Is Curiosity All You Need? On the Utility of Emergent Behaviours from Curious Exploration

Oliver Groth[1,2], Markus Wulfmeier[1], Giulia Vezzani[1], Vibhavari Dasagi[1,3], Tim Hertweck[1], Roland Hafner[1], Nicolas Heess[1] and Martin Riedmiller[1]

[1]DeepMind, [2]University of Oxford, [3]Queensland University of Technology

Curiosity-based reward schemes can present powerful exploration mechanisms which facilitate the discovery of solutions for complex, sparse or long-horizon tasks. However, as the agent learns to reach previously unexplored spaces and the objective adapts to reward new areas, many behaviours emerge only to disappear due to being overwritten by the constantly shifting objective. We argue that merely using curiosity for fast environment exploration or as a bonus reward for a specific task does not harness the full potential of this technique and misses useful skills. Instead, we propose to shift the focus towards retaining the behaviours which emerge *during* curiosity-based learning. We posit that these self-discovered behaviours serve as valuable skills in an agent's repertoire to solve related tasks. Our experiments demonstrate the continuous shift in behaviour throughout training and the benefits of a simple policy snapshot method to reuse discovered behaviour for transfer tasks.

1. Introduction

Intrinsic motivation ([Baranes and Oudeyer, 2009](#); [Oudeyer and Kaplan, 2009](#); [Oudeyer et al., 2007](#); [Schmidhuber, 1991, 2010](#)) can be a powerful concept to endow an agent with an automated mechanism to continuously explore its environment in the absence of task information. One common way to implement intrinsic motivation is to train a predictive model alongside the agent's policy and use the model's prediction error as a reward signal for the agent encouraging the exploration of previously unfamiliar transitions in the environment - a method also known as *curiosity learning* ([Pathak et al., 2017](#)). Curiosity-esque reward schemes have been used in different ways to facilitate exploration in sparse tasks ([Burda et al., 2018b](#); [Houthooft et al., 2016](#)) or pre-train policy networks before fine-tuning them on difficult downstream tasks ([Sekar et al., 2020](#)). In environments where the main task objective is highly correlated with thorough exploration, curiosity-based approaches have also been shown to solve the main task without any additional reward signal ([Burda et al., 2018a](#)).

However, in environments with multiple possible tasks – e.g. in manipulation scenarios where objects could be interacted with or re-arranged in different ways – not only the final behaviour of a curious exploration run might be of interest, but intermediate behaviours can correlate with solutions to different tasks. Naturally, the constantly changing curiosity objective leads to the emergence of diverse behaviours during training – much akin to the learning process of infants which develop useful skills by playing ([Haber et al., 2018](#)). Yet, only a fraction of this diversity is ultimately retained in a downstream, task-specific policy – which might also be biased towards exploration as a side-effect – or it is even completely overwritten after fine-tuning. This problem of *catastrophic forgetting* ([McCloskey and Cohen, 1989](#)) is well-known for any neural network which operates under a shifting data distribution. However, the intermediate behaviours which emerge and disappear *during* learning based on curiosity can be of relevance for different tasks of interest. If we were able to extract and leverage emergent behaviour, we could turn the process of exploration from a service for task-driven reinforcement learning into a rich continual learning setup in its own right ([Hadsell et al., 2020](#)).

Despite technical challenges, the discovery of self-induced curricula of skills holds a tantalising prospect for an agent's ability to solve broad sets
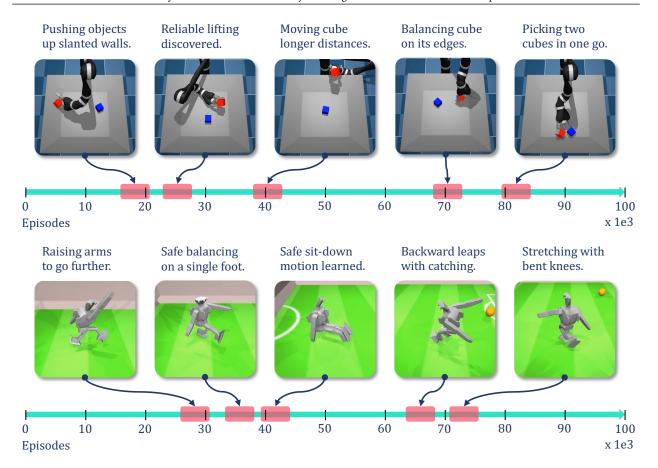
Figure 1 | Two example timelines depicting the emergence of behaviour while pursuing a curiosity objective on a 9-DoF JACO arm (top) and on a 20-DoF OP3 humanoid robot (bottom). Each timeline represents the evolution of behaviour from a single random seed on a single simulated actor. At each point in time, the agent exhibits a single behaviour which slowly evolves over time as the curiosity objective changes. A detailed description of the emergent behaviour in this experiment is provided in section 4.1 and corresponding quantitative results are shown in figs. 3 and 4. The corresponding videos can be found at: https://deepmind.com/research/publications/2021/Is-Curiosity-All-You-Need-On-the-Utility-of-Emergent-Behaviours-from-Curious-Exploration.

of long-horizon tasks when it is able to draw upon potentially useful skills. For instance, a robotic arm which has already discovered how to reach, grasp and lift objects in its workspace has a much easier time exploring and learning a policy to stack objects later on, if it recombined its previously acquired skills. Recent works (Hertweck et al., 2020; Riedmiller et al., 2018; Wulfmeier et al., 2020b) have studied the influence of specific tasks across a spectrum of manual engineering effort like in the stacking example on the learning success of complex manipulation policies. A curiosity-based approach could further reduce the effort of designing a curriculum of tasks and reward functions with a set of self-discovered skills.

In this paper, we study behaviour which emerges based on a curiosity objective in two continuous control settings: manipulation and locomotion. In contrast to prior work in this domain, we implement curiosity-based exploration in an off-policy learning setting which improves upon on-policy implementations in terms of data-efficiency and presumably increases the diversity of emerging behaviours. Furthermore, we look at the utilisation of the self-discovered behaviour for learning new downstream tasks. In particular, we find that even a naïve baseline which treats snapshots of a curiosity policy as fixed exploration skills in a hierarchical learning setup can perform commensurately with a hand-designed curriculum learning setup. Our findings suggest that the identification and exploitation of self-discovered behaviours can be a

fruitful avenue for future work in the learning of complex robotics tasks.

In summary, we make the following two contributions: First, we introduce *SelMo*, an off-policy realisation of a self-motivated, curiosity-based method for exploration which is applied to two robotic manipulation and locomotion domains in simulation. We show that even in those complex, 3D environments, meaningful and diverse behaviour emerges solely based on the optimisation of the curiosity objective. Second, we propose to extend the focus in the application of curiosity learning towards the identification and retention of emerging intermediate behaviours and support this conjecture with a baseline experiment which uses self-discovered behaviours as auxiliary skills in a hierarchical reinforcement learning setup.

2. Related Work

The utilisation of forward modelling error as reward signal has been implemented in deterministic and probabilistic settings (Achiam and Sastry, 2017; Shelhamer et al., 2016) and is commonly known as *curiosity learning*. The error reward signal encourages the exploration of unfamiliar parts of the state-action space which are not yet well-predictable. Pathak et al. (2017) derive the curiosity reward from an inverse dynamics model which is simultaneously less prone to be confused by unpredictable elements of the environment (cf. *white-noise problem* (Schmidhuber, 2010)). Burda et al. (2018b) use a randomly initialised projection from observation into latent space and the predictor driving the learning process is tasked with learning this projection. Huang et al. (2019) use the error of a predictive model of penalties in dexterous manipulation and find that the additional intrinsically motivated exploration helps in the development of gentle grasping policies. Curiosity learning can also be seen through an information theoretic lens using an information gain objective like in Still and Precup (2012) or Houthooft et al. (2016). Lastly, the *disagreement* between an ensemble of forward models (Pathak et al., 2019) can also be treated as a proxy for model uncertainty and exploited as a reward signal. In addition to serving as a reward generator, the predictors can also be used for targeted exploration. Lowrey et al. (2018) and Sekar et al. (2020) employ world models to deliberately explore in regions where the expected prediction error is high *a priori* - as opposed to realising that something surprising has been observed *a posteriori* like in standard curiosity approaches.

Besides curiosity learning, the literature features an extensive body of work on structured exploration methods in reinforcement learning. Classical methods like count-based exploration schemes have been revisited in continuous settings, e.g. in Bellemare et al. (2016) changes in state density estimations are used as exploration bonus rewards. Another line of work revolves around the central notion of *empowerment* (Klyubin et al., 2005) which aims to find new behaviours which are increasingly controllable by the agent (Gregor et al., 2016; Mohamed and Rezende, 2015). The *estimation of model learning progress* has also been discussed as a reward signal for exploration but this approach is much harder to implement as it relies on a measure of model improvement (Lopes et al., 2012). Closely related to model-improvement-based exploration is the idea of *PowerPlay* (Schmidhuber, 2013) which describes an inductive task proposal scheme equipping the agent with a mechanism which infers the current frontier of soluble tasks and inductively creates a novel task which can be solved by employing the agent's current knowledge. However, this exploration scheme has mostly remained conceptual so far with experiments limited to simple pattern recognition tasks (Srivastava et al., 2013). Related to PowerPlay's idea of task proposition but supposedly more tractable is the concept of *self-play* (Sukhbaatar et al., 2017). In this paradigm, two agents play a competitive game where one player is rewarded for inventing a behaviour which the other agent cannot imitate. This approach has recently led to the emergence of highly complex object manipulation behaviour on a simulated robotic arm (OpenAI et al., 2021).

The importance of diverse interactions with the environment – similar to the playful behaviour exhibited by children – has been stressed in recent works in reinforcement learning, e.g. in Haber et al. (2018) or Lynch et al. (2020). Such intrinsically motivated exploration has been shown to lead to the discovery of diverse behaviour and re-
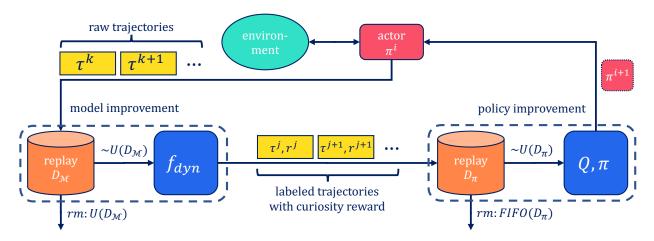
Figure 2 | An overview over the SelMo system architecture. The agent collects trajectories $\tau^k, \tau^{k+1}, \ldots$ in the environment using its current policy π^i and stores it in a model replay buffer $D_{\mathcal{M}}$. When $D_{\mathcal{M}}$ is full, trajectories are replaced with a uniform removal strategy. The dynamics model f_{dyn} samples uniformly from this buffer and updates its parameters for forward prediction using *stochastic gradient descent* (SGD). The sampled trajectories $\tau^j, \tau^{j+1}, \ldots$ are then assigned a curiosity reward $r^j, r^{j+1}, \ldots$ scaled by their respective prediction error under the current $f_{dyn}^{(j)}$. The labeled trajectories are passed on to the policy replay buffer D_{π} which runs a *FIFO* removal strategy. *Maximum a posteriori policy optimisation* (MPO) is used to fit Q-function and policy π based on uniformly drawn samples from the policy replay. The resulting policy π^{i+1} is then synced back into the actor. Note that both model and policy learning is executed in independent loops.

usable skills as a natural 'by-product' of interacting with the environment (Singh et al., 2005). Several recent works have identified the *diversity of behaviour* as a central objective to optimise for during unsupervised exploration (Eysenbach et al., 2018; Sharma et al., 2019). Sharma et al. (2020) have extended this line of work and also showed that the acquired latent 'skill space' can be leveraged in model-predictive control fashion for goal-conditioned navigation on a real-world quadruped.

Our work builds upon the curiosity learning approach utilising the forward prediction error of a dynamics model as a reward signal. However, in contrast to typical curiosity setups (Burda et al., 2018a) which are optimised *on-policy* we employ an *off-policy* method to train the agent. Our method is also set apart from prior art with regards to the utilisation of self-discovered behaviour. Instead of using *model-predictive control* (Sharma et al., 2020), we leverage emergent behaviour directly by employing policy snapshots as modular skills in a mixture policy (Wulfmeier et al., 2020a, 2021).

3. Method

In this section, we present SelMo – a self-motivated exploration method which optimises a curiosity objective in an off-policy fashion. Our system is designed around two key components: A forward dynamics model $f_{dyn}: S \times A \mapsto S$ which aims to approximate the state transition function of the environment and a policy $\pi(a_t|s_t)$ which aims to take transitions in the environment for which the prediction error of f_{dyn} is high.

Given potentially different requirements for the dynamics model and the exploration policy, the implementation builds on two separate learning processes. We have decided to implement the system in this distributed way to more easily realise the data labeling process and have convenient control over learning rate and data flow parameters. However, other implementations which fuse both learning processes and data buffers are also conceivable. Crucially, our setup deviates from recent on-policy approaches for curiosity-based rewards in two important aspects: First, we optimise in an off-policy fashion based on a diverse set of experi-

enced transitions in the environment. In this way, the policy is encouraged to to cover a broader spectrum of exploration avenues as the learning evolves. Second, we employ an approximate but efficient data labelling strategy which only assigns curiosity rewards to trajectories when they are used to update the dynamics model, but refrain from re-labeling all trajectories in the policy replay after every model update. We provide a general overview over the whole system in fig. 2 and explain each component in detail in the paragraphs below.

Model Replay The fixed-size model replay buffer $D_{\mathcal{M}}$ stores trajectories $\tau^k = \left[(s_0^k, a_0^k, s_1^k), \ldots, (s_{T-1}^k, a_{T-1}^k, s_T^k) \right]$ collected by the actor in the environment. The values of all environment transitions (s_t, a_t, s_{t+1}) are normalised in the range $[-1, 1]$. Batches of training data for the world model are sampled uniformly from this buffer as $\mathcal{B}_{\mathcal{M}} := \{\tau^1, \ldots, \tau^B\} \sim \mathcal{U}(D_{\mathcal{M}})$. In order to preserve a diverse sampling of environment transitions during the whole learning process, each trajectory in $D_{\mathcal{M}}$ can be sampled at most n_{max} times. Additionally, old trajectories in the buffer are replaced by new ones at random when the buffer size limit is exceeded.

World Model We describe the environment in which the agent operates as $\mathcal{E} = (S, A, P)$ with state and action spaces S and A as well as a state transition function $s_{t+1} = P(s_t, a_t)$ over discrete time steps which describes the environment's *transition dynamics*. The world model is a forward-predictive model $f_{dyn} : S \times A \mapsto S$ which approximates the environment's transition dynamics as:

$$\hat{s}_{t+1} = f_{dyn}(s_t, a_t; \theta) \tag{1}$$

In our case, eq. (1) is implemented as a two-layer MLP with parameters θ. Besides estimating the transition dynamics from observed data, the world model plays a crucial role in assigning the reward for each observed transition (s_t, a_t, s_{t+1}). When a transition is evaluated by the world model, the assigned reward is scaled by the model's current prediction error.

$$r^{(j)}(s_t, a_t, s_{t+1}) = \tanh(\eta_r * (f_{dyn}^{(j)}(s_t, a_t) - s_{t+1})^2) \tag{2}$$

The 'state' of the world model is indicated by the number of gradient updates j which have been performed on it so far. We scale the reward via a hyper-parameter η_r and pass it through a tanh to keep it bounded for the downstream policy learning procedure. When a new batch of data $\mathcal{B}$ is sampled from the model replay, the world model performs two operations. First, it labels each $\tau \in \mathcal{B}$ by assigning curiosity rewards r^C according to eq. (2). Second, it performs one gradient update $\theta^{(j+1)} \leftarrow \theta^{(j)} + \eta_{\mathcal{M}} \partial \mathcal{L}_{dyn}^{(j)}(\mathcal{B}) / \partial \theta^{(j)}$ by minimising its prediction loss:

$$\mathcal{L}_{dyn}^{(j)}(\mathcal{B}) = \sum_{\tau \in \mathcal{B}} \sum_{(s_t, a_t, s_{t+1}) \in \tau} (f_{dyn}^{(j)}(s_t, a_t) - s_{t+1})^2 \tag{3}$$

After one world model update, the relabeled batch of trajectories $\tilde{\mathcal{B}}$ is stored in the policy replay buffer D_π.

Policy Replay The fixed-size policy replay D_π stores tuples (τ^j, r^j) representing trajectories which have been labeled with curiosity rewards by the world model. This off-policy setup provides the policy learner with a diverse training set to optimise for useful exploration actions *globally* and not only in the vicinity of the most recent experience. During policy learning, data batches are sampled uniformly from this buffer as $\mathcal{B}_\pi := \{(\tau^1, r^1), \ldots, (\tau^B, r^B)\} \sim \mathcal{U}(D_\pi)$. Similar to the model replay, each tuple can be sampled up to m_{max} times to increase the utilisation of each data point during policy learning. However, the removal strategy of the policy replay is *FIFO* to ensure that trajectories with the most outdated curiosity rewards get replaced first to reflect the change in the world model, albeit with a certain delay.

Policy The *Markov Decision Process* (MDP) which is induced by this setup can be written as: $\mathcal{M}^{(j)} = (S, A, P, r^{(j)})$. Since the world model $f_{dyn}^{(j)}$ changes with every gradient update, the reward function $r^{(j)}$ changes continuously. Consequently, the reward varies with the model training timesteps j and the policy π is required to keep adapting. The policy replay D_π is filled by the world model's training loop. It contains data of the most recent

$\kappa = |D_\pi|/|\mathcal{B}|$ versions of the MDP. Hence, the resulting policy is optimising for a mixture of MDPs $\{(S, A, P, r^{(\iota)} \mid \iota \in [j - \kappa, \ldots, j]\}$. Policy and critic network are implemented as two separate MLPs. The policy is optimised off-policy using MPO (Abdolmaleki et al., 2018) and a separate learning rate η_π.

4. Experiments

This section is split in two parts. First, we report our analysis of the emergence of behaviours when optimising only for a curiosity reward in section 4.1. Second, we present an empirical utilisation of self-discovered behaviour for accelerated learning of new downstream tasks in section 4.2. For details regarding the simulation domains and model hyperparameters, we refer the reader to appendix A and appendix B respectively.

4.1. Emergence of Behaviour

We start our investigation with an analysis of the behaviour which emerges in our two simulated domains as depicted in fig. 1. The JACO domain features a 9 DoF robotic arm and two cubes; the OP3 domain features a 20 DoF humanoid robot. For this experiment, we run the SelMo learning loop (cf. fig. 2) with a single actor for 100K episodes on each of the environments. During training, the agent solely optimises its curiosity objective which is defined by eq. (2). The visual inspection of the experiments reveal that in both cases, diverse sets of human-interpretable behaviour emerge consistently and are exhibited by the agent for extended periods of time before the ever-changing curiosity reward function shifts the learning towards a new behaviour. Below, we describe the observed behaviours in each domain in greater detail.

Emergent Manipulation Behaviour on JACO

A qualitative example timeline of emerging behaviours on the JACO arm is depicted in fig. 1 (top) and supplemented by a plot evaluating reaching and lifting behaviour during the run in fig. 3. We find that the agent is very quickly driven towards both cubes with equal attention and starts interacting with them by pushing them around. Soon

thereafter, it discovers that pushing them up the slanted walls of the bin facilitates picking them up before it stably latches onto a mode in which it prefers manipulating the red cube over the blue one after approximately 15K episodes. This also coincides with a first period of sustained lifting of the red cube. We hypothesise that the discovery of lifting said cube reinforces the interaction with it as it opens up a new dimension along which model prediction error can be rewarded: the height of the object. After about 25K episodes, it has learned to pick up an object reliably even without the help of the slopes.

After approximately 40K episodes, the curiosity objective pushes the agent to deliberately take objects outside of the workspace and perform pick-and-place operations which move a cube over a long distance but at a lower height. Interestingly, the policy does not degenerate into extreme behaviours like spinning motions which have been observed in related work (Sekar et al., 2020) but stays focused on the objects and keeps exploring their physical properties. For instance, at around 70K episodes, the policy investigates the stability of cube poses in a targeted way by balancing them on their edges and corners. Finally, after about 80K episodes, it starts exploring the possibilities of moving both cubes simultaneously.

Emergent Locomotion Behaviour on OP3

Similar to the JACO arm, we present a timeline of emerging behaviour on the OP3 in fig. 1 (bottom) and a corresponding evaluation of locomotion behaviour in fig. 4. Unsurprisingly, the agent spends roughly the first 2K episodes – indicated by the steep rise in walking rewards – just on learning a sense of balance because an episode is terminated early when the torso constraint (cf. appendix A) is violated and the agent is about to fall. This also corresponds to maximising the experienced episode length because this increases the chances of further increasing the accumulated reward. This finding is in line with earlier work (Pathak et al., 2017) which has shown that the avoidance of a 'death' event is a natural by-product of curiosity-driven learning with a positive reward and favourably shapes the emerging policy.
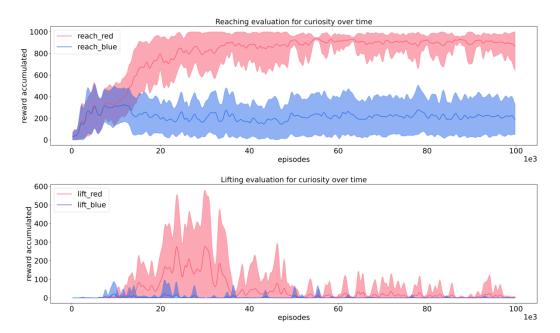
Figure 3 | Manipulation task evaluation in the JACO environment over the lifetime of one experiment while the agent is only trained on the curiosity objective (cf. eq. (2)). A snapshot of the curiosity policy is saved every 100 episodes and evaluated on reaching and lifting the red and blue cubes respectively. Mean and standard deviation are plotted over 20 evaluation runs per policy snapshot and the plot is smoothed with an exponential filter of $\sigma = 1.5$.

Once the agent has learned to stay upright, it slowly starts to develop basic locomotion in the form of stumbling forwards and backwards with only small foot lifting heights which is also reflected in minor oscillations during the evaluation of the walking rewards in fig. 4. Interestingly, after around 30K episodes, the agent has discovered to swing its arms to take bigger steps. This is most impressively first demonstrated after nearly 40K episodes when the agent balances on one foot while stretching out the other leg using its arms for counterbalancing moves. Using the arms also opens new avenues for exploration. Approximately 40K episodes into training, the agent has learned to catch itself when falling backwards. This leads to the discovery of a sit-down behaviour which does not violate the environment's torso constraints. After the agent has explored various 'ground exercises' it switches back to walking gaits at around 55K episodes. Then, the whole body movement has become considerably more nimble and its movement repertoire now features quick turns, stumbling reflexes and even safe backward leaps. After

about 70K episodes the agent starts revisiting earlier behaviour, e.g. the balancing skill, but keeps adding variations to it like knee-bending or stretching.

4.2. Utilisation of Emergent Behaviour

As we have discussed in the previous section, the constantly evolving curiosity policy develops behaviours which correspond to the solution of concrete tasks (cf. fig. 3, fig. 4). In order to retain those diverse behaviours to accelerate the learning of new tasks, we have devised the following experiment: Assuming that the curiosity policies exhibit undirected yet versatile exploratory behaviour, we investigate how well they could serve as auxiliary skills in a modular learning setup of a downstream task.

We employ *Regularized Hierarchical Policy Optimization* (RHPO) (Wulfmeier et al., 2020a) as this framework allows us to compose multiple policies in a hierarchical manner. In each environment, we define a downstream target task which we are inter-
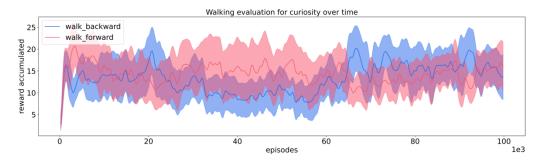
Figure 4 | Task evaluation in the OP3 environment over the lifetime of one experiment while the agent is only trained on the curiosity objective (cf. eq. (2)). A snapshot of the curiosity policy is saved every 100 episodes and evaluated on locomotion (`walk_{forward,backward}`) tasks. Mean and standard deviation are plotted over 20 evaluation runs per policy snapshot and the plot is smoothed with an exponential filter of $\sigma = 1.5$.

ested in learning and provide five policy snapshots from a SelMo experiment as auxiliary exploration skills. During the SelMo experiment, we optimise solely for the curiosity objective and save a snapshot of the curiosity policy every 100 episodes. The RHPO experiment subsequently samples SelMo policy snapshots and utilises the behaviour exhibited by them to assist the exploration for the desired downstream task.

For this experiment, we refer again to the JACO environment where we define the target task to be `lift_red`. While the policy for the target task is randomly initialised, the auxiliary policies are randomly sampled from the SelMo snapshots and kept fixed during the entire RHPO training. We differentiate between three different phases from which the SelMo snapshots are chosen: `early` comprises of snapshots which have been trained on up to 10K episodes, `mid` snapshots are from the interval between 10K and 20K episodes and `late` snapshots are sampled between 20K and 30K training episodes. In fig. 5 we compare the learning progress of the downstream task with the sampled SelMo auxiliary skills against a baseline featuring hand-designed task curricula in an SAC-X framework (Riedmiller et al., 2018). In the case of the JACO environment, the agent is given a curriculum of reward functions which help to `reach` and `move` the `red` cube.

In the case of `lift_red` on JACO, we find that SelMo auxiliaries from the `mid` and `late` exploration periods give the learning of the lifting policy

a significant boost which is commensurate with a tuned SAC-X baseline featuring multiple auxiliary rewards which have been hand-designed to facilitate the learning of `lift_red`. This result is in line with the performance observed in fig. 3 where the curiosity-based policy has developed a sustained lifting behaviour between 20K and 35K episodes. Consequently, snapshot auxiliaries sampled from that range are particularly useful when a targeted lifting policy is to be learned.

This experiment shows that even a simple behaviour retention strategy like policy snapshotting can already provide clear benefits for downstream learning of new tasks. The self-discovered behaviours from a curiosity training phase afford a task learning scaffold which can be commensurate with a specifically designed set of auxiliary reward functions. This is a promising result suggesting that independent curious exploration could be used in lieu of human-engineered task curricula in complex manipulation scenarios.

5. Discussion

Our experiments have shown that complex manipulation and locomotion behaviour such as grasping, lifting, balancing, sitting and leaping emerges completely unsupervised in an off-policy curiosity learning setup on a 9 DoF robot arm and a 20 DoF humanoid. This observation supports our hypothesis that self-discovered behaviour can provide a valuable skill repertoire for the learning of
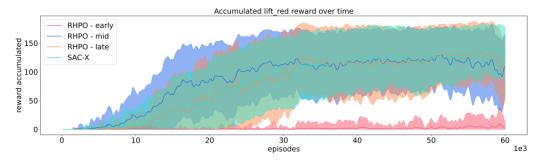
Figure 5 | Learning curves for hierarchical skill learning of `lift_red` using SelMo auxiliary policies in the JACO environment. The SAC-X baseline uses auxiliary reward functions {`reach,move`}`_red`. Each RHPO run uses five randomly sampled SelMo policies from the respective intervals as auxiliary skills (cf. section 4.2). Mean and standard deviation are plotted for five random seeds for each model and the plot is smoothed with an exponential filter of $\sigma = 1.5$.

new downstream tasks. We add further evidence to this hypothesis by utilising randomly selected policy snapshots from a curiosity training as auxiliary skills in a modular learning setup and show that they provide a learning scaffold commensurate with hand-designed auxiliary reward functions for the respective tasks. This suggests that curiosity-based exploration should be treated as an independent aspect of a learning system as opposed to a mere bonus reward or policy pre-training phase. In this study, we provide a baseline for harnessing self-discovered behaviour by randomly sampling policy snapshots and treating them as fixed skills in a modular learning setup. However, more sophisticated techniques for identification, retention and utilisation of behaviour in curious exploration settings are conceivable and will be briefly touched upon in this discussion section.

Identification of Emerging Behaviour As we have shown in section 4.1, complex behaviour can emerge in an unsupervised way when optimising a policy for a curiosity objective. However, it is revealed in fig. 3 and fig. 4 that pre-conceived reward functions are only able to capture a fraction of the emergent behaviour like reaching and lifting of individual objects or basic walking gaits. More involved behaviours are not covered by basic reward functions and implementing reward functions to identify a broad set of behaviour a priori does not scale well with the unsupervised nature of curiosity learning. Diversity-based approaches (Eysenbach

et al., 2018; Sharma et al., 2019, 2020) address this challenge via a latent 'skill space' which modulates the policy network to capture different modes of operation like jumping and walking. The temporal and hierarchical abstraction provided by the latent vector also facilitates planning over the skill space. However, the identification of particularly useful skills worth retaining and comparatively useless skills which could be overwritten is still an open question, especially in never-ending learning settings where a curiosity-driven exploration keeps discovering new behaviours or revisiting old ones.

Retention and Utilisation of Self-Discovered Behaviour Closely related to the question of behaviour identification is the question of its retention and utilisation. In our setup we have treated snapshots of the same curiosity policy as different skills as they represent different behaviour over time. Using a latent skill space as an arbiter for different behaviours in the same policy network (Eysenbach et al., 2018; Sharma et al., 2019) has also been shown to work effectively as this formulation also enables planning over the learned space of skills. However, Lynch et al. (2020) have also demonstrated the utility of raw *play data* in the training of versatile goal-conditioned policies which would position the curiosity-driven exploration as a data collector for a downstream policy distillation instead of a re-usable behaviour in itself. Lastly, Riedmiller et al. (2018) and Hertweck et al. (2020) have shown the benefits of curricula of reward

functions for the learning of complex manipulation tasks which opens another avenue for the utilisation of curiosity-based learning: Instead of using frozen snapshots of the policy as a repertoire of skills one could also treat different versions of the world model as a set of distinct reward functions to encourage the optimisation for diverse behaviour during learning of new downstream tasks.

6. Conclusion

In this paper we have studied the emerging behaviour when optimising an exploration policy for a curiosity objective derived from a forward-predictive world model. To this end, we have presented SelMo, a curiosity-based, off-policy exploration method and applied it in two continuous control domains: a simulated robotic arm and humanoid robot. We have observed that complex behaviour emerges in both settings and provided a baseline for the utilisation of this self-discovered behaviour in a modular downstream learning scenario. Despite the remaining technical challenges, we believe that the automatic identification and retention of useful emerging behaviour from curious exploration is a fruitful avenue of future investigation in unsupervised reinforcement learning.

Author Contributions

Oliver Groth developed and implemented the SelMo model, conducted the curiosity experiments, studied related work on self-motivated reinforcement learning and wrote the paper. **Markus Wulfmeier** advised during the implementation and experimentation phase and helped writing the paper. **Giulia Vezzani** conducted the RHPO experiments and created the experiment evaluation plots. **Vibhavari Dasagi** assisted in compiling related work and provided feedback on the method and discussion section. **Tim Hertweck** provided technical support during the implementation and experimentation phase and created the supplementary videos. **Roland Hafner** conducted the SAC-X baseline experiments and provided support in setting up the JACO and OP3 environments. **Nicolas Heess** provided feedback on the experimental results and during writing and revision of the paper. **Martin Riedmiller** conceived the idea of self-motivated skill learning, provided conceptual feedback and supervised the project.

References

A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

J. Achiam and S. Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

A. Baranes and P.-Y. Oudeyer. R-iac: Robust intrinsically motivated exploration and active learning. *IEEE Transactions on Autonomous Mental Development*, 1(3):155–169, 2009.

M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29:1471–1479, 2016.

Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018a.

Y. Burda, H. Edwards, A. Storkey, and O. Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018b.

B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

K. Gregor, D. J. Rezende, and D. Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

N. Haber, D. Mrowca, S. Wang, F.-F. Li, and D. L. Yamins. Learning to play with intrinsically-motivated, self-aware agents. In *NeurIPS*, pages 8398–8409, 2018.

R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in cognitive sciences*, 2020.

T. Hertweck, M. Riedmiller, M. Bloesch, J. T. Springenberg, N. Siegel, M. Wulfmeier, R. Hafner, and N. Heess. Simple sensor intentions for exploration. *arXiv preprint arXiv:2005.07541*, 2020.

R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. Vime: Variational information maximizing exploration. *arXiv preprint arXiv:1605.09674*, 2016.

S. H. Huang, M. Zambelli, J. Kay, M. F. Martins, Y. Tassa, P. M. Pilarski, and R. Hadsell. Learning gentle object manipulation with curiosity-driven deep reinforcement learning. *arXiv preprint arXiv:1903.08542*, 2019.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A. S. Klyubin, D. Polani, and C. L. Nehaniv. Empowerment: A universal agent-centric measure of control. In *2005 ieee congress on evolutionary computation*, volume 1, pages 128–135. IEEE, 2005.

M. Lopes, T. Lang, M. Toussaint, and P.-Y. Oudeyer. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *Neural Information Processing Systems (NIPS)*, 2012.

K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch. Plan online, learn offline: Efficient learning and exploration via model-based control. *arXiv preprint arXiv:1811.01848*, 2018.

C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. Learning latent plans from play. In *Conference on Robot Learning*, pages 1113–1132. PMLR, 2020.

M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

S. Mohamed and D. J. Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *arXiv preprint arXiv:1509.08731*, 2015.

O. OpenAI, M. Plappert, R. Sampedro, T. Xu, I. Akkaya, V. Kosaraju, P. Welinder, R. D'Sa, A. Petron, H. P. d. O. Pinto, et al. Asymmetric self-play for automatic goal discovery in robotic manipulation. *arXiv preprint arXiv:2101.04882*, 2021.

P.-Y. Oudeyer and F. Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pages 2778–2787. PMLR, 2017.

D. Pathak, D. Gandhi, and A. Gupta. Self-supervised exploration via disagreement. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5062–5071. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/pathak19a.html.

M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Wiele, V. Mnih, N. Heess, and J. T. Springenberg. Learning by playing solving sparse reward tasks from scratch. In *International Conference on Machine Learning*, pages 4344–4353. PMLR, 2018.

J. Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pages 222–227, 1991.

J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3): 230–247, 2010.

J. Schmidhuber. Powerplay: Training an increasingly general problem solver by continually search-

ing for the simplest still unsolvable problem. *Frontiers in psychology*, 4:313, 2013.

R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pages 8583–8592. PMLR, 2020.

A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

A. Sharma, M. Ahn, S. Levine, V. Kumar, K. Hausman, and S. Gu. Emergent real-world robotic skills via unsupervised off-policy reinforcement learning. *arXiv preprint arXiv:2004.12974*, 2020.

E. Shelhamer, P. Mahmoudieh, M. Argus, and T. Darrell. Loss is its own reward: Self-supervision for reinforcement learning. *arXiv preprint arXiv:1612.07307*, 2016.

S. Singh, A. G. Barto, and N. Chentanez. Intrinsically motivated reinforcement learning. Technical report, MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE, 2005.

R. K. Srivastava, B. R. Steunebrink, and J. Schmidhuber. First experiments with powerplay. *Neural Networks*, 41:130–136, 2013.

S. Still and D. Precup. An information-theoretic approach to curiosity-driven reinforcement learning. *Theory in Biosciences*, 131(3):139–148, 2012.

S. Sukhbaatar, Z. Lin, I. Kostrikov, G. Synnaeve, A. Szlam, and R. Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*, 2017.

M. Wulfmeier, A. Abdolmaleki, R. Hafner, J. T. Springenberg, M. Neunert, T. Hertweck, T. Lampe, N. Siegel, N. Heess, and M. Riedmiller. Compositional transfer in hierarchical reinforcement learning. In *Robotics: Science and Systems*. Robotics: Science and Systems Foundation, 2020a.

M. Wulfmeier, A. Byravan, T. Hertweck, I. Higgins, A. Gupta, T. Kulkarni, M. Reynolds, D. Teplyashin, R. Hafner, T. Lampe, et al. Representation matters: Improving perception and exploration for robotics. *arXiv preprint arXiv:2011.01758*, 2020b.

M. Wulfmeier, D. Rao, R. Hafner, T. Lampe, A. Abdolmaleki, T. Hertweck, M. Neunert, D. Tirumala, N. Siegel, N. Heess, et al. Data-efficient hindsight off-policy option learning. In *International Conference on Machine Learning*, pages 11340–11350. PMLR, 2021.

A. Simulation Environments

In this section we provide details about the two robotic simulation domains used in our experiments.
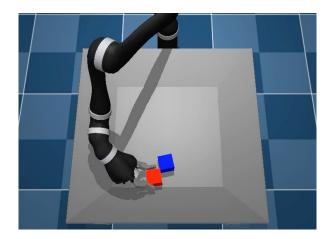

Figure 6 | The JACO manipulation environment. The 6 DoF robot arm with a 3 DoF gripper can interact with multiple same-sized cubes in its workspace.

Table 1 | State and action space semantics of JACO environment.

FEATURE	DIMENSION
arm/joints_pos	6
arm/joints_vel	6
arm/hand/finger_joints_pos	3
arm/hand/finger_joints_vel	3
arm/hand/fingertip_sensors	3
arm/hand/pinch_site_pos	3
proprioception	$\sum = 24$
O<i>/rel_pos_wrt_tcp	3
O<i>/pos	3
O<i>/orientation	4
O<i>/linvel	3
O<i>/angvel	3
O<i>/proptype	5
O<i>/dimensions	3
perception per object <i>	$\sum = 24$
arm	6
hand	3
action space	$\sum = 9$

A.1. JACO Manipulation Environment

This environment is designed to study manipulation tasks like object lifting and stacking with a robotic arm (cf. fig. 6). The state observation space consists of 72 dimensions: 24 features are used to represent the robot's proprioception as well as the state of each object. The action space spans 9 dimensions: 6 concerning the arm and 3 concerning the three-point gripper. A detailed description of the environment is provided in table 1. Interactions with the two objects (O1 = red cube, O2 = blue cube) are evaluated using the sparse reward functions reach_{red,blue} and lift_{red,blue}. Each episode in this environment lasts 20 seconds or 400 control timesteps.

Figure 7 | The OP3 locomotion environment. The 20 DoF humanoid robot can walk around on a plane. The episode terminates early, if the OP3 is about to fall over.

A.2. OP3 Locomotion Environment

This environment is designed to study locomotion with a humanoid robot (cf. fig. 7). The state observation consists of 49 proprioceptive features. The 20-dimensional action space controls the orientations of the robot's head, ankle, elbow, hip, knee and shoulder. Actions passed to the robot are smoothed with an exponential filter to reduce motion jerk. A detailed description of the environment is provided in table 2. The robot always spawns in a standing, upright position. Locomotion is evaluated by the dense reward functions walk_{forward,backward} for forward and back-

Table 2 | State and action space semantics of OP3 environment.

FEATURE	DIMENSION
`walker/joints/pos`	20
`walker/imu/linear_acc`	3
`walker/imu/angular_vel`	3
`walker/imu/gravity`	3
`scaled/action_filter/state`	20
proprioception	$\sum = 49$
`head_{pan,tilt}`	2
`{l,r}_ankle_{pitch,roll}`	4
`{l,r}_elbow`	2
`{l,r}_hip_{pitch,roll,yaw}`	6
`{l,r}_knee`	2
`{l,r}_shoulder_{pitch,roll}`	4
action space	$\sum = 20$

ward walking gaits respectively. Each episode in this environment lasts 10 seconds or 200 control timesteps. If the robot's hip angle deviates more than 15° from an upright orientation, the simulation terminates early effectively preventing the robot from falling over.

B. Model and Training Details

Across all experiments with the SelMo architecture, we consistently use the following hyper-parameters and model architectures. Each SelMo experiment is run with a single actor for $N = 1e5$ episodes.

World Model f_{dyn} The world model is implemented as a *multi-layer perceptron* (MLP) with the following layer sizes and activation functions: [FC(256), elu($\cdot$), FC(256), elu($\cdot$), FC(size_state)] For each environment, `size_state` is the sum of the dimensions for proprioception and perception (cf. appendix A). The world model is optimised using Adam (Kingma and Ba, 2014) with a learning rate of $\eta_{\mathcal{M}} = 3e - 4$.

Policy Both policy π and critic Q are implemented as two independent MLPs with the following layer sizes and activation functions:

- Q: [tanh($\cdot$), FC(512), elu($\cdot$), FC(512),

elu($\cdot$), FC(256), FC(1)]
- π: [FC(256), elu($\cdot$), FC(256), elu($\cdot$), FC(128), FC(size_action)]

The `size_action` is different for each environment (cf. appendix A). The policy is optimised using Adam (Kingma and Ba, 2014) with a learning rate of $\eta_{\pi} = 3e - 4$. The reward scale is set to $\eta_r = 10.0$ across all experiments.

Replays The replays $D_{\mathcal{M}}$ and D_{π} store trajectories with a length of $T = 50$ transitions. The buffer sizes used are $|D_{\mathcal{M}}| = |D_{\pi}| = 5e4$ and each trajectory in the buffers can be sampled at most $n_{max} = m_{max} = 32$ times (cf. section 3). The batch size of samples drawn from the replays is set to $B = 64$.