Novelty Detection in Reinforcement Learning with World Models

Anonymous authors
Paper under double-blind review

Abstract

Reinforcement learning (RL) using world models has found significant recent successes. However, when a sudden change to world mechanics or properties occurs then agent performance and reliability can dramatically decline. We refer to the sudden change in visual properties or state transitions as *novelties*. Implementing novelty detection within generated world model frameworks is a crucial task for protecting the agent when deployed. In this paper, we propose straightforward bounding approaches to incorporate novelty detection into world model RL agents, by utilizing the misalignment of the world model's hallucinated states and the true observed states as a novelty score. We provide effective approaches to detecting novelties in a distribution of transitions learned by an agent in a world model. Finally, we show the advantage of our work in a novel environment compared to traditional machine learning novelty detection methods as well as currently accepted RL focused novelty detection algorithms.

1 Introduction

Reinforcement learning (RL) using world models has found significant recent successes due to its strength in sampling efficiency and ability to incorporate well-studied Markov Decision Process techniques (Moerland et al., 2020; Robine et al., 2021). A *world model* (Ha & Schmidhuber, 2018b) is a model that predicts the world state given a current state and the execution that was executed (or will be executed).

In this paper, we address novelty detection, a form of anomaly detection, which is relatively unexplored in reinforcement learning Müller et al. (2022). *Novelties* are sudden changes to the observation space or environment state transition dynamics that occur *at inference time* that are unanticipated (or unanticipatable) by the agent during training (Balloch et al., 2022). Unlike anomalies, novelties represent a permanent shift in observation space or environment state transition dynamics, as opposed to a one-off sensory error or transition error. In cases where there is a novel change, the RL agent's converged policy is no longer reliable and the agent can make catastrophic mistakes. In some cases, the agent may just flounder ineffectually. But in other cases, the agent can mistakenly take action that put itself in harms way or put others in harms way. While RL agents are often trained and evaluated in environments with stationary transition functions, the real world can undergo distributional shifts in the underlying transition dynamics.

There are several ways of addressing inference-time novelty, depending on the nature of the agent's task. One may halt execution because continuing to run the policy has become potentially dangerous, especially in the case of robotic platform, but also agent interacting with people, financial systems, etc. Once halted, an operator can figure out the best way to retrain the agent. One may, alternatively, attempt *novelty adaptation* where the agent attempts to update its own policy during online inference time (Zhao et al., 2019; Wilson & Cook, 2020).

Prior to halting or adapting, however, is to *detect* that novelty has occurred. The standard approach—as exemplified by an approach based on recurrent implicit quantile networks (RIQN) (Danesh & Fern, 2021)—is to treat novelty as a distribution shift in either the observation

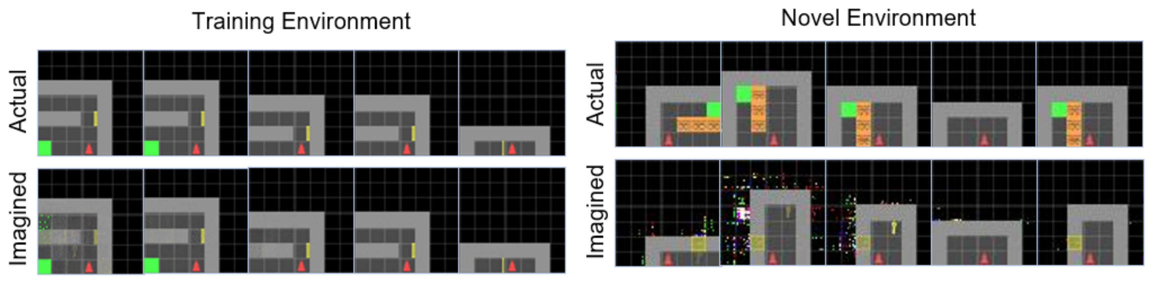

Figure 1: A world model-based reinforcement learning agent, DreamerV2 (Hafner et al., 2020), operating in nominal conditions (Left) and after experiencing a novelty (Right). Under nominal conditions the agent's predicted state closely matches the ground-truth state. After experiencing novelty, the agent's predicted state can radically diverge.

space or the state-action transition probability. This traditionally requires one to set a threshold hyperparameter to determine when distribution shift is significant enough to be considered a novelty. Setting such a parameter requires at least implicit understanding of what novelties can occur or examples of novelties against which to tune the parameter.

We introduce a novel technique to novelty detection *without* the need to manually specify thresholds, and *without* the need of additional augmented data. Our technique calculates a **novelty threshold bound** without additional hyper-parameters as a form of *Bayesian surprise* (Itti & Baldi, 2005). To motivate the work, consider Figure 1. On the top, we see normal operation of the agent in a world that has not deviated significantly from the training environment. The first row is the agent's actual state, and the second row is the agent's "imagined" state, as predicted by the world model. The bottom half of the figure shows the same agent, which was trained to convergence and a high-rate of task completion, encountering novelty. The agent's "imagined" state radically diverges from the actual states encountered as the agent executes its policy.

We evaluate our method on a variation of the MiniGrid environment (Chevalier-Boisvert et al., 2018) that has been updated to inject a variety of novelties quickly during testing time (Balloch et al., 2022). Due to the dearth of established novelty detection techniques in reinforcement learning, we compare to the one state-of-the-art RL novelty detection technique of recurrent implicit quantile networks as well as ablations of our technique that more closely resemble standard assumptions about detection; showing the possible dangers of the necessity of finetuning a threshold fit for a novelty detector.

2 Related Work

Novelty detection has taken on different names depending on the context (Pimentel et al., 2014). There are important applications that can benefit from RL novelty detection frameworks (Fu et al., 2017), and yet novelty detection has not been well-studied to date in the realm of RL.

Many complications arise from applying traditional machine learning novelty detection methods to an online setting (Müller et al., 2022). Generalization techniques exist, such as procedural generation to train agents to be more robust to novel situations (Cobbe et al., 2019) and data augmentation (Lee et al., 2019) to better train the agent, however these techniques suffer from high sample complexity (Müller et al., 2022). Furthermore, there may not be a sufficient representation the agent's evaluation environment to be able to detect novel Out-of-Distribution (OOD) transitions. Recent research shows that learning an OOD model is a difficult problem to solve and may even prove to be impossible in certain situations (Zhang et al., 2021) (Fang et al., 2022).

Some methods attempt to use a measure of reward signal deterioration as a way to detect if a novelty has occurred, but that can prove to be dangerous in a high stakes environment where it may be too late to recover from a negative signal.(Greenberg & Mannor, 2020) This method may also prove to have am incorrect mapping to the novelty that caused the reward shift.

The RIQN (Danesh & Fern, 2021) framework has proven to be successful by focusing on the aleoric uncertainty of the agent, and draws upon the generated distributions of Implicit Quartille networks (Dabney et al., 2018). Modeling after IQN, RIQN detects novelty by predicting action values from state-action pairs as input, and predicts the feature distribution at each time step, based on the previous values of the features.

3 Background

Partially observable Markov Decision Processes We study episodic Partially Observable Markov decision processes (POMDPs) denoted by the tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \Omega, O, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}$ is the transition distribution $\mathcal{T}(s_t|s_{t-1}, a_{t-1})$, r is the reward function, O is the observation space, Ω is an emissions model from ground truth states to observations, and γ is the discounting factor (Åström, 1965).

DreamerV2 World Model We conduct experiments applying our methods to the state-of-the-art DreamerV2 (Hafner et al., 2020) world model dynamics framework, due to its VAE and history component, which is similar to other world model architectures Ha & Schmidhuber (2018a)Ha & Schmidhuber (2018b). DreamerV2 learns a policy by first learning a world model and then using the world model to roll out trials to train a policy model. The framework is composed of an image autoencoder and a Recurrent State-Space Model (RSSM). Relevant components of DreamerV2 that we will use throughout the paper are: x_t: the current image observation, h_t: the encoded history of the agent, z_t: an encoding of the current image x_t that incorporates the learned dynamics of the world, and $s_t = (h_t, z_t)$: the agent's compact model state. Given each representation of state s_t, DreamerV2 defines six other learned, conditionally-independent transition distributions given by the trained world model:

$$\text{DreamerV2:} \begin{cases} \text{Recurrent model:} h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1}) \\ \text{Representation model: } p_\phi(z_t|h_t, x_t) \\ \text{Transition prediction model: } p_\phi(\hat{z}_t|h_t) \\ \text{Image prediction model: } p_\phi(\hat{x_t}|h_t, z_t) \\ \text{Reward prediction model: } p_\phi(\hat{r_t}|h_t, z_t) \\ \text{Discount prediction model: } p_\phi(\hat{\gamma}_t|h_t, z_t) \end{cases} \tag{1}$$

where ϕ describes the parameter vector for all distributions optimized. The loss function during training (Hafner et al., 2020) is:

$$\begin{aligned} \mathcal{L}(\phi) = \mathop{\mathbb{E}}_{p_\phi(z|a,x)} \Bigg[\sum_t^T & - \ln p_\phi(x_t|h_t, z_t) - \ln p_\phi(r_t|h_t, z_t) - \ln p_\phi(\gamma_t|h_t, z_t) \\ & + \beta KL \left[p_\phi(z_t|h_t, x_t) || p_\phi(z_t|h_t) \right] \Bigg] \end{aligned} \tag{2}$$

where $\beta KL[p_\phi(z_t|h_t, x_t)||p_\phi(z_t|h_t)]$ is minimized by improving the prior dynamics towards the more informed posterior through KL Balancing (Hafner et al., 2020).

The goal of the DreamerV2 world model is to learn the dynamics, and predictors for the observation x_t, reward and discount factor of the training environment. We consider a single agent online RL setting, where an agent at time t will traverse through each state s_t in which observations are represented in the form of x_t. The agent will rely on these representations along with r_t to construct its belief state and make decisions to achieve the optimal discounted sum of rewards.

We consider an agent that has trained in a stationary training environment and has been deployed to a non-stationary evaluation environment that undergoes a change that is *a priori* unknown and unanticipated by during agent development and training.

4 Detection Method

For latent-based detection, we investigate the world model and its learned probabilities. Our goal is to **safely halt** the agent in response to all possible Markovian novel observations. Therefore, we do not consider techniques that are reliant on reward deterioration, as those methods can involve temporally extended action sequences to be conducted between sparse rewards. Even with dense rewards, novelty may only be detected after some amount of time of reward decrease. We also do not consider generalization techniques that are aimed to model possible novelties that the agent may experience, as it is generally impossible to construct a model that generalizes towards every unseen stimuli (Wang et al., 2020).

We establish a bound that tests the latent space model's current performance. The KL loss equation (2) implies that, as training progresses, the divergence between the latent representation of the hidden state given the latent current state representation with and without the direct observation goes to zero for local transitions, i.e., $KL[p_\phi(z_t|h_t, x_t)||p_\phi(z_t|h_t)] \leq \epsilon$. Given the current training mechanism in place, learning the transition distribution may be difficult due to the task of avoiding regularizing the representations toward a poorly trained prior Hafner et al. (2020).

A direct consequence of using common KL balancing techniques such as in Equation (2) to bias the loss towards the prior error is that the world model is initially reliant on ground truth x_t in contrast to a noisy h_t. This stabilizes $p(z_t|h_t, x_t)$ for proper training of $p(z_t|h_t)$ and $p(\hat{x_t}|h_t, z_t)$ (Asperti & Trentin, 2020) until the model understands the role of history h_t. Since the world model does not directly measure this relationship with x_t, we introduce the **cross entropy score comparison**:

$$H(p_\phi(z_t|h_t, x_t), p_\phi(z_t|h_0)) - H(p_\phi(z_t|h_t, x_t), p_\phi(z_t|h_0, x_t)) \tag{3}$$

where h_0 is simply an empty hidden state passed to the model in order to simulate the dropout of h_t and compute the influence that the ground truth x_t has on the final prediction of the distribution of z_t. This gives us an empirical measure of the world model's current reliance and improved performance given the ground truth observation x_t, given that the cross entropy score comparison *increases* if we find the entropy to be minimized with respect to h_0.

As the divergence between latent predictions with and without the ground truth observable decreases, $KL[p_\phi(z_t|h_t, x_t)||p_\phi(z_t|h_t)] \to \epsilon$, we expect a direct result of this to be the reduction of the impact of x_t. We model this relationship with the novelty detection bound:

$$\begin{aligned} KL[p_\phi(z_t|h_t, x_t)||p_\phi(z_t|h_t)] \leq & KL[p_\phi(z_t|h_t, x_t)||p_\phi(z_t|h_0)] \\ & - KL[p_\phi(z_t|h_t, x_t)||p_\phi(z_t|h_0, x_t)] \end{aligned} \tag{4}$$

where we substitute KL divergence in place of cross entropy loss to use the world model loss equation (2).

The intuition behind the bound is as follows. For an unforeseen observation $\hat{x_t}$, if this relationship becomes disturbed—that is, if the model is incapable of constructing a mapping such that the the cross entropy of $H(p_\phi(z_t|h_t, x_t), p_\phi(z_t|h_0, x_t))$ is not minimized with respect to the performance of $H(p_\phi(z_t|h_t, x_t), p_\phi(z_t|h_0))$—the threshold (the right-hand side of the equality in (4)) will decrease. But it is also expected that the left-hand side, $KL[p_\phi(z_t|h_t, x_t)||p_\phi(z_t|h_t)]$, reaches extremely high values, depending on the current reliance of h_t.

Two cases of detection arise. First, if the world model performance somehow increases without data, then the right side of (4) will become negative, which is then immediately flagged as novelty due to the convexity of the left side KL divergence. Second, if the the world model performance improves with data, then the bound becomes a dropout generalization test (Srivastava et al., 2014) of the model via a simple reordering of (4) i.e:

$$KL[p_\phi(z_t|h_t, x_t)||p_\phi(z_t|h_t)] + KL[p_\phi(z_t|h_t, x_t)||p_\phi(z_t|h_0, x_t)] \leq KL[p_\phi(z_t|h_t, x_t)||p_\phi(z_t|h_0)] \tag{5}$$

where the model is tasked to simultaneously have sufficiently good performance with the dropout of x_t and with the dropout of h_t. Therefore, our bound is both combating

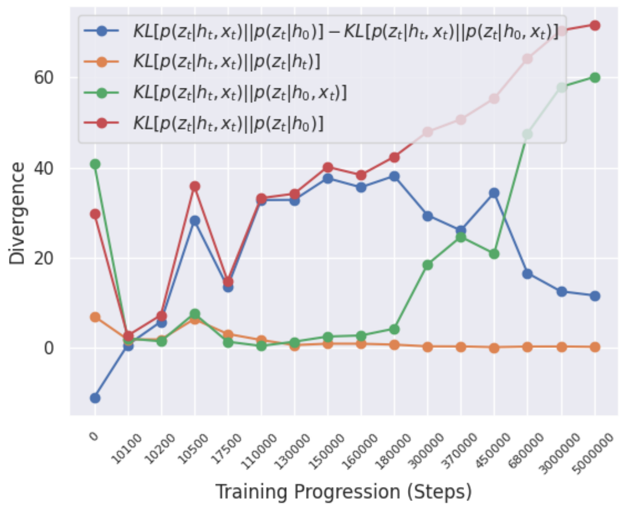

Figure 2: Visualization of the average levels of divergence: proposed bound (Blue); divergence of the RSSM predicted distributions given (h_t) and (h_t, x_t) (Orange); divergence between the RSSM given (h_t, x_t) and the RSSM given only the image (h_0, x_t) (Green) in the Nominal environment; and the divergence of the RSSM given (h_t, x_t) and receiving no data at all (h_0) (Red) as training progresses; novelty is triggered when blue is below orange.

posterior collapse i.e $P_\theta(z_t|x_t) = P_\theta(z_t|h_0)$, as well as measuring possible over-fitting i.e $KL[P_\theta(z_t|h_t, x_t)||P_\theta(z_t|h_0, x_t)] >> c$ for some large c. A properly trained model is one in which its learned representations, h_t and x_t, are useful in the prediction of distribution $p_\phi(z_t|h_t, x_t)$. Figure 2 illustrates that failure to do so may result in x_t being interpreted as a noisy signal (divergence of blue is collapses to red) or the transition prediction model h_t has not yet met sufficient performance (orange exceeds blue).

5 Experiments

To empirically evaluate our bound, experiments are conducted in the MiniGrid (Chevalier-Boisvert et al., 2018) environment. We design and implement custom environments to showcase a larger variety of non-trivial novelties than is currently available in the default MiniGrid library. We consider the detection of two types of novelties: **visual novelties** and **functional novelties**. Visual novelties are considered to be observations where the image state representation has changed to an out-of-distribution visual with respect to the original training observations such as a new color and/or an addition/removal of an object. Functional novelties are instances where the learned Markov decision process produced from the training environment has solely had a change to its state-action transition distribution and/or action space without a visual cue. Our functional novelties are:

- `BrokenDoor`: The door no longer opens, even with the key.
- `ActionFlip`: Restrict the agent to left and right turns only.
- `Teleport`: Teleport the agent randomly then perform the selected action.
- `HeavyKey`: The key can no longer be picked up.

Our visual novelties are:

- `LavaGap`: The goal is to maneuver around the lava gap.
- `Empty`: An empty grid with a goal.
- `Fetch`: The agent's goal is to collect various colored items in the environment.
- `DoorGone`: The door is removed from the environment and is replaced with empty space.

Importantly, each functional novelty environment is visually identical to the nominal environment. `MiniGrid-Doorkey-6x6` serves as our nominal case.

We first train an agent to learn a ϵ-optimal policy in the nominal environment of which the agent always gets to the goal, then transfer the agent to one of the several mini-grid environments with visual or functional novelties during testing time. We then let the trained agent take steps in each novel environment, capturing 50,000 steps within various independent and identically distributed episodes; tracking each time step where novelty is detected to imitate possible agent halting situations. The agent's trained policy is sub-optimal in the novel environments and is not re-trained at any point.

5.1 Baselines

RIQN We compare our bound against the Recurrent Implicit Quantile Network anomaly detection model (RIQN) (Danesh & Fern, 2021), a traditional and accepted RL focused novelty detection approach, and take note of some of the practical desiderata discussed in (Müller et al., 2022) to ground our evaluation techniques. We explicitly train the RIQN model using nominal transitions given by a trained policy as recommended (Danesh & Fern, 2021). We test the trained RIQN algorithm with the recommended hyper-parameters, as well as use the recommended cusum algorithm to detect novelties (Page, 1954).

CMTRE Because visual observational novelty manifests itself as reconstruction error, we also compare our method against *Classical Multi-Threshold Reconstruction Error* (CMTRE), a commonly used approach for novel observation detection in traditional machine learning (Pimentel et al., 2014). CMTRE works under the assumption that the Mean Absolute Reconstruction Error (MARE) distribution of the novelties will be greater than the the mean of the distribution of MARE visuals x_t from the original training environment, by some standard deviation between $1.0 - 0.5$. Therefore to implement CTMRE in this setting, we construct a threshold of 0.5 standard deviation above the MARE from samples of the agent's past experiences in its experience replay buffer, classifying observations with MARE above the threshold as novelty. Since the distribution of observations of a trained agent differ from the observations of an untrained agent, we experiment using three separate thresholds generated from three different means: (1) a trained agent's MARE of observations; (2) an untrained agent's MARE of observations; and (3) the equally distributed MARE of observations from both the trained and untrained agents.

PP-MARE CMTRE assumes that the training environment is stationary, which does not hold in the RL setting because the "ground truth" relies on the replay buffer (Dao & Lee, 2019), which is constantly changing. CMTRE also relies on the extent of the encoder's convergence during the storing of the reconstruction errors, which can skew thresholds if collected too early in training. Consequently, we derive a novel reconstruction error technique over the induced DreamerV2's RSSM *prior* and *posterior* reconstructions. Note that locally successful training of the world model implies that $\beta KL[p_\phi(z_t|x_t, h_t)||p_\phi(\hat{z}_t|h_t))] \leq \epsilon$. To properly utilize reconstruction error for trained world models, we induce $P(x_t|h_t, \hat{z}_t)$ from 2 where $h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1})$, $z_{t-1} \sim P(z_{t-1}|h_{t-1}, x_{t-1})$, and $\hat{z}_t \sim P(z_t|h_t)$ to compare the i pixel reconstruction losses between the generated 1-step prior and posterior x_t observations and bound the difference by a small ϵ defined by the user during deployment:

$$\frac{\sum_i^N |\hat{x}_{t prior}^i - \hat{x}_{t posterior}^i|}{N} \leq \epsilon \tag{6}$$

where $\hat{x}_{t\text{prior}} \sim p_\phi(\hat{x}_t|h_t, \hat{z}_t)$, and $\hat{x}_{t\text{posterior}} \sim p_\phi(\hat{x}_t|h_t, z_t)$. This removes direct dependence on the replay buffer by using only the final performance of the encoder and the current state (z_t, h_t). As seen in Figure 3, the prior and posterior can have dramatic changes in representation when the predicted latent representation z_t is revealed. In the absence of information about the nature of novelties to be experienced, we set $\epsilon = 1.0$ for our PP-MARE bound method during all experiments. PP-MARE represents the class of reconstruction-based baselines when correcting for the RL setting.

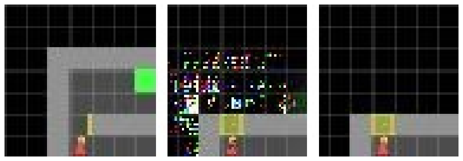
Figure 3: From left to right, $\hat{x}_{tprior}, \hat{x}_{tposterior}$ and x_t observations of agent trying to open door in the `BrokenDoor` custom minigrid environment where the door fails to open despite having the correct key.

5.2 Metrics

Average delay error is the measure of how long it takes to detect the novelty. To penalize preemptive and delayed detection i.e a false positive or false negative signal, we define *novelty detection delay error* as the absolute difference between the earliest time step that novelty is first observable and the time step that the novelty is detected. We calculate the agent's *average delay error* (ADE) by averaging the absolute delay error across all novel environment episodes. We use the original nominal training environment to test for false positives; we do not consider false negatives when in the training environment. Ideal performance primarily minimizes the average delay error and false alarm/positive rates (Müller et al., 2022).

For all environments, we also measure *precision*, *recall*, *accuracy* , and *area under the curve* (AUC), a common anomaly-detection metric which is independent of detection thresholds and depends only on the quality of the anomaly scores of a detector. Accuracy, precision and recall are optionally collected after the first sign of novelty in Appendix 2, of which consequent signs are used to measure performance. Since we mainly focus on fast halting time, we will primarily focus on average delay error and false alarm rates.

5.3 Results

Figure 4 shows average delay error for all novelty environments and overall false alarm rates when traversing the normal environment. For *visual* novelty environments, we found that `Empty` was the most difficult environment for our methods, but all methods other than RIQN have average delay error below 1 time step. All proposed methods had lower average delay error in all visual environments compared to RIQN, except `LavaGap`. However, this does not account for the fact that RIQN has a very high false negative rate in the environment. RIQN also appears to be reliant on visual features, in contrast to latent based detection, which could explain why RIQN was able to detect drastic visual changes to the enviroment such as `Empty` or `LavaGap`.

The CTMRE observational method variants could not detect functional novelties and are absent in Figure 4. CTMRE anomaly scores were found to be unable to sufficiently linearly separate the normal and visual novel observations, in comparison to PP-MARE, KL and RIQN; for illustration and further results please refer to Figure 6 in Appendix 5.

False Alarms For functional novelties, Figure 4(bottom right) shows the overall false positive rates. The KL method false positives rate is $\leq 10^{-4}\%$ and all techniques have dramatically lower false positive rates. We hypothesize that KL's high performance in avoiding false alarms coincides with the behavior observed from Figure 2, and is a direct result of the bounds formulation. RIQN's high false positive rate was also observed by Danesh & Fern (2021).

AUC Scores Table 1 provides Area Under Curve (AUC) scores for different methods on different novelty-induced environments. KL has competitively higher AUC rates on all but one functional novelty environment. It can be reasoned from the observed AUC scores that RIQN suffers the most from the inability to tune/train its detection threshold, relying on the recommended CUMSUM value of 1 to perform detection. We come to a similar conclusion to that of the authors: further tuning of

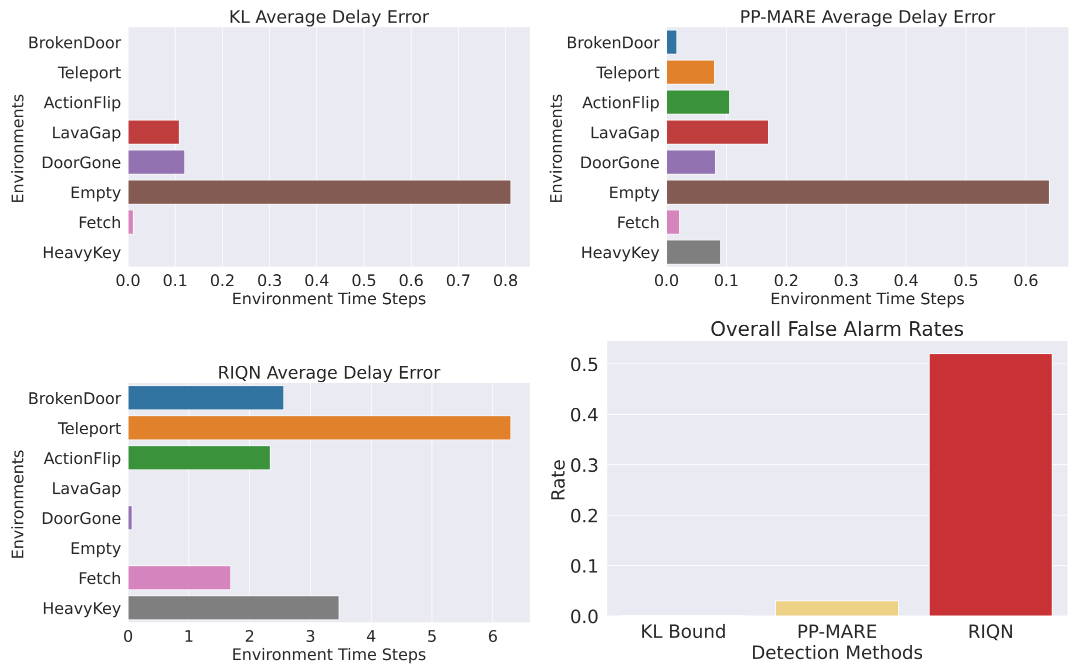

Figure 4: Average Delay Error alongside the Average False Alarm rate for all methods against all tested Minigrid environments (values below 10^{-4} are truncated). Note that KL bound and PP-MARE delay errors are less than 1, whereas RIQN delay errors are between 0 and 6.

Table 1: AUC scores for all functional and visual environments.

Environment	KL	PPMARE	RIQN	Environment	KL	PPMARE	RIQN
HeavyKey	.78	.38	1	LavaGap	.732	.765	.76
BrokenDoor	.939	.7840	.92	Empty	.811	.51721	.71
Teleport	.9919	.959	.95	Fetch	1	.99	.50
ActionFlip	.99121	.96299	.89	DoorGone	.940	.685	.60

the CUSUM algorithm may be able to achieve some improvement without significantly impacting other metrics, but it is likely that fundamentally different approaches will be needed (Danesh & Fern, 2021). PP-MARE generally falls behind KL, likely partially due to the fact that the mean reconstruction error it is based on is not a measure optimized for computing differences between the observations. Further, PP-MARE assumes a false correlation between pixels with high reconstruction error and novel regions of input images (Feeney & Hughes, 2021).

6 Conclusion

This paper proposed a novel way to construct bounds based on model characteristics alone, without the need for hyper-parameter tuning or preparation for possible unforeseen novelties. Results on the Minigrid environments designed explicitly to introduce novelties that confound an agent confirm that these successfully detect novelties compared to traditional Machine learning and RL focused detection techniques. We have learned that there exists ways to further utilize the learned world model transition distribution model to improve the debugging and safety performance for open-world environments, and believe that our methods can be used to formulate stronger bounds for the Bayesian surprise given different assumptions of the evaluation environment, as well as formulate other possible reactions once the novel transition is detected. Our paper supports the overall value of world model implementations and that world models provide a means for improved robustness, which is essential in open-worlds due to the possible degraded performance that can occur. We hope to advance and encourage further research in the field of novelty detection in reinforcement learning.

References

Andrea Asperti and Matteo Trentin. Balancing reconstruction error and kullback-leibler divergence in variational autoencoders, 2020.

Karl Johan Åström. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10:174–205, 1965.

Jonathan Balloch, Zhiyu Lin, Mustafa Hussain, Aarun Srinivas, Robert Wright, Xiangyu Peng, Julia Kim, and Mark Riedl. Novgrid: A flexible grid world for evaluating agent response to novelty, 2022. URL https://arxiv.org/abs/2203.12117.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for gymnasium, 2018. URL https://github.com/Farama-Foundation/Minigrid.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. *CoRR*, abs/1912.01588, 2019. URL http://arxiv.org/abs/1912.01588.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning, 2018. URL https://arxiv.org/abs/1806.06923.

Mohamad H Danesh and Alan Fern. Out-of-distribution dynamics detection: Rl-relevant benchmarks and results. In *International Conference on Machine Learning, Uncertainty & Robustness in Deep Learning Workshop*, 2021.

Giang Dao and Minwoo Lee. Relevant experiences in replay buffer. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 94–101, 2019. doi: 10.1109/SSCI44817.2019.9002745.

Zhen Fang, Yixuan Li, Jie Lu, Jiahua Dong, Bo Han, and Feng Liu. Is out-of-distribution detection learnable? In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=sde_7ZzGXOE.

Patrick Feeney and Michael C. Hughes. Evaluating the use of reconstruction error for novelty localization, 2021. URL https://arxiv.org/abs/2107.13379.

Justin Fu, John D. Co-Reyes, and Sergey Levine. EX2: exploration with exemplar models for deep reinforcement learning. *CoRR*, abs/1703.01260, 2017. URL http://arxiv.org/abs/1703.01260.

Ido Greenberg and Shie Mannor. Detecting rewards deterioration in episodic reinforcement learning, 2020. URL https://arxiv.org/abs/2010.11660.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems 31*, pp. 2451–2463. Curran Associates, Inc., 2018a. URL https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution. https://worldmodels.github.io.

David Ha and Jürgen Schmidhuber. World models, March 2018b. URL https://doi.org/10.5281/zenodo.1207631. Interactive version of the article at https://worldmodels.github.io.

Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *CoRR*, abs/2010.02193, 2020. URL https://arxiv.org/abs/2010.02193.

Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. In Y. Weiss, B. Schölkopf, and J. Platt (eds.), *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005. URL https://proceedings.neurips.cc/paper/2005/file/0172d289da48c48de8c5ebf3de9f7ee1-Paper.pdf.

Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. A simple randomization technique for generalization in deep reinforcement learning. *CoRR*, abs/1910.05396, 2019. URL http://arxiv.org/abs/1910.05396.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey, 2020. URL https://arxiv.org/abs/2006.16712.

Robert Müller, Steffen Illium, Thomy Phan, Tom Haider, and Claudia Linnhoff-Popien. Towards anomaly detection in reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, pp. 1799–1803, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954. ISSN 00063444. URL http://www.jstor.org/stable/2333009.

Marco A. F. Pimentel, David A. Clifton, Lei A. Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal Process.*, 99:215–249, 2014.

Jan Robine, Tobias Uelwer, and Stefan Harmeling. Smaller world models for reinforcement learning, 2021.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Rui Wang, Joel Lehman, Aditya Rawal, Jiale Zhi, Yulun Li, Jeff Clune, and Kenneth O. Stanley. Enhanced poet: Open-ended reinforcement learning through unbounded invention of learning challenges and their solutions, 2020.

Garrett Wilson and Diane J. Cook. A survey of unsupervised deep domain adaptation. *ACM Trans. Intell. Syst. Technol.*, 11(5), jul 2020. ISSN 2157-6904. doi: 10.1145/3400066. URL https://doi.org/10.1145/3400066.

Lily H. Zhang, Mark Goldstein, and Rajesh Ranganath. Understanding failures in out-of-distribution detection with deep generative models. *CoRR*, abs/2107.06908, 2021. URL https://arxiv.org/abs/2107.06908.

Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J. Gordon. On learning invariant representation for domain adaptation, 2019.

A CMTRE Comparisons

Figure 5 shows the false positives of CTMRE and PP-MARE when operating in the novelty-free `MiniGrid-Doorkey-6x6` environment. We hypothesize that the PP-MARE technique can be susceptible to generating false positives in the beginning of the episode because the DreamerV2 RSSM cannot generate reasonable predictions of x_0 due to the initial arbitrary h_0 encoding.

CTMRE incorrectly detects novelty within the first step of any episode at a lower false positive rate, but accumulates greater false positive as the episode progresses. The CTMRE method, however, also has high false negative and true positive thresholds, as shown in Figure 6.

Thus, if there is high likelihood that the agent experiences novelties with visual cues in the first steps of an episode, the CMTRE method is recommended. If visual cues are not likely to be immediately present, the PP-MARE visual detection method should be applied because PP-MARE tends to do better at minimizing false negatives in these environments but at the cost of an initially higher false positive rate. This tradeoff can be appealing if positives are likely to be an adversarial attack.

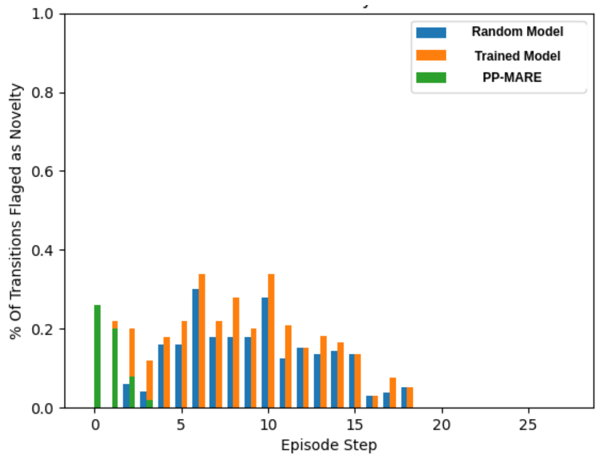

Figure 5: Percentage of transitions flagged as novelty within the original training environment, using observational based detection. We illustrate the errors of the CMTRE method using the random model threshold, trained agent threshold (as described in section Classical Multi-threshold Reconstruction Error), and Post Prior MARE using $\epsilon = 1$.

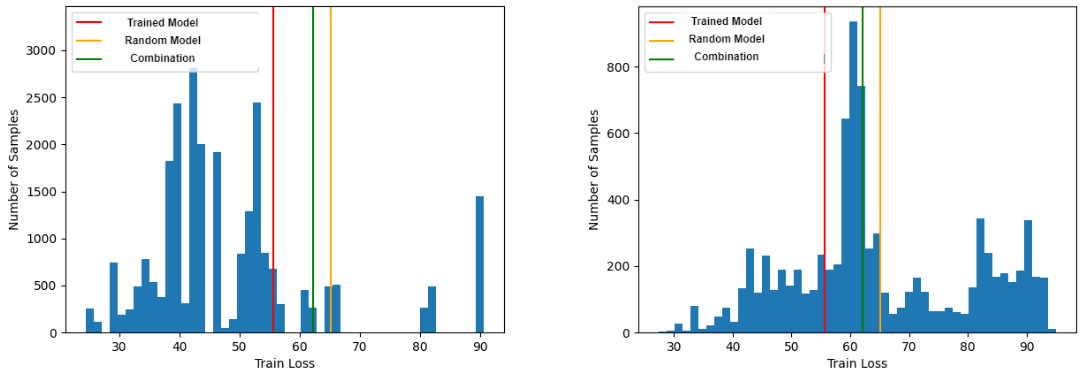

Figure 6: Reconstruction error alongside the three tested thresholds: Random Model, Trained Model and Combination model (as described in section Classical Multi-threshold Reconstruction Error). Reconstruction error from the normal `DoorKey-6x6` environment (Left), and Reconstruction error novel `LavaGap` environment (Right). Each vertical line refers to the cut off value for the corresponding threshold. Samples to the right of the line are classified as novelty. Reconstruction Error was generated from a trained agent.

B Precision, Recall, Accuracy, and AUC Results

Table 2: Precision, Recall, Accuracy, and AUC for each environment.

	Metric	**KLBound**	**PostPriorMARE**	**RIQN**	**-**
1***BrokenDoor**	**Precision**	.9908	.916	.93277	
	1***Recall**	.63	.62	.75778	
	1***Accuracy**	.811	.7810	.73171	
	1***AUC**	.939	.7840	.92	
1***ActionFlip**	**Precision**	.9999	.994	.932778	
	1***Recall**	.975	.983	.75778	
	1***Accuracy**	.9779	.9801	.63989	
	1***AUC**	.99121	.96299	.89	
1***HeavyKey**	**Precision**	.707	.023	.967	
	1***Recall**	.66	.01	.704	
	1***Accuracy**	.79	.50	.69	
	1***AUC**	.78	.38	1	
1***Teleport**	**Precision**	.9998	.9957	.8526	
	1***Recall**	.955	.97	.6011	
	1***Accuracy**	.95	.97	.580441	
	1***AUC**	.9919	.959	.95	
1***Lava**	**Precision**	.84743	.937	.7005	
	1***Recall**	.620	.9119	.696	
	1***Accuracy**	.595	.8718	.6319	
	1***AUC**	.732	.765	.76	
1***DoorGone**	**Precision**	.971	.973	.676	
	1***Recall**	.136	.395	.7004	
	1***Accuracy**	.39177	.573	.607	
	1***AUC**	.940	.685	.60	
1***Empty**	**Precision**	.603	.6033	.585	
	1***Recall**	.954	.9939	.643	
	1***Accuracy**	.599	.607	.608	
	1***AUC**	.811	.51721	.71	
1***Fetch**	**Precision**	1	.94	.6450	
	1***Recall**	.943	.99	.744	
	1***Accuracy**	.97	.97	.555	
	1***AUC**	1	.99	.50	

C Hardware requirements

All experiments can be sufficiently reproduced utilizing a NVIDIA GeForce GTX 1080 GPU with at least 8 GB of VRAM for environment complexity, a AMD Ryzen 5 5600X 6-Core Processor and at least 50 MB for files, excluding training data which is dependant on environment and model hyper-parameters. Runtime for time-steps for experiments varies between 5 and 10 hours.