Intrinsic Motivation Model Based on Reward Gating

Matej Pecháč and Igor Farkaš*

Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava, Slovak Republic

Abstract. Intrinsic motivation (IM) research is a promising part of reinforcement learning which can push artificial agents to completely new frontiers. Namely, from agents with a simple action repertoire, driven by the human engineered reward, to more autonomous agents with their own goals and skill development, able to act successfully in the environments which are unknown to their human designers. In this paper, we introduce an IM model, which combines via gating two different motivational signals: a prediction error estimated by the forward model and a predictive surprise estimated by the meta-critic. This approach accelerates the exploration of the environment and hence the agent is able to find sources of an external reward in a shorter time than the baseline agents, especially in case of sparse reward. We test this prediction using two environments with dense reward (HalfCheetah and Ant) and two with sparse reward (MountainCar and AerisNavigate), and show the superior performance of an agent with a gated reward in most cases as expected. The models are also compared using reliability measures related to dispersion and risk, calculated during learning. The source code is available at `https://github.com/Iskandor/MotivationModels`.

Keywords: Reinforcement learning · Intrinsic motivation · Prediction error · Predictive surprise · Active exploration

1 Introduction

The development of reinforcement learning (RL) methods has achieved much success over the last decade, since together with advances in computer vision [15] [11], it became possible to teach agents to solve various tasks, play simple computer games [20], even surpassing human players [19]. Nevertheless, these are still concrete single tasks. A lot of computational time has to be spent, and the agents are given a lot of resources to manage to learn the aforementioned challenges in a reasonable time. However, coping with a complex (continuous) environment such as our world is still a challenge. There are several pathways offering research opportunities. One is the search for new optimization and learning methods that would shorten the learning time or reduce the amount of resources needed. Another is hardware development, which attempts to adapt to

* This research was supported by KEGA grant no. 042UK-4/2019

the requirements of neural networks that are currently being used in the field of reinforcement learning.

The most popular approach to make RL more efficient is based on *intrinsic motivation* (IM) [2]. IM has a strong psychological motivation [25], since children acquire skills and knowledge about the world using their own drive and experience without obvious reward from the outer environment. If we want to achieve an open-ended development with artificial agents, we have to master this first step and equip them with an ability to generate their own goals and acquire new skills. Therefore, computational approaches concerned with IMs and open-ended development are thought to have the potential to lead to the construction of more intelligent artificial systems, in particular systems that are capable of improving their own skills and knowledge autonomously and indefinitely [2].

In this paper, we introduce a new version of a IM-based agent that is shown to efficiently learn the tasks at hand. It selects between two different motivation signals generated by the forward model and the meta-critic. The selection is based on simple rule performed by the gating module and its output signal is added to external reward from the environment and serves as input for critic which in turn generates the learning signal for actor.

In particular, we provide two main contributions: First, inspired by the definition of the predictive surprise motivation [22], we propose modifications to the original formula and explored its impact on the learning process of agents. Second, we explore a *gating approach* to exploit the prediction error and predictive surprise motivation signals generated in the intrinsic module of the agent. The learning models are statistically compared using the reliability measures.

2 Related work

The concept of intrinsic (and extrinsic) motivation was first studied in psychology [25], and later entered the RL literature where the first taxonomy of computational models appeared in [22]. Following this taxonomy, we can divide the concept of motivation into external and internal, depending on the mechanism that generates motivation for the agent. If the source of motivation comes from outside, we are talking about *external* motivation, and it is always associated with a particular goal in the environment. If the motivation is generated within the structures that make up the agent, it is an *internal* motivation.

Another dimension for the differentiation, extrinsic or intrinsic, is less obvious. *Extrinsic* motivations pertain to behaviors whenever an activity is done in order to attain some separable outcome. Some variability exists in this context, since these behaviors can vary in the extent to which they represent self-determination (see the details in [25]). On the other hand, *intrinsic* motivation is defined as doing an activity for its inherent satisfactions rather than for some separable consequence (or instrumental value). It has been operationally defined in various ways, backed up by different psychological theories, which point to some uncertainty in what IM exactly means. Nevertheless, Baldassarre [1] offers a solution of an operational definition of IMs as processes that can drive the

acquisition of knowledge and skills in the absence of extrinsic motivations. Furthermore, he proposes (and explains why) a new term of *epistemic motivations* as a suitable substitution for intrinsic motivations.

According to the prevailing view, the computational approaches to IM can be divided into two main categories with adaptive motivations. *Knowledge-based* approach is focused on acquisition of knowledge of the world and draws on the theory of drives, theory of cognitive dissonance and optimal incongruity theory. *Competence-based* approach focuses on acquisition of skills by motivating the agent to achieve a higher level of performance in the environment, which means to acquire desired actions to achieve self-generated goals. Its psychological basis includes the theory of effectance and the theory of flow.

The knowledge-based category is commonly divided into *prediction-based* and *novelty-based* approaches. Prediction-based approaches often use a forward model (e.g. [28,3,23]) or a variational autoencoder [14] to compute the prediction error (for more details, see [5]). The novelty-based approaches monitor the state novelty and the intrinsic signal is based on its value. The first models were based on count-based approach [31]. This method is impractical for large or continuous state spaces and it was extended by introducing pseudo-count and neural density models [21,18,17]. A similar method to pseudo-count was used by a random network distillation model [6] with a lower complexity.

It is an empirical question what is the best IM signal for a given task [26]. The difficulty increases if an agent is supposed to learn multiple skills in the shortest time. For instance, in [26] it is shown that intrinsic reinforcements purely based on the knowledge of the system are not appropriate to guide the acquisition of multiple skills and that the stronger the link between the IM signal and the competence of the system, the better the performance. Hence, the combination of both types seems to be useful. In a recent work [24] it is shown that the combination of knowledge-based and competence-based IM signals leads to more efficient exploration and task learning.

The concept of a meta-critic (MC), or a module that learns to predict the prediction error is not new in reinforcement learning; it was introduced in early 1990s within the adaptive curiosity framework [27], and has been extended in various forms since then. Also the concept of exploration has been studied intensively, one of the first being the idea of an exploration bonus [30], later analyzed in alternative ways in [10,29]. Related work on surprise-based approaches includes Bayesian bio-inspired approach where surprise measures how data affects an observer, in terms of differences between posterior and prior beliefs about the world [12]. We use a MC module in a novel role of gating two different motivational signals, based on a prediction error and predictive surprise.

3 Preliminaries

The decision making problem in the environment using RL is formalized as a Markov decision process which consists of a state space S, action space A, transition function $T(s;a;s') = p(s_{t+1} = s' | s_t = s; a_t = a)$, reward function R

and a discount factor . The main goal of the agent is to maximize the discounted return $R_t = \sum_{k=0}^{1} {}^{k} r_{t+k}$ in each state. Stochastic policy is defined as a state dependent probability function : S A ! [0; 1], such that $_t(s;a) = p(a_t = a|s_t = s)$ and $\sum_{a2A} (s;a) = 1$ and the deterministic policy : S ! A is defined as $(s) = a$.

An agent following the optimal policy maximizes the expected return R. The methods searching for the optimal policy can be divided into on-policy (family of actor–critic algorithms), and off-policy (family of Q-learning algorithms) methods. Actor–critic algorithms are based on two separate modules: an *actor* which approximates agent's policy and generates actions and a *critic* that estimates the state value function V^{π} defined as:

$$V^{\pi}(s) = \sum_{a} (s;a) \sum_{s'} T(s;a;s') [R(s;a;s') + V^{\pi}(s')] \quad (1)$$

or action-state value function Q^{π} defined as:

$$Q^{\pi}(s;a) = \sum_{s'} T(s;a;s') [R(s;a;s') + V^{\pi}(s')] \quad (2)$$

The actor then updates its policy to maximize return R based on critic's value function estimations.

4 Methods

In this section we describe the formal approach to the intrinsic module based on the prediction error and predictive surprise as shown in Fig. 1. The module provides for a short time a larger amount of intrinsic reward to the agent, especially in the first phases of learning. These bursts of intrinsic reward can be interpreted as predictive surprise, because there is a large difference between an estimated and the actual error of the forward model.

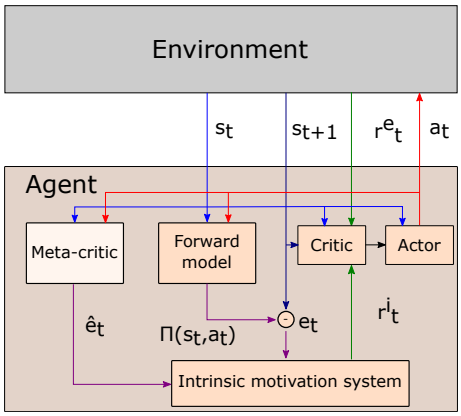


Fig. 1: Proposed intrinsic motivation model with a meta-critic module.

We propose two hypotheses: First, the gating mechanism can take the best of both reward signals and significantly improve the learning process in environments with sparse reward, so the agent should reach optimal policy in shorter time and accumulate more external reward during the training. Second, we expect that performance in environments with dense rewards will be also improved because of more rapid exploration performed mainly in the first period of learning process. Both hypotheses are tested in experiments. Now we describe its individual components.

4.1 Meta-critic

Our motivational module is based on two prediction modules. The first module is the forward model $\Pi(s_t; a_t)$ with parameters $_{\text{fm}}$ which predicts the next state $\hat{s}_{t+1}$ from current state and action

$$\Pi(s_t; a_t; \ _{\text{fm}}) = \hat{s}_{t+1} \tag{3}$$

The prediction error e_t is defined as the normalized squared Euclidean distance between the predicted state $\hat{s}_{t+1}$ and the next observed state s_{t+1}

$$e_t = \frac{1}{n} k s_{t+1} \quad \hat{s}_{t+1} k_2^2 \tag{4}$$

where n is the dimensionality of the state space. The intrinsic reward based on the prediction error is defined as

$$r_t^{\text{ifm}} = e_t \tag{5}$$

Such intrinsic reward decreases as the FM improves its predictions. That ideally occurs the transition from state s_t using action a_t to the next state s_{t+1} which are well-known to the agent's FM, because they were experienced several times, and hence they no longer serve as a source of intrinsic motivation.

The second module estimates predictive surprise motivation which rewards the states that occur but were not expected, or do not occur but were expected. To formalize the expectations, we introduce another predictor MetaΠ and refer to it as a *meta-critic*.[1] It aims to estimate the error e_t of the first predictor Π at time t

$$\text{MetaΠ}(s_t; a_t; \ _{\text{mc}}) = \hat{e}_t \tag{6}$$

where $_{\text{mc}}$ are MC parameters. In this way, we obtain qualitatively new information about the state of the agent's internal model about the environment, which describes how confident the agent is about its predictions. Based on this information we propose a new intrinsic reward function

$$r_t^{\text{imc}} = \begin{cases} e_t = \hat{e}_t + \hat{e}_t = e_t \quad 2; & \text{if } j e_t \quad \hat{e}_t j > \\ 0; & \text{otherwise} \end{cases} \tag{7}$$

If the MC correctly estimates the prediction error, the reward is close to 0 due to constant 2 which is subtracted from the term. To prevent cases where the error estimation and prediction error are very small, but still generate some reward, we introduced a *sensitivity* threshold which has to be exceeded. The reward function defined in this way can stimulate an agent if the prediction error is low and its estimate is high, or vice versa, when the prediction error is high and its estimate is low. The training of the proposed intrinsic module is straightforward and can be approached as an optimization problem, formulated as

$$\min_{\theta_{\text{fm}}, \theta_{\text{mc}}} \quad \frac{1}{n} k s_{t+1} \quad \hat{s}_{t+1} k^2 + k e_t \quad \hat{e}_t k^2 \tag{8}$$

[1] There is no connection to a critic estimating the value functions.

4.2 Intrinsic reward gating

The proposed motivation model has two prediction modules generating two different IM signals. We decided to introduce the gating of reward signals such that in each step of an episode, only one of the two signals is passed through. This is aimed to model situations when the rarely occurring, unexpected event overrides the prediction error reward whose magnitude is much smaller. The final intrinsic reward added to an external reward is defined as

$$r_t^{\mathsf{i}} = \max(\ _{\mathsf{fm}} \tanh(r_t^{\mathsf{ifm}});\ _{\mathsf{mc}} \tanh(r_t^{\mathsf{imc}})) \tag{9}$$

where the reward signals from both modules are scaled to the interval (1; 1) and then independently scaled by a respective factor . This procedure was informed by an observation that predictive surprise motivation often outperforms common prediction error motivation and leads to an effect of sudden surprise for the agent. Without surprise the agent is driven by prediction error motivation.

The final instantaneous reward r_t provided to the critic is defined as

$$r_t = r_t^{\mathsf{e}} + r_t^{\mathsf{i}} \tag{10}$$

where r_t^{e} is the instantaneous external reward and r_t^{i} was defined in eq. 9. The above mentioned types of reward were used in four different agents listed in Table 1.

Table 1: Agents with their respective motivation signals.

Agent type	Motivation
Baseline	none
Forward model (FM)	r^{ifm} (eq. 5)
Meta-critic (MC)	r^{imc} (eq. 7)
Meta-critic gated (gMC)	r^{i} (eq. 9)

5 Experiments

To appreciate the behavior of the proposed models, we tested them in four environments of different complexity, namely *MountainCar* available in OpenAI Gym [4], *AerisNavigate* available in gym-aeris package, then *HalfCheetah* and *Ant* from PyBullet Gym [8]. All environments have continuous state and action spaces. MountainCar present a challenge for exploration, because the agent receives a negative reward according to the magnitude of its action vector, and if it does not find a positive reward fast enough, the policy will converge into the agent's inactivity in the extreme case. AerisNavigate environment is the most difficult due to a very sparse reward obtained only at the end of an episode. The goal of MountainCar, and AerisNavigate agents is to reach a specific location in the state space: the top of the hill and the target area that changes in each episode, respectively. The next two environments (HalfCheetah, Ant) provide a dense reward signal, as a mixture of positive and negative rewards per step. Here the task of the agents is to reach the maximum distance from the starting location until the step limit is over.

We divided our experiments in two parts. The first consists of testing the agent with the MC module in described environments to compare the results

with the baseline models. In the second part, we focus on a statistical analysis of all models using the specific metrics, measuring model reliability, intrinsic reward density and distribution.

5.1 Model training setup

All our agents are trained using DDPG algorithm [16] that has been shown to work well in many tasks. The agent's deterministic policy is approximated by an *actor* and Q-value function is approximated by a *critic*. The actor and critic are represented by three-layer neural networks and for parameter optimization of both modules we used Adam algorithm [13]. The learning rates of actor and critic in all environments were $_{\mathsf{act}} = 0.0001$ and $_{\mathsf{crit}} = 0.0002$, respectively. Exploration was performed by adding noise to the actor's output, generated by random variable with Gaussian distribution and monotonically decreasing standard deviation. All environments had a discount factor set to $= 0.99$ and in all our experiments, $_{\mathsf{fm}} = _{\mathsf{mc}} = 1$, except the experiments in AerisNavigate environment, where $_{\mathsf{fm}} = _{\mathsf{mc}} = 0.01$. More hyper-parameters and further details of the learning process can be found in our source codes. To model a less complex environment, with low state space dimension (MountainCar), we used three-layer neural networks (for both FM and MC) and the models were trained by Adam algorithm, in online manner adapting to actual samples experienced by the agent. We chose the learning rate values $_{\mathsf{fm}} = 0.0001;\ _{\mathsf{mc}} = 0.0002$, respectively, slightly increasing the learning speed of MC to improve the speed of estimation of FM error which represents moving target in this case. Based on preliminary tests we increased the depth of neural networks for more complex environments (HalfCheetah and Ant) to increase their capacity and we decided to use five-layer neural networks. We also employed an experience replay buffer, often used in off-policy learning algorithms to decorrelate the samples, e.g. [20], generating batches of size 32 used for learning of FM as well as MC (each having its own sample batch). Slightly different modules were needed for AerisNavigate environment, where the input was represented as multi-channel tensor of Lidar signals. To implement the FM we used four 1D-conv operators and then one transposed 1D-conv followed by one 1D-conv operator to create prediction about the next states. The MC module has the same structure with an additional linear layer on the top, estimating the FM error.

Table 2: Average cumulative reward per step for all models and tasks.

Model / Env	MountainCar	AerisNavigate	HalfCheetah	Ant
Baseline	53.4 ± 58.4	0.35 ± 0.90	1021.8 ± 414.8	990.4 ± 450.8
Forward model	54.8 ± 59.3	0.44 ± 0.86	1060.1 ± 447.6	**1287.9** ± 562.0
Meta-critic	60.2 ± 52.7	0.30 ± 0.89	1010.3 ± 401.6	1178.6 ± 540.1
Meta-critic gated	**66.1** ± 55.6	**0.48** ± 0.86	**1073.7** ± 421.1	1246.6 ± 563.8

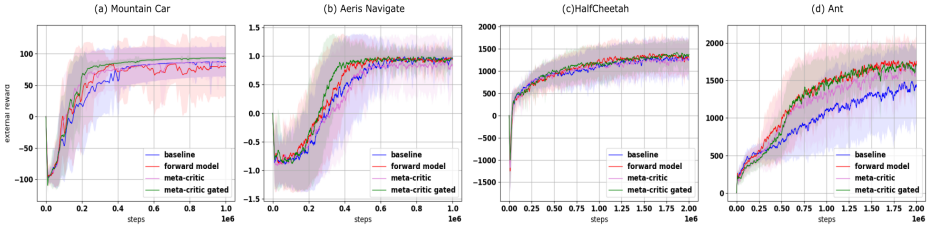
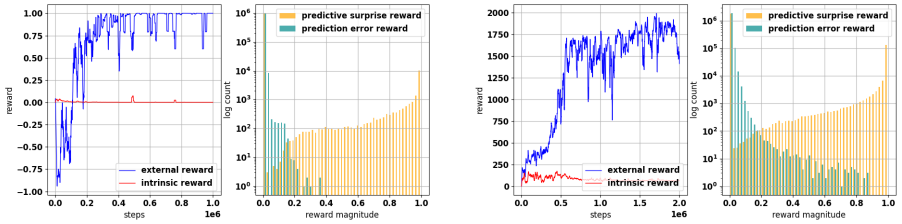


Fig. 2: Simulation results of the four agent types trained in four environments. The gMC agent was the most successful in MountainCar and AerisNavigate environments, and also reached interesting performance in Ant environment. The learned policies of IM-based agents in HalfCheetah environment do not differ much from the baseline.



(a) AerisNavigate (b) Ant

Fig. 3: Detailed analysis of the gated meta-critic for two chosen environments showing measured quantities within a single run. The first chart of 3a shows cumulative rewards and the second chart reveals a distribution of magnitude of intrinsic reward within the entire training. The same holds for 3b. All charts are smoothed by a moving average with window size of 10,000 steps (for interpretation, see the text).

Table 3: Relative proportion (prediction error / predictive surprise) of intrinsic reward signals (averaged over runs) for the gated meta-critic agents across four quarters of the training.

Environment	**Q1**	**Q2**	**Q3**	**Q4**
MountainCar	99.64 / 0.36 %	100.00 / 0.00%	100.00 / 0.00%	100.00 / 0.00%
AerisNavigate	97.96 / 2.04%	98.71 / 1.29%	98.64 / 1.36%	98.50 / 1.50%
HalfCheetah	91.15 / 8.85%	93.73 / 6.27%	93.90 / 6.10%	94.38 / 5.62%
Ant	90.28 / 9.72%	90.78 / 9.22%	92.31 / 7.69%	93.03 / 6.97%

5.2 Model comparison

For all the environments we performed 15 training runs of each variant: the baseline had no motivation, the FM used the prediction error motivation, MC had only predictive surprise motivation and finally the gMC combined both predictive motivations. To evaluate performance of agents we ran basic analysis in which we calculated mean and standard deviation of accumulated external reward and the results can be found in Tab. 2 and Fig. 2. Each curve represents an average cumulative external reward for each step smoothed by running average with window size of 10^5 steps (10 episodes). According to these metrics, gMC agent reached the highest values in four environments (MountainCar, AerisNavigate and HalfCheetah). In two cases (HalfCheetah, Ant) the results were very similar to the other agents, hence not supporting our hypothesis. In Fig. 3 we present single runs of two chosen environments. We can see the evolution of external and intrinsic rewards (left graph) and a distribution of the prediction error and predictive surprise rewards (right graph).

To measure how often is predictive surprise the source of intrinsic reward we divided each training run into 4 quarters (e.g. for run with 1M steps Q1: 0–0.25M, Q2: 0.25–0.5M, etc.). We evaluated average density of predictive surprise occurrence for each quarter. The results are provided in Tab. 3. For completeness and comparison, we also added data for prediction error based reward. In most cases, we can see a decreasing tendency of predictive surprise average density as the learning proceeds to its final phase (Q4).

5.3 Assessment of model reliability during learning

To obtain a quantitative comparison of the RL models, we evaluated selected measures of reliability, following [7]. They proposed three axes of variability, of which the first two capture reliability "during training". Across Time measures the algorithm stability within each run, whereas Across Runs measures consistently reproducible performances across multiple training runs.

For both axes of variability, two kinds of measures are evaluated: dispersion and risk. Dispersion, as the width of the distribution, is taken as the Interquartile range (IQR) (i.e. the difference between the 75th and 25th percentiles), which is suitable for nonnormal distributions. Risk is defined as the heaviness and extent of the lower tail of the distribution. To measure risk, Conditional Value at Risk (CVaR) is used, measuring the expected loss in the worst-case scenarios. For motivation for these measures and more detailed explanation, the reader is referred to [7].

The RL algorithms are evaluated in Fig. 4, separately for sparse and dense rewards, in terms of dispersion and risk across runs which were found most informative. Dispersion profiles are consistent with variability of learning curves in Fig. 2 and reveal the fact the IM-based agents, and in particular gMC agent, outperform the baseline in most cases. Risk profiles provide a new information, since they focus on worst-case behaviors. Here, the evidence shows that gMC excels in two cases (MountainCar and Ant) and is never inferior to other agents.

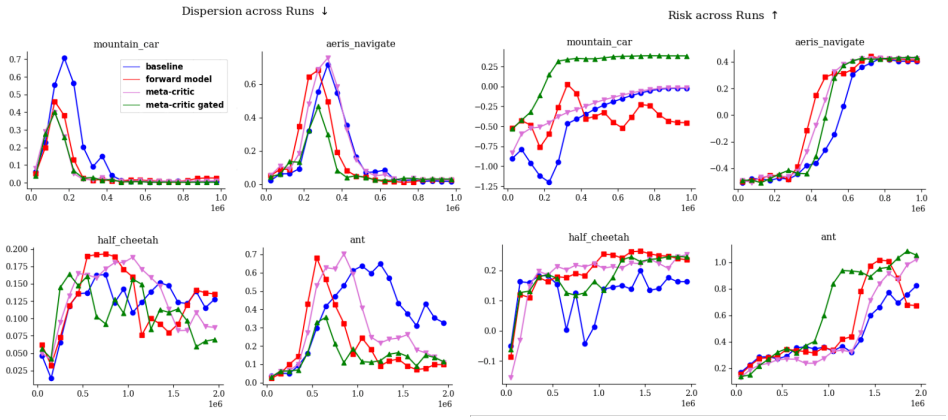


Fig. 4: Selected reliability measures (across runs) assessed for each environment and four learning agent types. Better reliability is indicated by less positive values in case of dispersion, and more positive values in case of risk.

6 Discussion

The process of learning with motivation based on gating the predictive surprise and the prediction error introduces quite complex interactions among all modules (actor, critic, forward model and meta-critic). Presented analyses suggest that it is necessary to employ a suitable FM architecture with a sufficient capacity and an appropriate training technique to take advantage of both signals. In case the FM cannot adapt quickly and hence exhibits unstable behaviour, it introduces much more noise into the model. It is also more difficult for the MC to estimate an error with higher variance, which leads to generating more surprise.

We consider the gMC agent successful in two sparse environments (MountainCar and AerisNavigate), where it outperformed the other agents. For the dense environment HalfCheetah, the external reward is sufficiently informative, and hence adding another source of reward did not induce significant improvement. A different situation occurred in Ant dense environment where the IM agents converged to more successful policies. There is also an open question of scaling the intrinsic reward. We set the scaling parameter so that the accumulated intrinsic reward had a magnitude similar to the accumulated external reward but it would be beneficial to try further experiments with different scales. One of the shortcomings of our approach is the forgetting in FM and MC despite their being trained from the replay buffer. This is noticable particularly in case of AerisNavigate (in Fig. 3). In the late phase of training, when the policy was quite stable and the agent was experiencing a smaller set of different trajectories, its actor module became overtrained and the policy collapsed for a short time. This caused large errors in FM predictions despite the fact, that this worse policy had been experienced earlier (or similar, resulting in a similar accumulated external reward). A sudden change in the prediction error induced surprise

which increased the amount of intrinsic reward motivating the agent to further explore this worsened policy, which obviously had an undesirable effect. We consider the presented method and results of our experiments as a viable proof of concept of a broader research focusing on combining different motivation signals. We proposed an intrinsic motivation model based on the prediction error and the predictive surprise, by introducing another predictor – meta-critic – that estimates the error of the forward model. Predictive surprise represents a qualitatively new information in the form of an intrinsic signal. We performed tests of models with motivation based on predictive surprise and models combining prediction error motivation and surprise by simple gating.

With the gating approach we obtained interesting results, which demonstrate in three tasks the benefit of adding the IM module and also provide insight that combining two motivation signals is a viable approach with not yet fully explored potential. Further improvements of the model will be sought, based on fine-tuning its parameters and modification of its architecture.

We were able to construct intrinsic signals based on the outputs of our predictive modules which can refer to different types of behavior. We plan to identify some basic behaviours in psychology and create corresponding intrinsic signals [9]. We believe that combination of these basic rewards could lead to more complex behaviours narrowing the gap between machines and humans.

References

1. Baldassarre, G.: Intrinsic motivations and open-ended learning (2019), arXiv:1912.13263v1 [cs.AI]
2. Baldassarre, G., Stafford, T., Mirolli, M., Redgrave, P., Ryan, R.M., Barto, A.: Intrinsic motivations and open-ended development in animals, humans, and robots: An overview. Frontiers in Psychology (2014). https://doi.org/10.3389/fpsyg.2014.00985
3. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research **47**, 253–279 (2013)
4. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: OpenAI Gym (2016), arXiv:1606.01540
5. Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., Efros, A.A.: Large-scale study of curiosity-driven learning (2018), arXiv:1808.04355
6. Burda, Y., Edwards, H., Storkey, A., Klimov, O.: Exploration by random network distillation (2018), arXiv:1810.12894
7. Chan, S.C., Fishman, S., Canny, J., Korattikara, A., Guadarrama, S.: Measuring the reliability of reinforcement learning algorithms. In: International Conference on Machine Learning (2020)
8. Coumans, E., Bai, Y.: PyBullet, a Python module for physics simulation for games, robotics and machine learning. `http://pybullet.org` (2016–2019)
9. Csikszentmihalyi, M.: Flow: The Psychology of Optimal Experience. Harper Perennial, New York, NY (1991)
10. Dayan, P., Sejnowski, T.: Exploration bonuses and dual control. Machine Learning **25**, 5–22 (1996)

11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition (2016)
12. Itti, L., Baldi, P.: Bayesian surprise attracts human attention. Vision Research **49**(10), 1295–1306 (2009)
13. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2013), arXiv:1312.6114
15. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Neural Information Processing Systems **25** (2012)
16. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning (2015), arXiv:1509.02971
17. Machado, M.C., Bellemare, M.G., Bowling, M.: Count-based exploration with the successor representation (2018), arXiv:1807.11622
18. Martin, J., Sasikumar, S.N., Everitt, T., Hutter, M.: Count-based exploration in feature space for reinforcement learning (2017), arXiv:1706.08090
19. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**, 529–533 (2015)
20. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing Atari with deep reinforcement learning (2013), arXiv:1312.5602
21. Ostrovski, G., Bellemare, M.G., van den Oord, A., Munos, R.: Count-based exploration with neural density models. In: International Conference on Machine Learning. pp. 2721–2730 (2017)
22. Oudeyer, P.Y., Kaplan, F.: What is intrinsic motivation? a typology of computational approaches. Frontiers in Neurorobotics **1**, 6 (2009)
23. Pathak, D., Agrawal, P., Efros, A.A., Darrell, T.: Curiosity-driven exploration by self-supervised prediction (2017), arXiv:1705.05363
24. Rayyes, R., Donat, H., Steil, J.: Efficient online interest-driven exploration for developmental robots. IEEE Transactions on Cognitive and Developmental Systems (2020)
25. Ryan, R., Deci, E.: Intrinsic and extrinsic motivations: Classic definitions and new directions. Contemporary Educational Psychology **25**(1), 54–67 (2000)
26. Santucci, V.G., Baldassarre, G., Mirolli, M.: Which is the best intrinsic motivation signal for learning multiple skills? Frontiers in Neurorobotics **7**, 22 (2013)
27. Schmidhuber, J.: Curious model-building control systems. In: Proceedings of the International Joint Conference on Neural Networks. pp. 1458–1463 (1991)
28. Stadie, B.C., Levine, S., Abbeel, P.: Incentivizing exploration in reinforcement learning with deep predictive models (2015), arXiv:1507.00814
29. Stadie, B.C., Levine, S., Abbeel, P.: Incentivizing exploration in reinforcement learning with deep predictive models. In: International Conference on Learning Representations (2016)
30. Sutton, R.: Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In: Machine Learning: Proceedings of the 7th International Conference. pp. 216–224 (1990)
31. Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, O.X., Duan, Y., Schulman, J., DeTurck, F., Abbeel, P.: #Exploration: A study of count-based exploration for deep reinforcement learning. In: Advances in neural information processing systems. pp. 2753–2762 (2017)