2015), or the distance between a robot arm and an object in a reaching task (Lillicrap et al., 2016). However, in many real-world scenarios, rewards extrinsic to the agent are extremely sparse or missing altogether, and it is not possible to construct a shaped reward function. This is a problem as the agent receives reinforcement for updating its policy only if it succeeds in reaching a pre-specified goal state. Hoping to stumble into a goal state by chance (i.e. random exploration) is likely to be futile for all but the simplest of environments.

(a) learn to explore in Level-1 (b) explore faster in Level-2

Figure 1. Discovering how to play *Super Mario Bros* **without rewards**. (a) Using only curiosity-driven exploration, the agent makes significant progress in Level-1. (b) The gained knowledge helps the agent explore subsequent levels much faster than when starting from scratch. Watch the video at `http://pathak22.github.io/noreward-rl/`

As human agents, we are accustomed to operating with rewards that are so sparse that we only experience them once or twice in a lifetime, if at all. To a three-year-old enjoying a sunny Sunday afternoon on a playground, most trappings of modern life – college, good job, a house, a family – are so far into the future, they provide no useful reinforcement signal. Yet, the three-year-old has no trouble entertaining herself in that playground using what psychologists call intrinsic motivation (Ryan, 2000) or curiosity (Silvia, 2012). Motivation/curiosity have been used to explain the need to explore the environment and discover novel states. More generally, curiosity is a way of learning new skills which might come handy for pursuing rewards in the future.

Similarly, in reinforcement learning, intrinsic motivation/rewards become critical whenever extrinsic rewards are sparse. Most formulations of intrinsic reward can be grouped into two broad classes: 1) encourage the agent to explore "novel" states (Bellemare et al., 2016; Lopes et al., 2012; Poupart et al., 2006) or, 2) encourage the agent to perform actions that reduce the error/uncertainty in the agent's ability to predict the consequence of its own actions (i.e. its knowledge about the environment) (Houthooft

[1]University of California, Berkeley. Correspondence to: Deepak Pathak <pathak@berkeley.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

et al., 2016; Mohamed & Rezende, 2015; Schmidhuber, 1991; 2010; Singh et al., 2005; Stadie et al., 2015).

Measuring "novelty" requires a statistical model of the distribution of the environmental states, whereas measuring prediction error/uncertainty requires building a model of environmental dynamics that predicts the next state (s_{t+1}) given the current state (s_t) and the action (a_t) executed at time t. Both these models are hard to build in high-dimensional continuous state spaces such as images. An additional challenge lies in dealing with the stochasticity of the agent-environment system, both due to the noise in the agent's actuation, and, more fundamentally, due to the inherent stochasticity in the environment. To give the example from (Schmidhuber, 2010), if the agent receiving images as state inputs is observing a television screen displaying white noise, every state will be novel as it would be impossible to predict the value of any pixel in the future. This means that the agent will remain curious about the television screen because it is unaware that some parts of the state space simply cannot be modeled and thus the agent can fall into an artificial curiosity trap and stall its exploration. Other examples of such stochasticity include appearance changes due to shadows from other moving entities or presence of distractor objects. Somewhat different, but related, is the challenge of generalization across physically (and perhaps also visually) distinct but functionally similar parts of an environment, which is crucial for large-scale problems. One proposed solution to all these problems is to only reward the agent when it encounters states that are hard to predict but are "learnable" (Schmidhuber, 1991). However, estimating learnability is a non-trivial problem (Lopes et al., 2012).

This work belongs to the broad category of methods that generate an intrinsic reward signal based on how hard it is for the agent to predict the consequences of its own actions. However, we manage to escape most pitfalls of previous prediction approaches with the following key insight: we only predict those changes in the environment that could possibly be due to the actions of our agent or affect the agent, and ignore the rest. That is, instead of making predictions in the raw sensory space (e.g. pixels), we transform the sensory input into a feature space where only the information relevant to the action performed by the agent is represented. We learn this feature space using self-supervision – training a neural network on a proxy inverse dynamics task of predicting the agent's action given its current and next states. Since the neural network is only required to predict the action, it has no incentive to represent within its feature embedding space the factors of variation in the environment that do not affect the agent itself. We then use this feature space to train a forward dynamics model that predicts the feature representation of the next state, given the feature representation of the current state and the action. We provide the prediction error of the forward dynamics model to the agent as an intrinsic reward to encourage its curiosity.

The role of curiosity has been widely studied in the context of solving tasks with sparse rewards. In our opinion, curiosity has two other fundamental uses. Curiosity helps an agent explore its environment in the quest for new knowledge (a desirable characteristic of exploratory behavior is that it should improve as the agent gains more knowledge). Further, curiosity is a mechanism for an agent to learn skills that might be helpful in future scenarios. In this paper, we evaluate the effectiveness of our curiosity formulation in all three of these roles.

We first compare the performance of an A3C agent (Mnih et al., 2016) with and without the curiosity signal on 3D navigation tasks with sparse extrinsic reward in the *VizDoom* environment. We show that a curiosity-driven intrinsic reward is crucial in accomplishing these tasks (see Section 4.1). Next, we show that even in the absence of any extrinsic rewards, a curious agent learns good exploration policies. For instance, an agent trained only with curiosity as its reward is able to cross a significant portion of Level-1 in *Super Mario Bros*. Similarly in *VizDoom*, the agent learns to walk intelligently along the corridors instead of bumping into walls or getting stuck in corners (see Section 4.2). A question that naturally follows is whether the learned exploratory behavior is specific to the physical space that the agent trained itself on, or if it enables the agent to perform better in unseen scenarios too? We show that the exploration policy learned in the first level of *Mario* helps the agent explore subsequent levels faster (shown in Figure 1), while the intelligent walking behavior learned by the curious *VizDoom* agent transfers to a completely new map with new textures (see Section 4.3). These results suggest that the proposed method enables an agent to learn generalizable skills even in the absence of an explicit goal.

2. Curiosity-Driven Exploration

Our agent is composed of two subsystems: a reward generator that outputs a curiosity-driven intrinsic reward signal and a policy that outputs a sequence of actions to maximize that reward signal. In addition to intrinsic rewards, the agent optionally may also receive some extrinsic reward from the environment. Let the intrinsic curiosity reward generated by the agent at time t be r_t^i and the extrinsic reward be r_t^e. The policy sub-system is trained to maximize the sum of these two rewards $r_t = r_t^i + r_t^e$, with r_t^e mostly (if not always) zero.

We represent the policy $\pi(s_t; \theta_P)$ by a deep neural network with parameters θ_P. Given the agent in state s_t, it executes the action $a_t \sim \pi(s_t; \theta_P)$ sampled from the policy. θ_P is

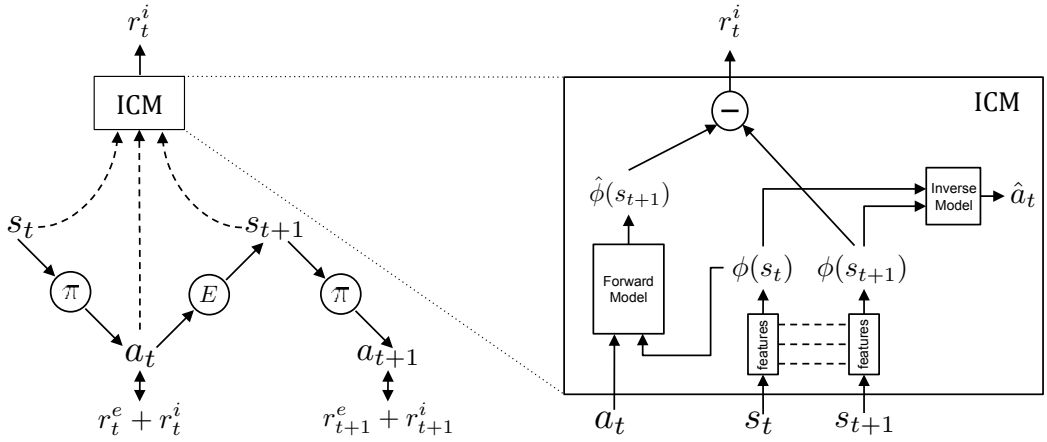


Figure 2. The agent in state s_t interacts with the environment by executing an action a_t sampled from its current policy π and ends up in the state s_{t+1}. The policy π is trained to optimize the sum of the extrinsic reward (r_t^e) provided by the environment *E* and the curiosity based intrinsic reward signal (r_t^i) generated by our proposed Intrinsic Curiosity Module (ICM). ICM encodes the states s_t, s_{t+1} into the features $\phi(s_t), \phi(s_{t+1})$ that are trained to predict a_t (i.e. inverse dynamics model). The forward model takes as inputs $\phi(s_t)$ and a_t and predicts the feature representation $\hat{\phi}(s_{t+1})$ of s_{t+1}. The prediction error in the feature space is used as the curiosity based intrinsic reward signal.

optimized to maximize the expected sum of rewards,

$$\max_{\theta_P} \mathbb{E}_{\pi(s_t;\theta_P)}[\Sigma_t r_t] \tag{1}$$

Unless specified otherwise, we use the notation $\pi(s)$ to denote the parameterized policy $\pi(s;\theta_P)$. Our curiosity reward model can potentially be used with a range of policy learning methods; in the experiments discussed here, we use the asynchronous advantage actor critic policy gradient (A3C) (Mnih et al., 2016) for policy learning. Our main contribution is in designing an intrinsic reward signal based on prediction error of the agent's knowledge about its environment that scales to high-dimensional continuous state spaces like images, bypasses the hard problem of predicting pixels and is unaffected by the unpredictable aspects of the environment that do not affect the agent.

2.1. Prediction error as curiosity reward

Making predictions in the raw sensory space (e.g. when s_t corresponds to images) is undesirable not only because it is hard to predict pixels directly, but also because some part of the input sensory space could be unpredictable and inconsequential to the agent, for e.g., the movement and location of tree leaves in a breeze in the environment.

For determining a good feature space for making future predictions, let's divide all sources that can influence the agent's observations into three cases: (1) things that can be controlled by the agent; (2) things that the agent cannot control but can affect the agent (e.g. a vehicle driven by another agent), and (3) things out of the agent's control and not affecting the agent (e.g. moving leaves). A good feature space for curiosity should model (1) and (2) and be unaffected by (3). The latter is because, if there is a source of variation that is inconsequential for the agent, then the agent has no incentive to know about it.

2.2. Self-supervised prediction for exploration

Instead of hand-designing features for every environment, we propose a general mechanism for learning features for prediction error based curiosity. Given the raw state s_t, we encode it using a deep neural network into a feature vector $\phi(s_t;\theta_E)$, denoted as $\phi(s_t)$ for succinctness. We propose to learn the parameters of this feature encoder using two sub-modules described as follows. The first sub-module is the neural network g which takes the feature encoding $\phi(s_t), \phi(s_{t+1})$ of two consequent states as input and predicts the action a_t taken by the agent to move from state s_t to s_{t+1}, defined as:

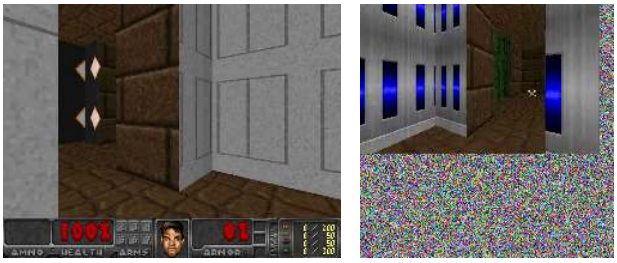
$$\hat{a}_t = g\Big(\phi(s_t), \phi(s_{t+1}); \theta_I\Big) \tag{2}$$

where, $\hat{a}_t$ is the predicted estimate of the action a_t. The neural network parameters θ_I, θ_E are trained to optimize,

$$\min_{\theta_I,\theta_E} L_I(\hat{a}_t, a_t) \tag{3}$$

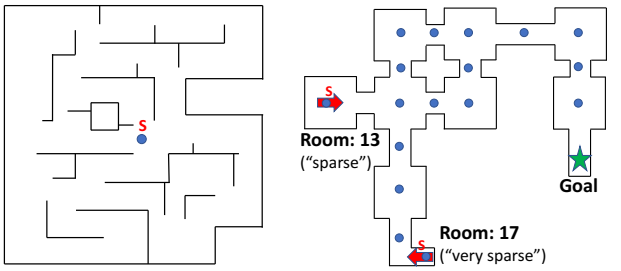
where, L_I measures the discrepancy between the predicted and actual actions. L_I is modeled as soft-max loss across all possible actions when a_t is discrete. The learned function g is also known as the inverse dynamics model and the tuple (s_t, a_t, s_{t+1}) required to learn g is obtained while the agent interacts with the environment using its current policy $\pi(s)$.

Simultaneously with the inverse model g, we train another sub-module that takes as inputs a_t and $\phi(s_t)$ to predict the



(a) Input snapshot in VizDoom (b) Input w/ noise

Figure 3. Frames from VizDoom 3D environment which agent takes as input: (a) Usual 3D navigation setup; (b) Setup when uncontrollable noise is added to the input.



(a) Train Map Scenario (b) Test Map Scenario

Figure 4. Maps for VizDoom 3D environment: (a) The map where the agent is pre-trained only using curiosity signal without any reward from environment. 'S' denotes the starting position. (b) Testing map for performance evaluation. Green star denotes goal location. Blue dots refer to 17 agent spawning locations in the map in the "dense" case. Rooms 13, 17 are the fixed start locations of agent in "sparse" and "very sparse" reward cases respectively. Note train and test maps have different textures as well.

feature encoding of the state at time step $t+1$,

$$\hat{\phi}(s_{t+1}) = f\Big(\phi(s_t), a_t; \theta_F\Big) \tag{4}$$

where $\hat{\phi}(s_{t+1})$ is the predicted estimate of $\phi(s_{t+1})$. The function f is also known as the forward dynamics model and is trained to optimize the regression loss,

$$\min_{\theta_F,\theta_E} L_F\Big(\hat{\phi}(s_{t+1}), \phi(s_{t+1})\Big) \tag{5}$$

Finally, the intrinsic reward signal r_t^i is computed as,

$$r_t^i = \frac{\eta}{2}\|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\|_2^2 \tag{6}$$

where $\eta > 0$ is a scaling factor. The inverse and forward dynamics losses, described in equations (3) and (5), are jointly optimized with the policy. The inverse model helps learn a feature space that encodes information relevant for predicting the agent's actions only and the forward model makes this learned feature representation more predictable. We refer to this proposed curiosity formulation as Intrinsic Curiosity Module (ICM). As there is no incentive for this feature space to encode any environmental features that are not influenced by the agent's actions, our agent will receive no rewards for reaching environmental states that are inherently unpredictable and its exploration strategy will be robust to nuisance sources of variation in the environment. See Figure 2 for illustration of the formulation.

The overall optimization problem can be written as,

$$\min_{\theta_P,\theta_I,\theta_F,\theta_E} \Bigg[-\lambda\mathbb{E}_{\pi(s_t;\theta_P)}[\Sigma_t r_t] + (1-\beta)L_I + \beta L_F\Bigg] \tag{7}$$

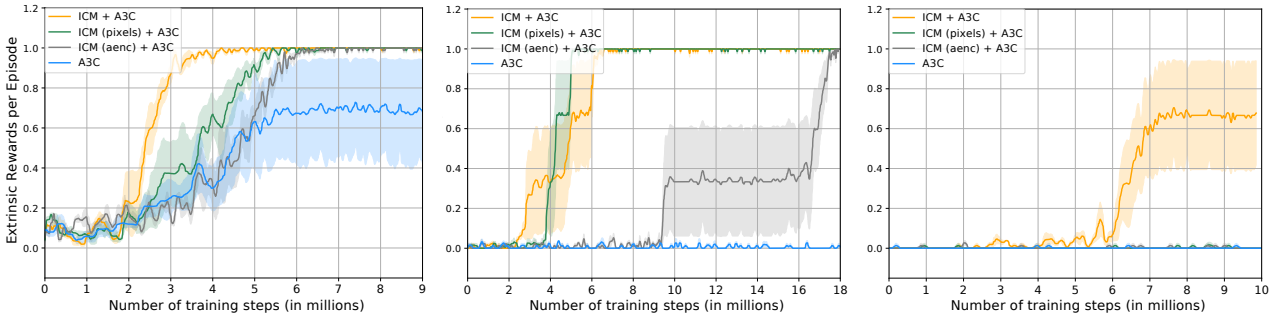
where $0 \leq \beta \leq 1$ is a scalar that weighs the inverse model loss against the forward model loss and $\lambda > 0$ weighs the importance of the policy gradient loss against the intrinsic reward signal. We do not backpropagate the policy gradient loss to the forward model to prevent degenerate solution of agent rewarding itself.

Previous work has investigated inverse models to learn features (Agrawal et al., 2015; 2016; Jayaraman & Grauman, 2015) and forward models to regularize those features (Agrawal et al., 2016) for recognition tasks. However, they do not learn any policy for the agent.

3. Experimental Setup

Environments Our first environment is the VizDoom (Kempka et al., 2016) game where we consider the 3D navigation task with four discrete actions – forward, left, right, and no-action. Our testing setup in all the experiments is the 'DoomMyWayHome-v0' environment which is available as part of OpenAI Gym (Brockman et al., 2016). The map consists of 9 rooms connected by corridors and the agent is tasked to reach some fixed goal location from its spawning location. Episodes are terminated either when the agent reaches the fixed goal or if the agent exceeds a maximum of 2100 time steps. The agent is only provided a sparse terminal reward of +1 if it finds the vest and zero otherwise. For generalization experiments, we pre-train on a different map with different random textures from (Dosovitskiy & Koltun, 2016) with 2100 step long episodes as there is no goal in pre-training. Sample frames from VizDoom are shown in Figure 3a, and maps are explained in Figure 4. It takes approximately 350 steps for an optimal policy to reach the vest location from the farthest room in this map (sparse reward).

Our second environment is the classic Nintendo game Super Mario Bros with a reparamterized 14 dimensional action space following (Paquette, 2016). The actual game is played using a joystick allowing for multiple simultaneous button presses, where the duration of the press affects what action is being taken. This property makes the game particularly hard, e.g. to make a long jump over tall pipes or wide gaps, the agent needs to predict the same action up to 12 times in a row, introducing long-range dependencies.



(a) "dense reward" setting (b) "sparse reward" setting (c) "very sparse reward" setting

Figure 5. Comparing the performance of the A3C agent with no curiosity (blue), ICM-pixels + A3C (green) and the proposed ICM + A3C agent (orange) in the "dense", "sparse" and "very sparse" reward scenarios of VizDoom. The curious A3C agents significantly outperforms baseline A3C agent as the sparsity of reward increases. Pixel based curiosity works in dense and sparse but fails in very sparse reward setting. The dark line and shaded area show mean and mean $\pm$ standard error averaged over three independent runs.

Baseline Methods We compare our approach (denoted as 'ICM + A3C') against (a) vanilla 'A3C' with ϵ-greedy exploration; (b) 'ICM-pixels + A3C' where the next observation is predicted in the pixel space instead of the inverse model feature space (see supplementary for details). (c) 'ICM-aenc + A3C' where the curiosity is computed using the features of pixel-based forward model. This baseline is representative of previous autoencoder based methods (Schmidhuber, 2010; Stadie et al., 2015); (d) state-of-the-art VIME (Houthooft et al., 2016) method.

4. Experiments

Three broad settings are evaluated: a) sparse extrinsic reward on reaching a goal (Section 4.1); b) exploration with no extrinsic reward (Section 4.2); and c) generalization to novel scenarios (Section 4.3). Generalization is evaluated on a novel map with novel textures in *VizDoom* and on subsequent game levels in *Mario*.

4.1. Sparse Extrinsic Reward Setting

In the 'DoomMyWayHome-v0' 3D navigation setup (see section 3), the agent is provided with a sparse extrinsic reward only when it reaches the goal located at a fixed location. We systematically varied the difficulty of this task and constructed "dense", "sparse" and "very-sparse" reward (see Figure 4b) scenarios by varying the distance between the initial spawning location of the agent and the location of the goal. In the "dense" reward case, the agent is randomly spawned in any of the 17 spawning locations uniformly distributed across the map. This is not a hard exploration task because sometimes the agent is randomly initialized close to the goal and therefore by random ϵ-greedy exploration it can reach the goal with reasonably high probability. In the "sparse" and "very sparse" reward cases, the agent is always spawned in Room-13 and Room-17 respectively which are 270 and 350 steps away from the goal under an optimal policy. A long sequence of directed actions is required to reach the goals from these rooms, making these settings hard goal directed exploration problems.

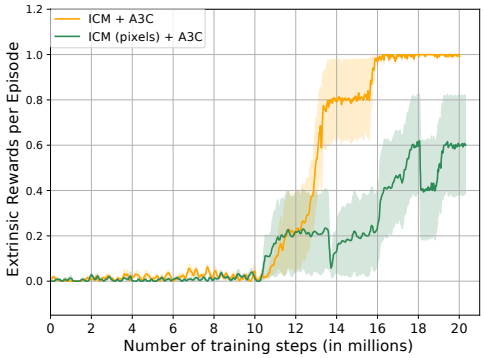


Figure 6. Evaluating the robustness of ICM when 40% of the agent's visual observation was replaced white noise (i.e. uncontrollable distractor; see Figure 3b). While ICM succeeds most of the times, the pixel prediction model struggles.

Results in Figure 5 show that curious agents learn much faster indicating that their exploration is more effective than ϵ-greedy exploration of the baseline agent. One possible explanation of the inferior performance of ICM-pixels in comparison to ICM is that in every episode the agent is spawned in one out of seventeen rooms with different textures. It is hard to learn a pixel-prediction model as the number of textures increases.

In the "sparse" reward case, as expected, the baseline A3C agent fails to solve the task, while the curious A3C agent is able to learn the task quickly. Note that ICM-pixels and ICM have similar convergence because, with a fixed spawning location of the agent, the ICM-pixels encounters the same textures at the starting of each episode which makes learning the pixel-prediction model easier as com-

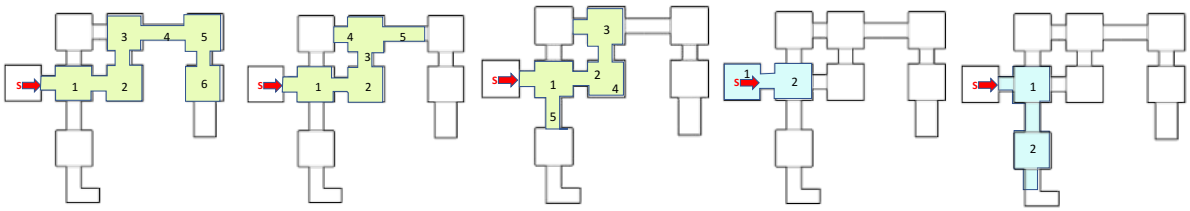


Figure 7. Each column in the figure shows the coverage of an agent by coloring the rooms it visits during 2100 steps of exploration. The red arrow shows the initial location and orientation of the agent at the start of the episode. The first three (in green) and the last two columns (in blue) show visitation of curious (ICM) and randomly exploring agents respectively. The results clearly show that the curious agent trained with intrinsic rewards explores a significantly larger number of rooms as compared to a randomly exploring agent.

pared to the "dense" reward case. Finally, in the "very sparse" reward case, both the A3C agent and ICM-pixels never succeed, while the ICM agent achieves a perfect score in 66% of the random runs. This indicates that ICM is better suited than ICM-pixels and vanilla A3C for hard goal directed exploration tasks.

Robustness to uncontrollable dynamics For testing this, we augmented the agent's observation with a fixed region of white noise which made up 40% of the image (see Figure 3b) and evaluated on "sparse" reward setup of *VizDoom*. In navigation, ideally the agent should be unaffected by this noise as the noise does not affect the agent in anyway and is merely a nuisance. Figure 6 shows that while the proposed ICM agent achieves a perfect score, ICM-pixels suffers significantly despite having succeeded at the "sparse reward" task when the inputs were not augmented with any noise (see Figure 5b). This indicates that in contrast to ICM-pixels, ICM is insensitive to nuisance changes in the environment.

Comparison to other baselines One possible reason for superior performance of the curious agent is that the intrinsic reward signal is simply acting as a regularizer by providing random rewards that push the agent out of the local minima. We systematically tested this hypothesis using many different random reward distributions on the "sparse VizDoom" task and found that with just random rewards the agents fail on sparse reward tasks. Please see supplementary materials for more details. Comparison to the state of the art TRPO-VIME (Houthooft et al., 2016) agent in the table below shows that the ICM agent is superior in performance. The hyper-parameters and accuracy for TRPO and VIME agents follow from the concurrent work (Fu et al., 2017).

Method ("sparse" reward setup)	Mean (Median) Score (at convergence)
TRPO	26.0 % (0.0 %)
A3C	0.0 % (0.0 %)
VIME + TRPO	46.1 % (27.1 %)
ICM + A3C	**100.0** % (100.0 %)

4.2. No Reward Setting

For investigating how well does the ICM agent explore the environment, we trained it on *VizDoom* and *Mario* without any rewards from the environment. We then evaluated how much of the map was visited in *VizDoom* and how much progress the agent made on *Mario*. To our surprise, we have found that in both cases, the no-reward agent was able to perform quite well (see video at `http://pathak22.github.io/noreward_rl/`).

VizDoom: Coverage during Exploration. An agent trained with no extrinsic rewards was able to learn to navigate corridors, walk between rooms, and explore many rooms in the 3D Doom environment. On many occasions, the agent traversed the entire map and reached rooms that were farthest away from the room it was initialized in. Given that the episode terminates in 2100 steps and farthest rooms are over 250 steps away (for an optimally-moving agent), this result is quite remarkable, demonstrating that it is possible to learn useful skills without the requirement of any external supervision of rewards. Example explorations are shown in Figure 7. The first 3 maps show our agent explores a much larger state space without any extrinsic signal, compared to a random exploration agent (last 2 maps).

Mario: Learning to play with no rewards. Without any extrinsic reward from environment, our Mario agent can learn to cross over 30% of Level-1. The agent received no reward for killing or dodging enemies or avoiding fatal events, yet it automatically discovered these behaviors (see video). One possible reason is that getting killed by the enemy will result in only seeing a small part of the game space, making its curiosity saturate. In order to remain curious, it is in the agent's interest to learn how to kill and dodge enemies so that it can reach new parts of the game space. This suggests that curiosity provides indirect supervision for learning interesting behaviors.

To the best of our knowledge, this is the first work to show that the agent learns to navigate a 3D environment and discovers how to play a game directly from pixels without any extrinsic reward. Prior works (Mirowski et al., 2017; Mnih et al., 2016) have trained agents for navigation and ATARI games from pixels, but using rewards from environment.

4.3. Generalization to Novel Scenarios

In the previous section, we showed that our agent learns to explore large parts of the space where its curiosity-driven exploration policy was trained. However it remains unclear, when exploring a space, how much of the learned behavior is specific to that particular space and how much is general enough to be useful in novel scenarios? To investigate this question, we train a no reward exploratory behavior in one scenario (e.g. Level-1 of Mario) and then evaluate the resulting exploration policy in three different ways: a) apply the learned policy "as is" to a new scenario; b) adapt the policy by fine-tuning with curiosity reward only; c) adapt the policy to maximize some extrinsic reward. Happily, in all three cases, we observe some promising generalization results:

Evaluate "as is": The distance covered by the agent on Levels 1, 2, and 3 when the policy learned by maximizing curiosity on Level-1 of *Mario* is executed without any adaptation is reported in Table 1. The agent performs surprisingly well on Level 3, suggesting good generalization, despite the fact that Level-3 has different structures and enemies compared to Level-1. However, note that the running "as is" on Level-2 does not do well. At first, this seems to contradict the generalization results on Level-3. However, note that Level-3 has similar global visual appearance (day world with sunlight) to Level-1, whereas Level-2 is significantly different (night world). If this is indeed the issue, then it should be possible to quickly adapt the agent's exploration policy to Level-2 with a little bit of "fine-tuning".

Fine-tuning with curiosity only: From Table 1, we see that when the agent pre-trained (using only curiosity as reward) on Level-1 is fine-tuned (using only curiosity as reward) on Level-2 it quickly overcomes the mismatch in global visual appearance and achieves a higher score than training from scratch with the same number of iterations. Interestingly, training "from scratch" on Level-2 is worse than the fine-tuned policy, even when training for more iterations than pre-training + fine-tuning combined. One possible reason is that Level-2 is more difficult than Level-1, so learning the basic skills such as moving, jumping, and killing enemies from scratch is harder than in the relative "safety" of Level-1. This result, therefore, might suggest that first pre-training on an earlier level and then fine-tuning on a later one produces a form of curriculum which aids learning and generalization. In other words, the agent is able to use the knowledge it acquired by playing Level-1 to better explore the subsequent levels. Of course, the game designers do this on purpose to allow the human players to gradually learn to play the game.

However, interestingly, fine-tuning the exploration policy pre-trained on Level-1 to Level-3 deteriorates the performance, compared to running "as is". This is because Level-3 is very hard for the agent to cross beyond a certain point – the agent hits a curiosity blockade and is unable to make any progress. As the agent has already learned about parts of the environment before the hard point, it receives almost no curiosity reward and as a result it attempts to update its policy with almost zero intrinsic rewards and the policy slowly degenerates. This behavior is vaguely analogous to boredom, where if the agent is unable to make progress it gets bored and stops exploring.

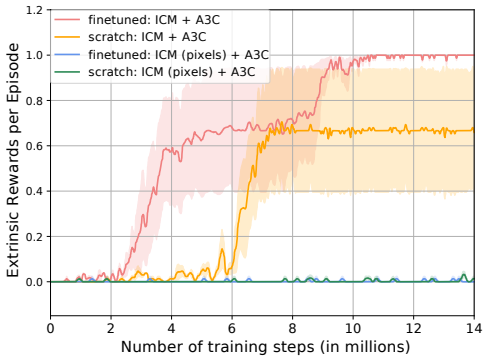


Figure 8. Curiosity pre-trained ICM + A3C when finetuned on the test map with environmental rewards outperforms ICM + A3C trained from scratch using both environmental and curiosity reward on the "very sparse" reward setting of *VizDoom*. The pixel prediction based ICM agent completely fails indicating that our curiosity formulation learns generalizable exploration policies.

Fine-tuning with extrinsic rewards: We first pre-trained an agent on *VizDoom* using only curiosity reward on the map shown in Figure 4a. We then test on the "very sparse" reward setting of 'DoomMyWayHome-v0' environment which uses a different map with novel textures (see Figure 4b). Results in Figure 8 show that the curiosity pre-trained ICM agent when fine-tuned with external rewards learns faster and achieves higher reward than an ICM agent trained from scratch to jointly maximize curiosity and the external rewards. This result confirms that the learned exploratory behavior is also useful when the agent is required to achieve goals in a new environment. It is also worth noting that ICM-pixels does not generalize to the test environment. This indicates that the proposed mechanism of measuring curiosity is significantly better for learning skills that generalize as compared to measuring curiosity in the raw sensory space. This is further consolidated by a similar result in "sparse" scenario (see supplementary).

5. Related Work

Curiosity-driven exploration is a well studied topic in the reinforcement learning literature and a good summary can be found in (Oudeyer & Kaplan, 2009; Oudeyer et al.,

Level Ids	Level-1	Level-2				Level-3			
Accuracy Iterations	Scratch 1.5M	Run as is 0	Fine-tuned 1.5M	Scratch 1.5M	Scratch 3.5M	Run as is 0	Fine-tuned 1.5M	Scratch 1.5M	Scratch 5.0M
Mean $\pm$ stderr	711 ± 59.3	31.9 ± 4.2	466 ± 37.9	399.7 ± 22.5	455.5 ± 33.4	319.3 ± 9.7	97.5 ± 17.4	11.8 ± 3.3	42.2 ± 6.4
% distance > 200	50.0 ± 0.0	0	64.2 ± 5.6	88.2 ± 3.3	69.6 ± 5.7	50.0 ± 0.0	1.5 ± 1.4	0	0
% distance > 400	35.0 ± 4.1	0	63.6 ± 6.6	33.2 ± 7.1	51.9 ± 5.7	8.4 ± 2.8	0	0	0
% distance > 600	35.8 ± 4.5	0	42.6 ± 6.1	14.9 ± 4.4	28.1 ± 5.4	0	0	0	0

Table 1. Quantitative evaluation of the policy learnt on Level-1 of Mario using only curiosity without any reward from the game when run "as is" or when further fine-tuned on subsequent levels. The performance is compared against the Mario agent trained from scratch in Level-2,3 using only curiosity without any extrinsic rewards. Evaluation metric is based on the distance covered by the Mario agent.

2007). Schmidhuber (1991; 2010) and Sun et al. (2011) use surprise and compression progress as intrinsic rewards. Classic work of Kearns et al. (1999) and Brafman et al. (2002) propose exploration algorithms polynomial in the number of state space parameters. Others have used empowerment, which is the information gain based on entropy of actions, as intrinsic rewards (Klyubin et al., 2005; Mohamed & Rezende, 2015). Stadie et al. (2015) use prediction error in the feature space of an auto-encoder as a measure of interesting states to explore. State visitation counts have also been investigated for exploration (Bellemare et al., 2016; Oh et al., 2015; Tang et al., 2016). Osband et al. (2016) train multiple value functions and makes use of bootstrapping and Thompson sampling for exploration. Many approaches measure information gain for exploration (Little & Sommer, 2014; Still & Precup, 2012; Storck et al., 1995). Houthooft et al. (2016) use an exploration strategy that maximizes information gain about the agent's belief of the environment's dynamics. Our approach of jointly training forward and inverse models for learning a feature space has similarities to (Agrawal et al., 2016; Jordan & Rumelhart, 1992; Wolpert et al., 1995), but these works use the learned models of dynamics for planning a sequence of actions instead of exploration. The idea of using a proxy task to learn a semantic feature embedding has been used in a number of works on self-supervised learning in computer vision (Agrawal et al., 2015; Doersch et al., 2015; Goroshin et al., 2015; Jayaraman & Grauman, 2015; Pathak et al., 2016; Wang & Gupta, 2015).

Concurrent work: A number of interesting related papers have appeared on Arxiv while the present work was in submission. Sukhbaatar et al. (2017) generates supervision for pre-training via asymmetric self-play between two agents to improve data efficiency during fine-tuning. Several methods propose improving data efficiency of RL algorithms using self-supervised prediction based auxiliary tasks (Jaderberg et al., 2017; Shelhamer et al., 2017). Fu et al. (2017) learn discriminative models, and Gregor et al. (2017) use empowerment based measure to tackle exploration in sparse reward setups. However, none of these works show learning without extrinsic rewards or generalization of policy to novel scenarios.

6. Discussion

In this work, we propose a mechanism for generating curiosity-driven intrinsic reward signal that scales to high dimensional visual inputs, bypasses the difficult problem of predicting pixels, and ensures that the exploration strategy of the agent is unaffected by nuisance factors in the environment. We demonstrate that our agent significantly outperforms the baseline methods.

In *VizDoom*, our agent learns the exploration behavior of moving along corridors and across rooms without any rewards from the environment. In *Mario* our agent crosses more than 30% of Level-1 without any rewards from the game. One reason why our agent is unable to go beyond this limit is the presence of a pit at 38% of the game that requires a very specific sequence of 15-20 key presses in order to jump across it. If the agent is unable to execute this sequence, it falls in the pit and dies, receiving no further rewards from the environment. Therefore it receives no gradient information indicating that there is a world beyond the pit that could potentially be explored. This issue is somewhat orthogonal to developing models of curiosity, but presents a challenging problem for policy learning.

It is common practice to evaluate reinforcement learning approaches in the same environment that was used for training. However, we feel that it is also important to evaluate on a separate "testing set" as well. This allows us to gauge how much of what has been learned is specific to the training environment (i.e. memorized), and how much might constitute "generalizable skills" that could be applied to new settings. In this paper, we evaluate generalization in two ways: 1) by applying the learned policy to a new scenario "as is" (no further learning), and 2) by fine-tuning the learned policy on a new scenario (we borrow the pre-training/fine-tuning nomenclature from the deep feature learning literature). We believe that evaluating generalization is a valuable tool and will allow the community to better understand the performance of various reinforcement learning algorithms. To further aid in this effort, we will make the code for our algorithm, as well as testing and environment setups freely available online.

Acknowledgements

We would like to thank Sergey Levine, Evan Shelhamer, Georgia Gkioxari, Saurabh Gupta, Phillip Isola and other members of the BAIR lab for fruitful discussions and comments. We thank Jacob Huh for help with Figure 2 and Alexey Dosovitskiy for VizDoom maps. This work was supported in part by NSF IIS-1212798, IIS-1427425, IIS-1536003, IIS-1633310, ONR MURI N00014-14-1-0671, Berkeley DeepDrive, equipment grant from Nvidia, NVIDIA Graduate Fellowship to DP, and the Valrhona Reinforcement Learning Fellowship.

References

Agrawal, Pulkit, Carreira, Joao, and Malik, Jitendra. Learning to see by moving. In *ICCV*, 2015.

Agrawal, Pulkit, Nair, Ashvin, Abbeel, Pieter, Malik, Jitendra, and Levine, Sergey. Learning to poke by poking: Experiential learning of intuitive physics. *NIPS*, 2016.

Bellemare, Marc, Srinivasan, Sriram, Ostrovski, Georg, Schaul, Tom, Saxton, David, and Munos, Remi. Unifying count-based exploration and intrinsic motivation. In *NIPS*, 2016.

Brafman, Ronen I and Tennenholtz, Moshe. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *JMLR*, 2002.

Brockman, Greg, Cheung, Vicki, Pettersson, Ludwig, Schneider, Jonas, Schulman, John, Tang, Jie, and Zaremba, Wojciech. Openai gym. *arXiv:1606.01540*, 2016.

Doersch, Carl, Gupta, Abhinav, and Efros, Alexei A. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

Dosovitskiy, Alexey and Koltun, Vladlen. Learning to act by predicting the future. *ICLR*, 2016.

Fu, Justin, Co-Reyes, John D, and Levine, Sergey. Ex2: Exploration with exemplar models for deep reinforcement learning. *arXiv:1703.01260*, 2017.

Goroshin, Ross, Bruna, Joan, Tompson, Jonathan, Eigen, David, and LeCun, Yann. Unsupervised feature learning from temporal data. *arXiv:1504.02518*, 2015.

Gregor, Karol, Rezende, Danilo Jimenez, and Wierstra, Daan. Variational intrinsic control. *ICLR Workshop*, 2017.

Houthooft, Rein, Chen, Xi, Duan, Yan, Schulman, John, De Turck, Filip, and Abbeel, Pieter. Vime: Variational information maximizing exploration. In *NIPS*, 2016.

Jaderberg, Max, Mnih, Volodymyr, Czarnecki, Wojciech Marian, Schaul, Tom, Leibo, Joel Z, Silver, David, and Kavukcuoglu, Koray. Reinforcement learning with unsupervised auxiliary tasks. *ICLR*, 2017.

Jayaraman, Dinesh and Grauman, Kristen. Learning image representations tied to ego-motion. In *ICCV*, 2015.

Jordan, Michael I and Rumelhart, David E. Forward models: Supervised learning with a distal teacher. *Cognitive science*, 1992.

Kearns, Michael and Koller, Daphne. Efficient reinforcement learning in factored mdps. In *IJCAI*, 1999.

Kempka, Michał, Wydmuch, Marek, Runc, Grzegorz, Toczek, Jakub, and Jaśkowski, Wojciech. Vizdoom: A doom-based ai research platform for visual reinforcement learning. *arXiv:1605.02097*, 2016.

Klyubin, Alexander S, Polani, Daniel, and Nehaniv, Chrystopher L. Empowerment: A universal agent-centric measure of control. In *Evolutionary Computation*, 2005.

Lillicrap, Timothy P, Hunt, Jonathan J, Pritzel, Alexander, Heess, Nicolas, Erez, Tom, Tassa, Yuval, Silver, David, and Wierstra, Daan. Continuous control with deep reinforcement learning. *ICLR*, 2016.

Little, Daniel Y and Sommer, Friedrich T. Learning and exploration in action-perception loops. *Closing the Loop Around Neural Systems*, 2014.

Lopes, Manuel, Lang, Tobias, Toussaint, Marc, and Oudeyer, Pierre-Yves. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *NIPS*, 2012.

Mirowski, Piotr, Pascanu, Razvan, Viola, Fabio, Soyer, Hubert, Ballard, Andy, Banino, Andrea, Denil, Misha, Goroshin, Ross, Sifre, Laurent, Kavukcuoglu, Koray, et al. Learning to navigate in complex environments. *ICLR*, 2017.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A, Veness, Joel, Bellemare, Marc G, Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K, Ostrovski, Georg, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

Mnih, Volodymyr, Badia, Adria Puigdomenech, Mirza, Mehdi, Graves, Alex, Lillicrap, Timothy P, Harley, Tim, Silver, David, and Kavukcuoglu, Koray. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016.

Mohamed, Shakir and Rezende, Danilo Jimenez. Variational information maximisation for intrinsically motivated reinforcement learning. In *NIPS*, 2015.

Oh, Junhyuk, Guo, Xiaoxiao, Lee, Honglak, Lewis, Richard L, and Singh, Satinder. Action-conditional video prediction using deep networks in atari games. In *NIPS*, 2015.

Osband, Ian, Blundell, Charles, Pritzel, Alexander, and Van Roy, Benjamin. Deep exploration via bootstrapped dqn. In *NIPS*, 2016.

Oudeyer, Pierre-Yves and Kaplan, Frederic. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 2009.

Oudeyer, Pierre-Yves, Kaplan, Frdric, and Hafner, Verena V. Intrinsic motivation systems for autonomous mental development. *Evolutionary Computation*, 2007.

Paquette, Philip. Super mario bros. in openai gym. *github:ppaquette/gym-super-mario*, 2016.

Pathak, Deepak, Krahenbuhl, Philipp, Donahue, Jeff, Darrell, Trevor, and Efros, Alexei A. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

Poupart, Pascal, Vlassis, Nikos, Hoey, Jesse, and Regan, Kevin. An analytic solution to discrete bayesian reinforcement learning. In *ICML*, 2006.

Ryan, Richard; Deci, Edward L. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 2000.

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

Schmidhuber, Jürgen. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2010.

Shelhamer, Evan, Mahmoudieh, Parsa, Argus, Max, and Darrell, Trevor. Loss is its own reward: Self-supervision for reinforcement learning. *arXiv:1612.07307*, 2017.

Silvia, Paul J. Curiosity and motivation. In *The Oxford Handbook of Human Motivation*, 2012.

Singh, Satinder P, Barto, Andrew G, and Chentanez, Nuttapong. Intrinsically motivated reinforcement learning. In *NIPS*, 2005.

Stadie, Bradly C, Levine, Sergey, and Abbeel, Pieter. Incentivizing exploration in reinforcement learning with deep predictive models. *NIPS Workshop*, 2015.

Still, Susanne and Precup, Doina. An information-theoretic approach to curiosity-driven reinforcement learning. *Theory in Biosciences*, 2012.

Storck, Jan, Hochreiter, Sepp, and Schmidhuber, Jürgen. Reinforcement driven information acquisition in non-deterministic environments. In *ICANN*, 1995.

Sukhbaatar, Sainbayar, Kostrikov, Ilya, Szlam, Arthur, and Fergus, Rob. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv:1703.05407*, 2017.

Sun, Yi, Gomez, Faustino, and Schmidhuber, Jürgen. Planning to be surprised: Optimal bayesian exploration in dynamic environments. In *AGI*, 2011.

Tang, Haoran, Houthooft, Rein, Foote, Davis, Stooke, Adam, Chen, Xi, Duan, Yan, Schulman, John, De Turck, Filip, and Abbeel, Pieter. # exploration: A study of count-based exploration for deep reinforcement learning. *arXiv:1611.04717*, 2016.

Wang, Xiaolong and Gupta, Abhinav. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

Wolpert, Daniel M, Ghahramani, Zoubin, and Jordan, Michael I. An internal model for sensorimotor integration. *Science-AAAS-Weekly Paper Edition*, 1995.