Exploration with Intrinsic Motivation using Object-Action-Outcome Latent Space

Melisa Idil Sener[1], Yukie Nagai[2], Erhan Oztop[3,4] and Emre Ugur[1]

[1]Bogazici University, Istanbul, Turkey. [2]The University of Tokyo, Tokyo, Japan. [3]Osaka University, Osaka, Japan. [4]Ozyegin University, Istanbul, Turkey.

***Abstract*—One effective approach for equipping artificial agents with sensorimotor skills is to use self-exploration. To do this efficiently is critical, as time and data collection are costly. In this study, we propose an exploration mechanism that blends action, object, and action outcome representations into a latent space, where local regions are formed to host forward model learning. The agent uses intrinsic motivation to select the forward model with the highest learning progress to adopt at a given exploration step. This parallels how infants learn, as high learning progress indicates that the learning problem is neither too easy nor too difficult in the selected region. The proposed approach is validated with a simulated robot in a table-top environment. The simulation scene comprises a robot and various objects, where the robot interacts with one of them each time using a set of parameterized actions and learns the outcomes of these interactions. With the proposed approach, the robot organizes its curriculum of learning as in existing intrinsic motivation approaches and outperforms them in learning speed. Moreover, the learning regime demonstrates features that partially match infant development; in particular, the proposed system learns to predict the outcomes of different skills in a staged manner.**

***Index Terms*—intrinsic motivation, effect prediction, representation learning, developmental robotics, open-ended learning.**

I. Introduction

FROM the moment they are born, babies begin learning about their bodies and the environment autonomously. Even when there is no immediate reward or explicit assistance from their caregiver, it is quite interesting that they conduct this learning process and develop sophisticated skills. Autonomous exploration has been regarded as an essential mechanism for the learning and development of living organisms [1], [2]. Exploratory behaviors, which enable us to adapt to different kinds of situations, learn complex skills, and practice our creativity, are observed not only in humans but also in other animals [3], [4]. Because of the need to feel competent and self-determining, humans engage in novelty-seeking behaviors such as exploration and play [5], which are later described with the generic term of "intrinsically motivated" behavior [6]. Recently, the neural correlates of such behaviors have been found to be linked to dopaminergic systems in the brain (see [7], [8]).

Intrinsically motivated strategies have been used along with various types of robot learning methods and applications such as socially guided learning [9]–[11], affordances [12]–[14], and planning [15]. Given the exploration space of the agent, a particular intrinsic motivation (IM) signal, *learning progress* [16], [17], aims to give priority to exploration regions that are neither too easy nor too difficult to learn, i.e., with the appropriate level of complexity which is inline with infant data [18].

Inspired by infant development, this paper studies how a manipulator robot can learn the outcomes of its actions via autonomous exploration and intrinsic motivation. Predicting the consequences of own actions is an important requirement for intelligent control and decision making in both biological and artificial systems. The importance of predictive learning in human sensorimotor and cognitive development has already been emphasized by [19]. The exploration space of the manipulator robot for predictive learning is composed of the space of objects that it encounters, the action space of the robot, and the outcomes that it observes. During its exploration, the robot is expected to select objects, actions, and outcomes intelligently in order to acquire the target prediction capability most efficiently. It is desirable to avoid pre-defining the set of objects, actions, and outcomes in unsupervised learning settings, where they are typically represented or parameterized by continuous variables. Therefore, the robot has to explore a continuous space with the help of IM to form predictive internal models that can be used for better control and decision-making. How animals and humans efficiently and effectively construct predictive internal models for representing high-dimensional object-action-outcome relations inspires our approach as well as several other computational approaches [19]–[21]. It is generally accepted that humans and animals are endowed with neural mechanisms that build internal models to facilitate effective control of objects and/or body parts in different dynamics [22]. An internal model is a computational structure that mimics (a part of) the sensorimotor system in terms of input-output relations, which may be conceived at different levels of motor control hierarchy (see, e.g., [23], [24]). For motor control, Wolpert and Kawato [21] proposed a computational model that is composed of multiple paired forward-inverse models. The contribution of each pair to the behavioral output is determined by a *responsibility signal* that is computed based on the model's prediction ability. The general benefits for adopting such a modular strategy are suggested as (1) efficient coding of the tasks that might be encountered in a variety of contexts; (2) simultaneous learning of such task contexts without interference; and (3) the possibility of learning a more complex context by reusing the knowledge captured in the learned modules.

In this study, we also adopt a modular approach for forward modeling, and use learning progress measure to gate learning. To form the modules, the exploration space of the robot, i.e., object-action-outcome space, is transformed into a compact latent space and then partitioned into regions, for which individual forward models are trained to become responsible for their region. Directly partitioning the object-action-outcome space is not feasible through standard clustering algorithms as this space is composed of diverse and complex set of variables such as pixel values of the top-down depth image of the objects, various parameters of the manipulation actions, and the position and orientation changes. For effectively partitioning the exploration space, first, a low-dimensional latent representation, that fuses the related triplets (object, action, outcome), is formed. Then, the formed latent space is clustered into regions. During forward model learning, the regions with the highest learning progress, i.e., the regions whose forward models exhibit a maximum decline in prediction error, are prioritized. Through simulation experiments involving a robot arm-hand system that reaches and grasps different types of objects placed in various orientations and sizes with varying arm and hand parameters, we showed that:

- the exploration regions formed in the blended object-action-outcome space correspond to semantically meaningful manipulation primitives,
- the proposed latent space IM approach outperforms competing IM methods (adapted to our setup) that utilize only object [25], action [12], or outcome/goal [26] spaces in terms of learning speed, and
- the exploration order that the proposed approach posits is partially parallel with the staged development of action prediction in infants. To be concrete, the proposed system learns to predict the basic grasp action outcomes before learning the outcome of purposeful push actions [27].

The rest of this paper is structured as follows: the related literature is first reviewed in Section II. Then, the proposed architecture, including its components and the experimental setup, are presented in Sections III and IV. Section V demonstrates the outperforming results of the proposed system. Finally, Section VI gives discussion and conclusions.

II. Related Work

A. Computational Models of Intrinsic Motivation

Regarding the high-dimensional and complex dynamics involved in physical systems, exploration is considered an essential problem in robot learning [28]. Intrinsically motivated strategies are widely used to address the exploration problem. Oudeyer and Kaplan [1] divide the computational approaches of IM into two classes as *Knowledge-Based IM (KB-IM)* and *Competence-Based IM (CB-IM)*. The KB-IM strategy is derived from the deviation of the agent's knowledge of the environment from reality [1]. The agent learns new skills while expanding its knowledge about the environment by exploring the situations outside of its current understanding [29]. The CB-IM strategy focuses on a specific state, i.e., the goal state, that changes adaptively according to the current competencies of the agent [29]. Mirolli and Baldassarre [30] state that both KB-IM and CB-IM can serve knowledge and competence acquisition. In other words, KB and CB-IM have a complementary relationship, where the former aims to detect which skills to train based on their novelties, and the latter to select the expert to achieve a particular goal, as shown in [31]. In terms of robotic application, intrinsic motivation has been studied under reinforcement learning [32], [33] and developmental robotics [16], [34]–[37].

a) Reinforcement Learning (RL): In reinforcement learning, an agent learns an optimal policy to accomplish certain goals typically by considering the extrinsic rewards, i.e., external rewards that stem from the task definition. However, in some settings, the extrinsic reward may be absent or sparse. Even if that is the case, an autonomous agent should be able to learn skills. To deal with this situation, intrinsic rewards are used in several RL studies. Some studies used only intrinsic reward [33], [38], [39], whereas others studied how to combine intrinsic and extrinsic rewards in RL settings [32], [40], [41]. Intrinsic motivation was also applied to RL at the different levels of the hierarchies [42]–[44]. All of these studies aim to make the agent learn skills to achieve a specific goal. By contrast, in our study, there is no particular goal that the agent needs to accomplish.

b) Developmental Robotics: In the seminal computational architecture of Oudeyer et al. [16], the sensorimotor space was incrementally split into regions, and the regions were learned by the local experts. Selection between the regions was made by considering the learning progress IM signal. In our previous work [45], similar to [16], we partitioned the sensorimotor space by considering a single parameter at each partitioning step in order to form exploration regions. Forestier et al. [46] developed an algorithmic procedure called "intrinsically motivated goal exploration processes" (IMGEP) that allows the autonomous discovery of skills in an open-ended learning setting. In their approach, the agent selected the goal to pursue using intrinsic motivation signal and learned skills by self-experimentation. As a result, the agent learned to discover and accomplish goals by following a self-generated curriculum with an increasing level of complexity. Mannella et al. [47] hypothesized that an agent learns the dynamics of its body by autonomous goal generation regulated by intrinsic motivation. To validate their hypothesis, they created a model that relied on an intrinsic motivation signal to form abstract representations of the observations and select goals to pursue and learn motor skills. Haber et al. [48] proposed a computational model of intrinsic motivation where the understanding of ego-motion, followed by the ability to interact with single and multiple objects, emerges from novelty-seeking exploration. In our current work, different from the previous studies, IM-exploration regions are formed by clustering a latent space that combines object, action, and outcome information.

B. Representation Learning in Robotics and IM

Most of the work in robot learning utilizes engineered feature representations to perform given tasks. However, to obtain full autonomy in intelligent systems, the agent also should be capable of building efficient feature representations

from raw sensory data. Representation learning in robotics is an important research direction that allows the learning systems to be efficient in computational resources, generalization ability, time efficiency, and abandons the need for feature engineering. Various studies in domains of robot learning [49], [50], planning [51], [52], control [52]–[54], and RL [33], [40], [41], [43], [55]–[58] focus on learning representations to foster autonomy. Among these, a number of studies utilized representation learning in IM-based exploration [33], [56], [59]–[62]. Bugur et al. [63] proposed an intrinsically motivated exploration scheme in action space. In their study, the action and effect space information was used to obtain a latent representation from which two regions are obtained for exploration via IM. Laversanne-Finot et al. [59] integrated a representation learning stage on top of IMGEP [46] to create the goal-spaces by encoding raw sensory observations. In that study, the agent first passively observes the environment to collect data for learning an embedding function. After that stage, learned representation was used to form goal spaces to be explored by the intrinsically motivated architecture they proposed previously. Hafez et al. [40] proposed an Actor-Critic algorithm that enables the learning of motor skills directly from visual observations in an RL setting. In their work, an embedding of visual input was used in actor and critic networks to create exploration regions incrementally utilizing Self-Organizing Maps. Like our work, each region has a prediction model whose learning progress is then used to guide the exploration. In summary, almost all these studies considered only the observation space to form the latent space, and [63] considered only action and effect space. In contrast, we exploit a latent representation that integrates high-dimensional object features, action parameters, and outcome observations in region formation and IM-based exploration.

III. Proposed system

A. Overview and General Flow

Fig. 1 illustrates the general framework and the learning cycle proposed in this study. Recall that our aim is to partition the object-action-outcome exploration space of the robot into regions and enable the robot to explore these regions in the most efficient way via IM. The upper panel of the figure shows how these regions are formed in a bootstrapping phase, and the lower panel shows how these regions are selected in each IM-based exploration step. As shown in the upper panel, to bootstrap the region formation, the simulated robot (shown on the left) undergoes a short exploration phase, in which it interacts with a set of objects via randomly parameterized actions and observes the outcome of its actions. In each interaction, the information of the object (depth image), action (arm and hand parameters), and outcome (change in object position and orientation) are collected. Using the data set obtained from these exploratory random interactions, the regions for predictive learning are found in two steps. First, the processed depth image (shown in (A)), action, and outcome features are blended together and mapped to a low-dimensional latent space, as shown in (B). Second, a clustering algorithm is applied to find regions for predictive learning in the latent space, as shown in (C). In the IM-based active learning phase, shown in the lower panel, a forward model that predicts the outcome given object and action features is trained for each region (E), and the region whose forward model exhibits the highest learning progress is selected for further learning (D). After a pair of object and action (parameter vector) is sampled from the selected region, the robot observes the outcome of the application of the sampled action (bottom-left) and updates the corresponding forward model (E) and the learning progress statistics of the region (D).

B. Object-Action-Outcome Representations

In each interaction, the robot executes its parametric action on an object and observes the outcome.

- **Object:** The top-down depth image of the object, taken before the execution of the action, is encoded through a Convolutional Autoencoder (CAE) into a low-dimensional feature vector (Fig. 1(A)), I_{enc} (8D). Hence, the object information to the system is represented by this low dimensional feature vector.
- **Action:** We assume that the robot is equipped with a basic movement capability involving the arm and the fingers, which we call the *reach and enclose* action. The action is parameterized and set to generate a semi-circular hand trajectory (see Fig. 2), mimicking a human-like radial motion allowing basic object interactions. The robot action parameters vector (5D) controls the radius of the hand trajectory (1D), the direction of the approach towards the object (1D), and the end-effector state (3D).
- **Outcome:** The outcome of an action is defined as the position and orientation change of the object. Thus, it is represented by a vector (5D) composed of the position change in the three coordinate axes and (sin&cos values of the) orientation change around the vertical axis.

C. Bootstrapping Region Formation

Formation of the Latent Space The interaction experience obtained from a short random exploration phase is used to form the latent space. The object, action, and outcome vectors are concatenated in a single feature vector for each interaction and processed via a feature extractor to form the latent space that compactly represents these three elements of the interaction (Fig. 1(B)). As the feature extractor, a Variational Autoencoder (VAE) with Gaussian prior was used. Following the input layer (18D), the encoder part of the VAE has an intermediate layer (9D) with ReLU non-linearity, followed by a hidden layer (3D) that is split into $\mu(z)$ and $\sigma(z)$ so that the network output can be considered to represent a Gaussian distribution [64]. The decoder part has a structure that is symmetrical to the encoder part. It takes z that is sampled from the encoder's output and has an output layer with sigmoid non-linearity. Binary cross-entropy is used as the reconstruction loss, and the VAE loss is calculated as in [64]. The VAE is trained with Adam [65] optimizer with a batch size of 100. The latent space is formed by using the $\mu(z)$ from the encoder's output.

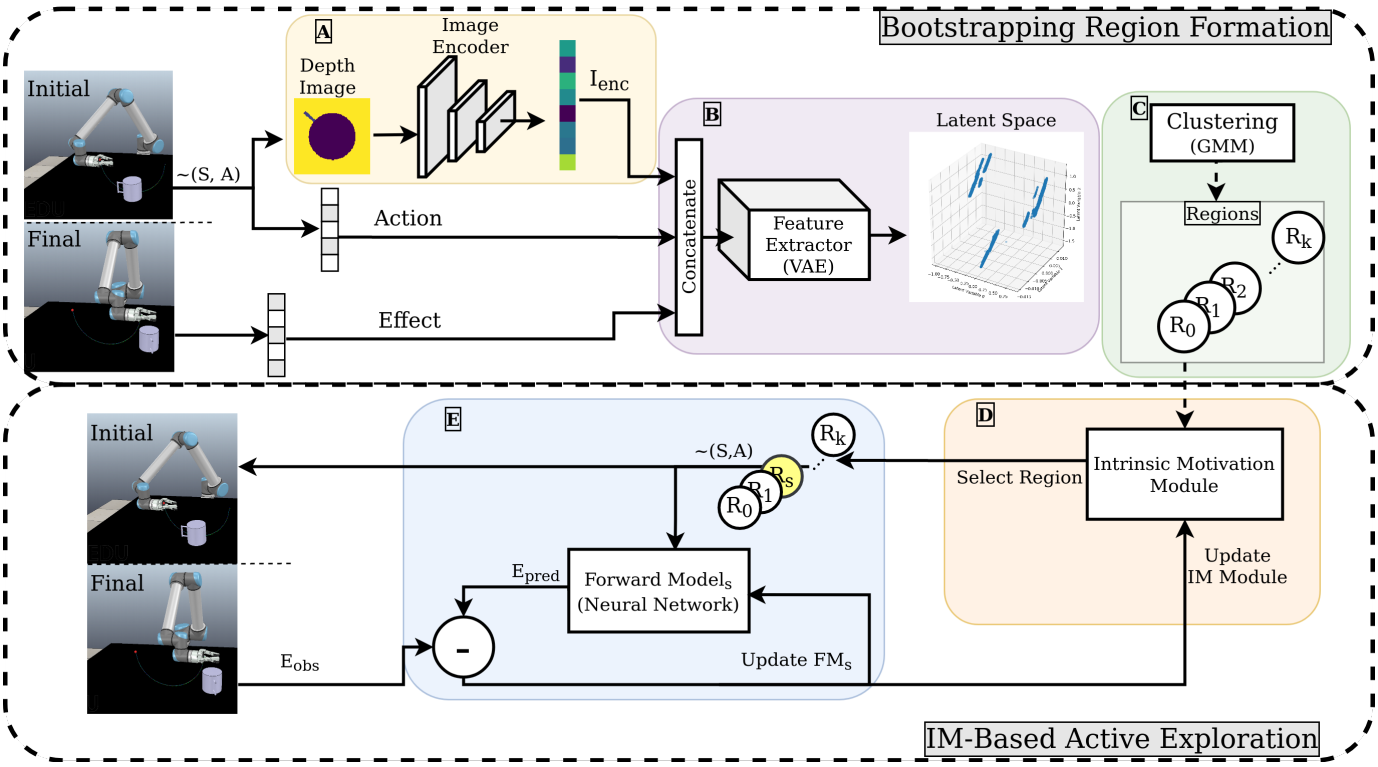

Fig. 1. The overview of the proposed framework and learning cycle. The regions are formed via random exploration, as shown in the upper panel, and actively selected for exploration by the IM module, as shown in the lower panel. See the details in the text.

Formation of the Exploration Regions To form the regions for forward model learning, the latent space is clustered using the Gaussian Mixture Model (GMM) algorithm with an empirically chosen number of clusters five in the current experiments, see Appendix for an analysis on this parameter), (Fig. 1(C)) where each cluster corresponds to a "region" (R_i) that the robot can build a local forward model for action outcome prediction. Note that the regions found in this step were frozen and not changed during IM-based predictive learning for computational convenience.

D. IM-based Active Exploration

Local Prediction Models Each region (R_i) found in the bootstrapping phase (Fig. 1(C)) is assigned to a forward model (FM) that is responsible for predicting the outcome given the object features (I_{enc}) and the action parameters in that region (Fig. 1(E)). The FMs are implemented as one hidden-layer feed-forward neural networks. Input, hidden, and output dimensions are set to 13, 512, and 5. The hidden unit non-linearity is provided by the ReLU activation function. At each predictive learning step, one FM is allowed to learn (see below). The learning in FMs is carried out by back-propagating the prediction error calculated as the mean square error (MSE). At each exploration step, a small batch (κ) is sampled from the FM's responsibility region. In order to avoid *catastrophic forgetting* [66], [67], the FM is continued training[1] with all the data it has encountered so far. To avoid overfitting, the FM is trained for only a small number (5) of epochs at each step.

Efficient Predictive Learning Learning progress based IM is used to select which region to target for improving the prediction ability (Fig. 1(D)). Intrinsic Motivation Module keeps statistics about the (FM) learning progress of each region. In each step, it selects the region with the highest learning progress using the ϵ-greedy [69] selection mechanism, and (object, action) pairs are sampled corresponding to the selected region for interaction.

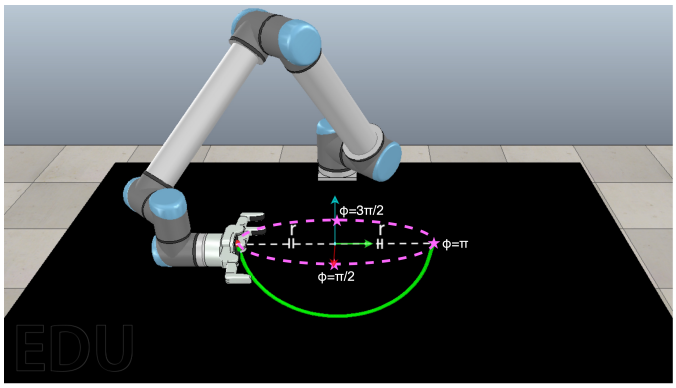

Fig. 2. The end-effector follows a semi-circular trajectory above the surface of the table. The action parameters control the radius of the semi-circle (r_{path}) and the approach position of the end-effector to the object (ϕ_{path}).

[1]FMs are implemented by using Keras [68] deep learning library.

Learning progress (LP) of a region is calculated from the prediction performance change of the corresponding FM after a given learning update cycle:

$$LP_n(t+1) = \gamma_n(t+1) - \gamma_n(t+1-\theta), \quad (1)$$

where γ_n indicates the mean error of FM_n at update cycle t, and is calculated as follows:

$$\gamma_n(t+1) = \frac{\sum_{i=0}^{\theta} e_n(t+1-i)}{\theta+1} \quad (2)$$

where the error of n^{th} region $e_n(t)$ is calculated by the MSE between the predicted effect E_{pred} and the observed effect E_{obs}. The window parameter θ allows the system to capture the trend of the errors by averaging them within a given learning period and prevents the fluctuations from affecting the IM signal. In general, a small θ makes the LP measure highly unstable, whereas a large θ makes the LP changes less precise [12]. Although in our experiments, θ is empirically set to 16 for computational convenience, it is possible to implement a mechanism to determine it (e.g., see [70]) automatically.

IV. Experiment Setup

The experiment setup was simulated in CoppeliaSim [71], where a six-degrees-of-freedom robot arm with a gripper (UR10)[2] was chosen as the manipulator to be used in the experiments. In order to capture the basic interaction infants create with their environment, a simple setup with three types of objects was created, which the robot could interact with through its *reach and enclose* actions. The details of the objects used, the action parameters, and the outcome representation are given below.

Objects: Three objects were used in the experiments with some changing sizes: a cup, a cylinder, and a sphere (Fig 3). The cup has a fixed height (15 cm) and radius (7.5 cm) and has a handle that is 12.5 cm apart from the center with a length of 10 cm. The orientation of the cup is changed around the vertical axis within $[0, 2\pi]$, i.e., the position of the handle varies around the body of the cup. Cylinders have a fixed height of $h = 15$ cm and radius within the range of $[1.5, 7.5]$ cm, and spheres have a radius within the range of $[3, 7.5]$ cm. A simulated Kinect camera is positioned on top of the table to record 128×128 top-down depth images of the objects.

Actions: The end-effector of the robot follows a semi-circular trajectory that has start and end-points with the same elevation from the tabletop. The closest point of the trajectory to the table is the halfway point, and it has a fixed offset from the surface of the table to avoid collision between the end-effector and the table. The semi-circular trajectory is defined by the radius of the semi-circle $r_{path} = [26, 31]$ cm and a z-orientation within $\phi_{path} = [0, 2\pi]$ radians. ϕ_{path} controls the approach direction of the end-effector to the object, i.e., determines the via point that the end-effector will pass while approaching the object (see Fig. 2). When the end-effector interacts with the object, it takes one of three states: *closed*, *half-open*, and *open*, and the fingers are enclosed, similar to a reflex, as soon as the object is contacted. In summary, the action parameter vector A is composed of a 5-dimensional vector $(r_{path}, \phi_{path}, closed, half open, open)$ where the last three parameters are binary and represent the one-hot encoding of the gripper state.

[2]https://www.universal-robots.com/products/ur10-robot/

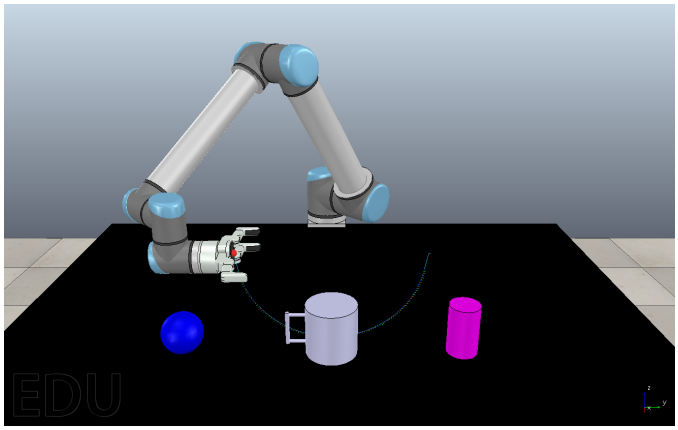

Fig. 3. The experiment setup. A manipulator robot interacts with one of three types of objects in a table-top environment.

Outcome: The outcome is defined as the change in the 3D position of the object, together with the (sine and cosine of the) orientation angle change with respect to the vertical axis: $O = (\Delta x, \Delta y, \Delta z, \sin\Delta\phi_z, \cos\Delta\phi_z)$. The outcome is calculated by taking the difference between the first and final pose of the object. We used sine and cosine values of $\Delta\phi_z$ to ensure continuity at the fundamental boundaries of the domain of sine and cosine. Note that, even if the robot executes the action with *open* and *half-open* end-effector aperture configurations, it may not be able to grasp and raise the object. This can be caused by misalignment of the object size, object pose, end-effector pose, and simulation noise. For example, if the robot approaches with an open end-effector to the cup from the side of its handle, due to the contact of the handle with the robot fingers, the object rotates and is pushed out of the finger enclosure; hence it can not be grasped even if the fingers are enclosed.

For each interaction, the simulation scene is reset, and the parameters of the selected type of object and actions are sampled from their corresponding intervals. Overall, the dataset consists of three different object types with three different end-effector states, each consist of 5184 interactions, in total $3 \times 3 \times 5184 = 46656$ interactions.

System Hyper-Parameters:

- The convolutional auto-encoder, whose bottleneck layer (8D) serves as the object features (I_{enc}), consists of stacks of convolutional layers followed by batch normalization and max-pooling operations, with channel numbers 512, 256, 128, 64, 32, 16, and 8. It is trained using binary cross-entropy as the reconstruction loss and Adadelta [72] optimizer.
- The initial bootstrapping phase uses 700 random interactions for region formation. After the bootstrapping phase, the FMs are initialized with an initial set of 128

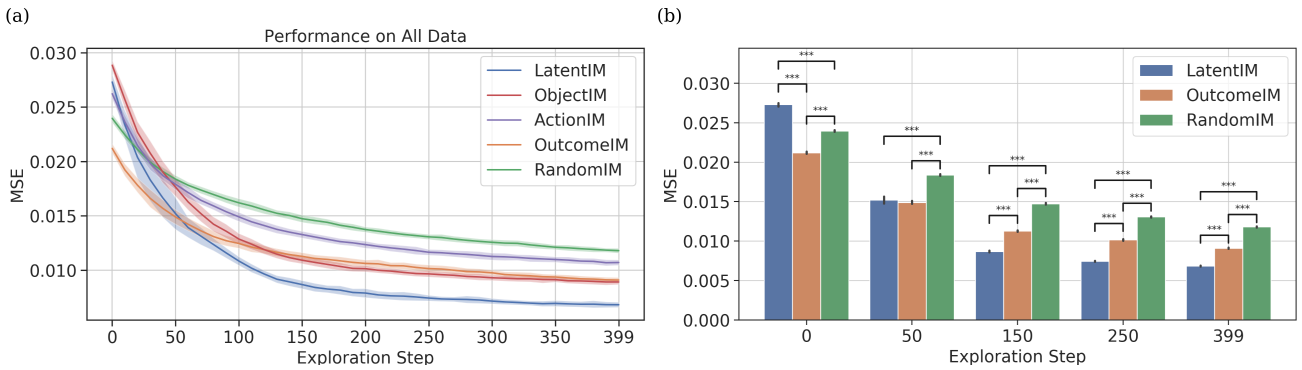

Fig. 4. Comparison of the prediction performances of LatentIM, OutcomeIM, ObjectIM, ActionIM and RandomIM. (a) shows the change in the average MSE during the IM-based active exploration phase of 40 independent runs. The shaded areas show the standard deviation. (b) shows the same data but comparing only LatentIM, OutcomeIM, and RandomIM, showing the statistical significance.

interactions, and the selected ones by IM are continued to be trained with the past interactions plus newly sampled $\kappa = 16$ interactions for 400 exploration steps. The number of regions is set to 5. Thus, the IM-Based active exploration phase uses approximately 7000 data.

- The other parameters are set as follows: $\epsilon = 0.3$, $\theta = 16$.

V. Results

In this section, we analyzed the results of our latent space based IM approach (LatentIM), and compared it with the alternatives that use only object (ObjectIM), action (ActionIM), and outcome (OutcomeIM) spaces in region partitioning with the same number of regions. These variants are the adapted versions of existing IM methods that utilize only object [25], action [12], and outcome [26]. As a basic baseline, we also provided the results of RandomIM that assigns regions to the data points randomly. We conducted experiments to answer the following questions:

1) How does the method of region formation affect the overall performance? (Section V-B)
2) What is the exploration order of prediction capabilities? (Section V-C)
3) What is the effect of the different hyper-parameters (number of clusters and ϵ), using inverse models, and using different implementations of the components, on overall performance? (Appendix)

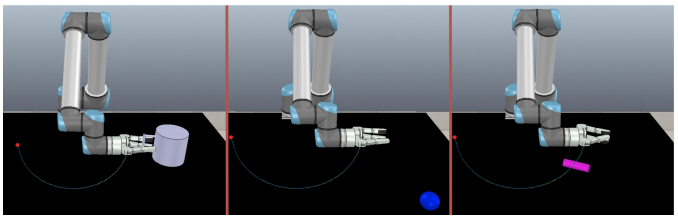

Fig. 5. The final snapshots from sample interactions of (l_pinch), (n_pinch), and (n_close) regions.

A. Regions formed by LatentIM

We analyzed the regions formed in the latent space and identified the following segregation: Region 1 includes actions with *half-open* gripper and objects with no change in z position; region 2 includes actions with *open* gripper and objects with no change in z position; region 3 includes *half-open* gripper and objects with changes in z position; region 4 includes *open* gripper and objects with changes in z position; region 5 includes *closed*. Considering these characteristics, we named region 1 as non-lifting pinch-grasp (n_pinch), region 2 as non-lifting power-grasp (n_power), region 3 as lifting pinch-grasp (l_pinch), region 4 as lifting power-grasp (l_power) and region 5 as non-lifting (n_close). Note that these labels are given to help the reader, and the system did not use any given labels. Sample snapshots from interactions of (l_pinch), (n_pinch), and (n_close) regions are provided in Fig. 5.

B. Comparison of Overall Performances

To investigate the effect of the region formation on the overall performance of the system, we analyzed five different models, namely LatentIM, ObjectIM, ActionIM, OutcomeIM, and RandomIM. Fig. 4(a) shows the average MSE calculated with Eq.(3) over 40 independent runs:

$$MSE = \frac{\sum_{i=1}^{N} n_i.\psi_i}{\sum_{i=1}^{N} n_i} \tag{3}$$

$$\psi_i = \frac{1}{n_i} \sum_{j=1}^{n_i} (E^i_{obs} - E^i_{pred})^2, \tag{4}$$

where ψ_i stands for the mean squared error made by region i; N and n_i indicate the number of regions and the number of data points in region i, respectively; E^i_{obs} and E^i_{pred} represent the observed and predicted outcomes in region i, respectively.

Note that the initial training of FMs with 128 samples is not included in the plot. As presented in the figure, LatentIM gives the lowest error among all five models. Following LatentIM, OutcomeIM, and ObjectIM perform similarly; the only difference between those two is that OutcomeIM is better at the beginning, but ObjectIM shows a more rapid decrease

in the MSE. It seems that among all the methods, OutcomeIM benefits from the initial set of interactions most, considering that it groups similar outcomes, i.e., the data distribution among its regions is more coherent than the others.

Depending on the actions applied, similar objects may give rise to observe different outcomes, and similar outcomes may be observed by applying similar actions on different objects. Therefore, observing that the performance of ObjectIM is close to OutcomeIM while ActionIM achieves a lower performance is an interesting result for us. This situation might be linked with using object-related information from the encoded representation, i.e., different actions might be more informative than the different I_{enc} to determine the outcome. ObjectIM and ActionIM are not included in the rest of the paper for the readability and clarity of the figures. We considered OutcomeIM as the competitor of our method and RandomIM as the baseline.

In Fig. 4(b), we present the statistical analysis of the differences between LatentIM, OutcomeIM, and RandomIM taken from the different exploration steps. We ran an analysis of variance (ANOVA) to check whether the MSE distributions of these three approaches are different. Then we carried out post-hoc ANOVA tests, i.e., Tukey's HSD and Games-Howell Test, depending on the equal and non-equal variance cases, respectively. We found that after $t = 50$, the performance of LatentIM and OutcomeIM differs significantly $(p < 0.001)$, LatentIM giving more accurate predictions.

C. Exploration Order of Skill Prediction

In this subsection, we analyzed the exploration order of regions and skill prediction that is regulated by the IM module. The analysis of the exploration order with single runs of LatentIM, OutcomeIM and RandomIM are presented in Fig. 6, and the average of 40 runs of LatentIM is presented in Fig. 7.

A positive LP value means that the predictions of the FM are improving. At each time step, the region with a higher LP value is most likely (due to the ϵ-greedy region selection) to be selected for the exploration. Fig. 6(a) shows the learning progress values throughout the IM based active exploration phase of LatentIM regions in a single run. The regions of LatentIM show a clear ordering. It first explores the lifting grasps (l_power & l_pinch), then shifts its attention to n_pinch, n_close, and n_power, respectively. Note that the order of skills may change across different runs due to the randomness involved in ϵ-greedy region selection and sampling inside the regions. Detailed investigation and statistical analysis of the exploration order formed by LatentIM will be discussed later in this subsection.

In Fig. 6(b), OutcomeIM gives priority to $R2$, which corresponds to interactions with the cup object with x, y position and orientation change. Following this, no clear ordering over the regions is observed. Similarly, RandomIM does not produce a distinctive exploration order (Fig. 6(c)). Note that through the end of the exploration phase, for all the strategies, the LP values reach zero because of the strong FM predictors that can nevertheless learn their regions from the data points.

Fig. 7 shows the mean change in the learning progress of the regions for LatentIM from 40 independent runs. The lines and shades correspond to the mean and standard deviations of the learning progress at the corresponding time steps. As shown in Fig. 7, we observe a consistent ordering as l_pinch, l_power, n_pinch, n_close, and n_power on average. This ordering is reasonable because, when grasped, the orientation change is ≈ 0. However, when pushed, the object may turn around or tumble. Thus, when the robot lifts the object, the effect is more predictable than the rest; hence the corresponding region was easier to learn.

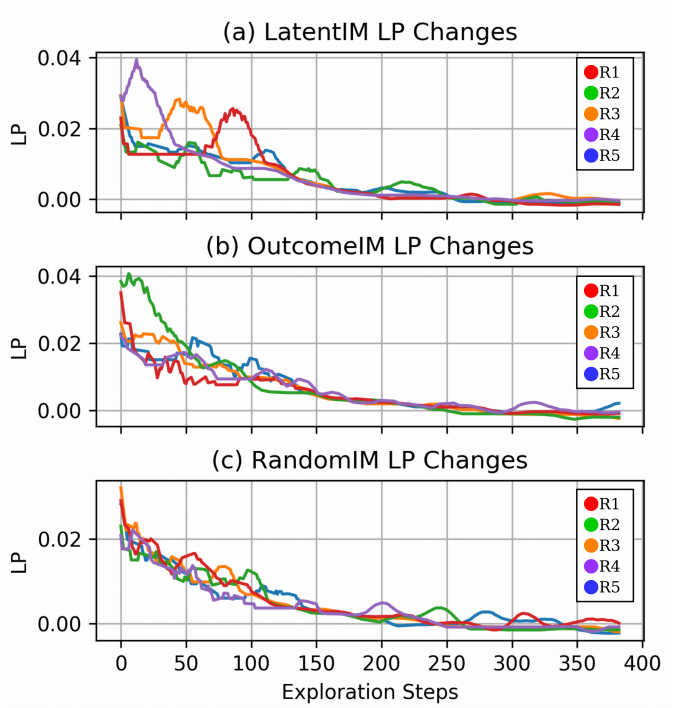

Fig. 6. Changes in learning progress of 5 regions during the IM-based active exploration phase of a single run. The plot shows learning progress values for (a) LatentIM, (b) OutcomeIM, and (c) RandomIM. Note that the learning progress signal is smoothed by a window ($\alpha = 16$) to make it easier to view. For LatentIM in (a), R1, R2, R3, R4 and R5 correspond n_pinch, n_power, l_pinch, l_power, and n_close, respectively. For OutcomeIM and RandomIM, no clear correspondence can be observed.

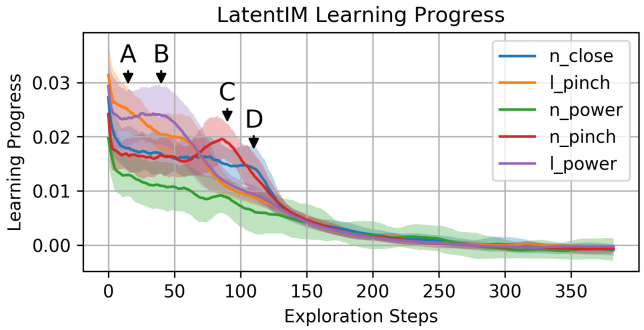

Fig. 7. Change of mean learning progress of 5 regions during the IM-based active exploration phase. The plot is smoothed by a time window ($\alpha = 16$). The plot shows the mean learning progress values of the LatentIM collected from 40 experiments. The shaded areas show the standard deviation. Please refer to the text for the statistical analysis of the learning progress values at points A, B, C, and D.

To evaluate the significance of the ordering presented in Fig. 7, A Kruskal-Wallis test was performed on the learning progress of the five different regions. The differences between the learning progress distributions of the regions taken from the interval $t = [0, 139]$ are significant with $H(4)$, $p < 0.01$. Following that, we also performed the Mann-Whitney U test to determine the significance of the learning progress values for the pairs of regions. In Fig. 7, first l_pinch has the maximum learning progress value. Taken from that interval at $Point\ A$ the LP of l_pinch is significantly greater than of l_power, $p < 0.05$, and the LP of l_power is significantly greater than the rest with $p < 0.001$. At the same time-step, n_pinch is not significantly different from n_close, while the LP of n_close being significantly greater than of n_power, $p < 0.01$. Following the plot, we see the dominance of l_power over l_pinch. Taken from that time window, at $Point\ B$, LP of l_power is significantly greater than of l_pinch, $p < 0.05$. While the LP of l_pinch is significantly greater than of n_close and n_pinch with $p < 0.001$, the difference between n_close and n_pinch is not significant, both being greater than n_power with $p < 0.01$. After the decrease of l_pinch and l_power, a significant increase in n_pinch is visible in Fig. 7. Being within that time window, at $Point\ C$, the LP of n_pinch is significantly greater than of n_close, $p < 0.05$. At the same time, LP of l_pinch is significantly less than of n_close with $p < 0.05$ it is significantly greater than of n_power, $p < 0.001$. And finally, there is a short primacy of n_close over the rest, $Point\ D$, the LP of n_close is significantly greater than n_pinch, $p < 0.05$.

VI. Discussion and Conclusion

Our experiment results show that the system can produce a sensorimotor learning curriculum that resembles some features of infants' sensorimotor development. This capability is attributable to the main ingredients of our work: (1) the latent space formed from object-action-outcome and (2) learning progress prioritization of local learning within the latent space. Besides exhibiting developmentally plausible learning, the proposed system facilitates the development of better prediction ability by smartly distributing the exploration among the local learning modules defined over the blended latent space.

Exploration order of prediction skills. The proposed system developed prediction ability in a staged manner for basic grasp actions before the prediction skill for purposeful push actions. This was an emergent feature realized through the coupling of learning progress based intrinsic motivation with the object-action-outcome blended space and resembles the order of emergence between grasp and push actions in infants. However, the order within the grasp action types was not the same as infant development. Since the precision pinch grasp requires finer control and precise movements in infants, it emerges later compared to the power grasp [2], [27]. Thus its related prediction ability should develop later as well. However, in our simulations, this order was reversed. The reason for this is easy to see, as in our simulations, the execution of the precision and power grasps has no differential difficulty due to the action parameterization used. Moreover, the robotic gripper used does not favor a power grasp, unlike a human hand that naturally conforms to the shape of the object once a basic hand enclosure is initiated [73]. On the contrary, the robotic gripper is more suitable for a precision pinch by design. Thus, in our experiments, the learning progress for pinch grasp turned out to be higher than that of power grasp, prioritizing the development of the prediction ability for precision pinch grasp. Yet, overall, the observed step-by-step improvement of skills in our experiments can be seen akin to the staged development observed during infancy [27].

Functional region emergence. Another important feature the proposed system developed is that the regions formed over the blended space corresponded to well-defined semantically meaningful action-outcome primitives. In particular, by analyzing the discovered regions, we could identify *push*, *grasp*, and *near-grasp* actions. It could be questioned why clear object-based regions were not formed. The answer lies in the compact blended representation that finds the categorical actions (qualitatively different outcome yielding motor parameters) as a better descriptor for the triplets (action, object, outcome). This is comparable to the sensorimotor brain organization of primates for action, where the dorsal where/how pathway represents the objects in terms of features related to manipulation affordances, but not object identification [74], [75].

Superiority of the LatentIM over the other variants. Initially, it was not clear whether using a latent space to define the local models and exploring this latent space with IM would yield better predictors. Yet, our experiments with forward model learning showed that the latent space based IM significantly outperformed other IM approaches that use spaces only of the object, action, or outcome in terms of the learning speed and prediction accuracy. Moreover, our experiments with inverse model learning (see Appendix) also supported the superiority of LatentIM over the other variants. Particularly, we have found that LatentIM and ActionIM outperformed the other three approaches, while LatentIM showed a lower prediction error than ActionIM. Furthermore, the emergence of regions and exploration order that we discussed above has only been observed clearly with the latent space based IM.

Implementation of the components. We believe that the general framework proposed well addresses the use of a diverse set of features observed during interactions in guiding IM exploration, whereas the particular implementation details might vary. For example, Variational Autoencoder (VAE), Gaussian Mixture Model (GMM), and Feed-Forward Neural Networks (FFNN) were used for latent space formation, the formation of exploration regions, and the effect prediction, respectively. Regarding these particular methods, we conducted experiments that used alternative methods with different capabilities (see Appendix). Our analysis showed that dimensionality reduction, only with linear transformation (e.g., PCA) or clustering without variance information (e.g., K-Means) did not change the prediction performance significantly. Furthermore, the regions of LatentIM formed with such methods appeared to be similar to those presented in Section V-A. However, we found that using an FFNN with a linear activation function degraded the prediction performance for all approaches.

Limitations. Finally, we would like to identify a number of limitations and possible future directions. First of all, the agents would encounter different situations and experience different interactions in a life-long learning scenario; therefore, mechanisms that allow assimilation and accommodation [76] of regions should be investigated. Regions found in our study reflect object-action or action-outcome synergies; however, regions corresponding to individual objects, actions, or outcomes might also emerge in increasingly more complex environments. As stated in Section IV, our experiment setup is a simplified version of infants' play environments, e.g., the variety of the objects, actions, and outcomes are limited, and the robot fixates only one object at each interaction. In order to handle multiple objects, attention mechanisms would be required. In future work, along with an attention mechanism, the scalability of our method to more complex environments will be tested. In addition, an experimental design that uses depth images for the outcome features is an exciting direction for our future research, especially in real-world scenarios with occlusions. As in [59], [77], using the vision system for outcome features in a setup that includes distractor objects would be a worthwhile challenge for the intrinsic motivation studies. We would like to study this issue in future work with a real-world application. Another point is that we used VAE for only latent space formation; however, the computational effort used in forming the latent space could have been exploited for also FM formation. In the current implementation, to ease the analysis, we decoupled latent space formation and FM learning by having separate mechanisms. As a future study, it would be interesting to explore the developmental progression of the system when a single neural architecture is used for both latent space formation and FM learning.

ACKNOWLEDGMENT

This research was partially supported by JST CREST "Cognitive Mirroring" [grant number: JPMJCR16E2] and TÜBİTAK (Scientific and Technological Research Council of Turkey) 2210-A scholarship. The authors would like to thank Serkan Bugur and Mert Imre for their comments on this study and this manuscript. We thank Mete Tuluhan Akbulut for proofreading this manuscript.

REFERENCES

[1] Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

[2] Angelo Cangelosi and Matthew Schlesinger. *Developmental robotics: From babies to robots*. MIT press, 2015.

[3] Robert W White. Motivation reconsidered: The concept of competence. *Psychological review*, 66(5):297, 1959.

[4] Daniel E Berlyne. Curiosity and exploration. *Science*, 153(3731):25–33, 1966.

[5] Edward L. Deci. *Intrinsic Motivation*. Springer US, Boston, MA, 1975.

[6] Edward L Deci and Richard Ryan. Intrinsic motivation and self-determination in human behavior. *New York: Plenum. doi*, 10:978–1, 1985.

[7] Jacqueline Gottlieb, Manuel Lopes, and Pierre-Yves Oudeyer. *Motivated Cognition: Neural and Computational Mechanisms of Curiosity, Attention, and Intrinsic Motivation*, volume 19 of *Advances in Motivation and Achievement*, pages 149–172. Emerald Group Publishing Limited, 2016.

[8] P-Y Oudeyer, Jacqueline Gottlieb, and Manuel Lopes. Intrinsic motivation, curiosity, and learning: Theory and applications in educational technologies. In *Progress in brain research*, volume 229, pages 257–284. Elsevier, 2016.

[9] Serena Ivaldi, Natalia Lyubova, Alain Droniou, Damien Gerardeaux-Viret, David Filliat, Vincent Padois, Olivier Sigaud, Pierre-Yves Oudeyer, et al. Learning to recognize objects through curiosity-driven manipulation with the icub humanoid robot. In *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–8. IEEE, 2013.

[10] Nicolas Duminy, Sao Mai Nguyen, and Dominique Duhaut. Learning a set of interrelated tasks by using a succession of motor policies for a socially guided intrinsically motivated learner. *Frontiers in neurorobotics*, 12:87, 2019.

[11] Pierre Fournier, Cédric Colas, Mohamed Chetouani, and Olivier Sigaud. Clic: Curriculum learning and imitation for object control in non-rewarding environments. *IEEE Transactions on Cognitive and Developmental Systems*, 2019.

[12] Emre Ugur and Justus Piater. Emergent structuring of interdependent affordance learning tasks using intrinsic motivation and empirical feature selection. *IEEE Transactions on Cognitive and Developmental Systems*, 9(4):328–340, 2016.

[13] Alexandre Manoury, Sao Mai Nguyen, and Cédric Buche. Hierarchical affordance discovery using intrinsic motivation. In *Proceedings of the 7th International Conference on Human-Agent Interaction*, pages 186–193, 2019.

[14] Gianluca Baldassarre, William Lord, Giovanni Granato, and Vieri Giuliano Santucci. An embodied agent learning affordances with intrinsic motivations and solving extrinsic tasks with attention and one-step planning. *Frontiers in neurorobotics*, 13:45, 2019.

[15] Sebastian Blaes, Marin Vlastelica Pogančić, Jiajie Zhu, and Georg Martius. Control what you can: Intrinsically motivated task-planning agent. In *Advances in Neural Information Processing Systems*, pages 12520–12531, 2019.

[16] Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

[17] Jürgen Schmidhuber. Curious model-building control systems. In *Proc. international joint conference on neural networks*, pages 1458–1463, 1991.

[18] Celeste Kidd, Steven T Piantadosi, and Richard N Aslin. The goldilocks effect: Human infants allocate attention to visual sequences that are neither too simple nor too complex. *PloS one*, 7(5):e36399, 2012.

[19] Yukie Nagai. Predictive learning: its key role in early cognitive development. *Philosophical Transactions of the Royal Society B*, 374(1771):20180030, 2019.

[20] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[21] Daniel M Wolpert and Mitsuo Kawato. Multiple paired forward and inverse models for motor control. *Neural networks*, 11(7-8):1317–1329, 1998.

[22] Mitsuo Kawato. Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727, 1999.

[23] He Cui. Forward prediction in the posterior parietal cortex and dynamic brain-machine interface. *Frontiers in integrative neuroscience*, 10:35–35, 2016.

[24] E. Oztop, D. Wolpert, and M. Kawato. Mental state inference using visual control parameters. *Brain Research: Cognitive Brain Research*, 22(2):129–51, 2005.

[25] Emre Ugur, Mehmet R Dogar, Maya Cakmak, and Erol Sahin. Curiosity-driven learning of traversability affordance on a mobile robot. In *2007 IEEE 6th International Conference on Development and Learning*, pages 13–18. IEEE, 2007.

[26] Adrien Baranes and Pierre-Yves Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013.

[27] Rebecca J Scharf, Graham J Scharf, and Annemarie Stroustrup. Developmental milestones. *Pediatrics in review*, 37(1):25, 2016.

[28] Manuel Lopes and Pierre-Yves Oudeyer. Guest editorial active learning and intrinsically motivated exploration in robots: Advances and challenges. *IEEE Transactions on Autonomous Mental Development*, 2(2):65–69, 2010.

[29] Vieri Giuliano Santucci, Gianluca Baldassarre, and Marco Mirolli. Which is the best intrinsic motivation signal for learning multiple skills? *Frontiers in neurorobotics*, 7:22, 2013.

[30] Marco Mirolli and Gianluca Baldassarre. Functions and mechanisms of intrinsic motivations. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 49–72. Springer, 2013.

[31] Rania Rayyes, Heiko Donat, and Jochen Steil. Efficient online interest-driven exploration for developmental robots. *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[32] Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288, 2005.

[33] Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 2125–2133, 2015.

[34] Douglas Blank, Deepak Kumar, Lisa Meeden, and James B Marshall. Bringing up robot: Fundamental mechanisms for creating a self-motivated, self-organizing architecture. *Cybernetics and Systems: An International Journal*, 36(2):125–150, 2005.

[35] Adrien Baranes and Pierre-Yves Oudeyer. R-iac: Robust intrinsically motivated exploration and active learning. *IEEE Transactions on Autonomous Mental Development*, 1(3):155–169, 2009.

[36] Clément Moulin-Frier, Pierre Rouanet, and Pierre-Yves Oudeyer. Explauto: an open-source python library to study autonomous exploration in developmental robotics. In *4th International Conference on Development and Learning and on Epigenetic Robotics*, pages 171–172. IEEE, 2014.

[37] Sébastien Forestier and Pierre-Yves Oudeyer. Modular active curiosity-driven discovery of tool use. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3965–3972. IEEE, 2016.

[38] Todd Hester and Peter Stone. Intrinsically motivated model learning for developing curious robots. *Artificial Intelligence*, 247:170–186, 2017.

[39] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[40] Muhammad Burhan Hafez, Cornelius Weber, Matthias Kerzel, and Stefan Wermter. Deep intrinsically motivated continuous actor-critic for efficient robotic visuomotor skill learning. *Paladyn, Journal of Behavioral Robotics*, 10(1):14–29, 2019.

[41] Tom Blau, Lionel Ott, and Fabio Ramos. Bayesian curiosity for efficient exploration in reinforcement learning. *arXiv preprint arXiv:1911.08701*, 2019.

[42] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems*, pages 3675–3683, 2016.

[43] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3540–3549. JMLR. org, 2017.

[44] Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 3303–3313, 2018.

[45] Melisa Idil Sener and Emre Ugur. Partitioning sensorimotor space by predictability principle in intrinsic motivation systems. In *2018 Joint IEEE 8th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 54–59. IEEE, 2018.

[46] Sébastien Forestier, Yoan Mollard, and Pierre-Yves Oudeyer. Intrinsically motivated goal exploration processes with automatic curriculum learning. *arXiv preprint arXiv:1708.02190*, 2017.

[47] Francesco Mannella, Vieri G Santucci, Eszter Somogyi, Lisa Jacquey, Kevin J O'Regan, and Gianluca Baldassarre. Know your body through intrinsic goals. *Frontiers in neurorobotics*, 12:30, 2018.

[48] Nick Haber, Damian Mrowca, Li Fei-Fei, and Daniel LK Yamins. Emergence of structured behaviors from curiosity-based intrinsic motivation. *arXiv preprint arXiv:1802.07461*, 2018.

[49] Rico Jonschkowski and Oliver Brock. State representation learning in robotics: Using prior knowledge about physical interaction. In *Robotics: Science and Systems*, 2014.

[50] Rico Jonschkowski and Oliver Brock. Learning state representations with robotic priors. *Autonomous Robots*, 39(3):407–428, 2015.

[51] Byron Boots, Sajid M Siddiqi, and Geoffrey J Gordon. Closing the learning-planning loop with predictive state representations. *The International Journal of Robotics Research*, 30(7):954–966, 2011.

[52] Ershad Banijamali, Rui Shu, Hung Bui, Ali Ghodsi, et al. Robust locally-linear controllable embedding. In *International Conference on Artificial Intelligence and Statistics*, pages 1751–1759, 2018.

[53] Sascha Lange, Martin Riedmiller, and Arne Voigtländer. Autonomous reinforcement learning on raw visual input data in a real world application. In *The 2012 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2012.

[54] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pages 2746–2754, 2015.

[55] Francisco Cruz, German I Parisi, Johannes Twiefel, and Stefan Wermter. Multi-modal integration of dynamic audiovisual patterns for an interactive reinforcement learning scenario. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 759–766. IEEE, 2016.

[56] Vieri Giuliano Santucci, Gianluca Baldassarre, and Marco Mirolli. Grail: a goal-discovering robotic architecture for intrinsically-motivated learning. *IEEE Transactions on Cognitive and Developmental Systems*, 8(3):214–231, 2016.

[57] Muhammad Burhan Hafez, Cornelius Weber, Matthias Kerzel, and Stefan Wermter. Efficient intrinsically motivated robotic grasping with learning-adaptive imagination in latent space. In *2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 1–7. IEEE, 2019.

[58] William Whitney, Rajat Agarwal, Kyunghyun Cho, and Abhinav Gupta. Dynamics-aware embeddings. In *International Conference on Learning Representations*, 2020.

[59] Adrien Laversanne-Finot, Alexandre Pere, and Pierre-Yves Oudeyer. Curiosity driven exploration of learned disentangled goal spaces. In *Conference on Robot Learning*, pages 487–504, 2018.

[60] Alexandre Péré, Sébastien Forestier, Olivier Sigaud, and Pierre-Yves Oudeyer. Unsupervised learning of goal spaces for intrinsically motivated goal exploration. In *International Conference on Learning Representations*, 2018.

[61] Alexandre Manoury, Cédric Buche, et al. Chime: An adaptive hierarchical representation for continuous intrinsically motivated exploration. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 167–170. IEEE, 2019.

[62] Guido Schillaci, Antonio Pico Villalpando, Verena V Hafner, Peter Hanappe, David Colliaux, and Timothée Wintz. Intrinsic motivation and episodic memories for robot exploration of high-dimensional sensory spaces. *Adaptive Behavior*, page 1059712320922916, 2020.

[63] Serkan Bugur, Erhan Oztop, Yukie Nagai, and Emre Ugur. Effect regulated projection of robot's action space for production and prediction of manipulation primitives through learning progress and predictability based exploration. *IEEE Transactions on Cognitive and Developmental Systems*, 2019.

[64] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[65] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[66] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

[67] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[68] François Chollet et al. Keras. https://keras.io, 2015.

[69] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[70] Guido Schillaci, Alejandra Ciria, and Bruno Lara. Tracking emotions: Intrinsic motivation grounded on multi-level prediction error dynamics. In *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 1–8. IEEE, 2020.

[71] Eric Rohmer, Surya PN Singh, and Marc Freese. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326. IEEE, 2013.

[72] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[73] Thomas E Twitchell. Reflex mechanisms and the development of prehension. *Mechanisms of motor skill development*, pages 25–38, 1970.

[74] M. A. Goodale, G. Króliczak, and D. A. Westwood. Dual routes to action: contributions of the dorsal and ventral streams to adaptive behavior. *Prog Brain Res*, 149:269–83, 2005.

[75] M. A. Goodale and A. D. Milner. Separate visual pathways for perception and action. *Trends Neurosci*, 15(1):20–5, 1992.
[76] Jean Piaget and Barbel Inhelder. *The Psychology of the Child*. Trans. Helen Weaver. New York: Basic Books, 1969.
[77] Cédric Colas, Pierre-Yves Oudeyer, Olivier Sigaud, Pierre Fournier, and Mohamed Chetouani. Curious: Intrinsically motivated modular multi-goal reinforcement learning. In *International Conference on Machine Learning*, pages 1331–1340, 2019.

APPENDIX

In this section, we provide additional experiment results to examine the effect of the hyperparameters (ϵ, and the number of clusters), using an inverse model for the IM-based active exploration and using different implementations of the subsystems on the prediction performance.

TABLE I
MEAN MSE VALUES FOR LATENTIM, OUTCOMEIM AND RANDOMIM WITH DIFFERENT ϵ VALUES.

	F-Score	μ**(LatentIM)**	μ**(OutcomeIM)**	μ**(RandomIM)**
ϵ = 0.0	1680.05	0.007121	0.009238	0.011895
ϵ = 0.5	3635.97	0.006753	0.009065	0.011748
ϵ = 0.7	3873.38	0.006724	0.009093	0.011675
ϵ = 1.0	4291.49	0.006819	0.008934	0.011766

Effect of ϵ Parameter

As explained in III-D, the ϵ parameter controls the ratio of exploration steps with random exploration to the ones with active exploration. We conducted $N = 30$ experiments for each ϵ value and verified our results with One-Way ANOVA $F(2, 87)$ tests followed by Tukey's HSD post-hoc on pairs (LatentIM vs. OutcomeIM, LatentIM vs. RandomIM, and OutcomeIM vs. RandomIM). The test yields that each pair is different with $p < 0.001$.

In Table I, we present the prediction errors of LatentIM, OutcomeIM, and RandomIM with different values of the ϵ parameter. For all the ϵ values, we observe that the LatentIM gives the lowest error among the other two. We also observe that except for $\epsilon = 0$, the performance of LatentIM does not change significantly.

In Fig. 8, we present the single run learning progress changes of LatentIM with different ϵ conditions. Increasing values of ϵ prevents seeing an ordering between different skills and does not provide a benefit for the predictor performance.

Effect of Number of Clusters

In Table II, we present the performance comparisons of LatentIM, OutcomeIM, and RandomIM with different numbers of clusters. For this experiment, we used the hyper-parameters given in Section IV, except for the number of clusters, which was the independent variable. The results show the mean and standard deviations of the methods' MSE calculated by conducting 30 independent experiments each. To check the statistical significance of our findings, we used the One-Way ANOVA test, followed by Tukey's HSD post-hoc analysis. We observe that with $5, 6$, and 7 clusters, LatentIM gives a lower prediction error that is statistically significant with $p < 0.001$.

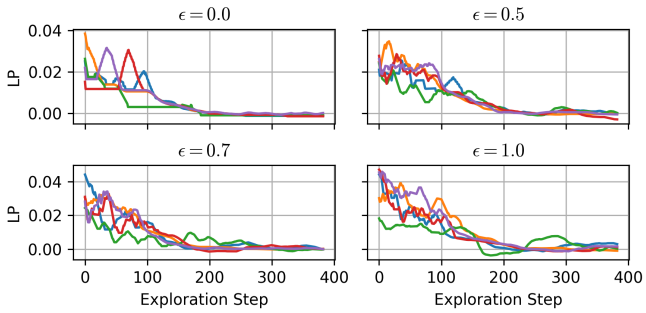

Fig. 8. Learning progress plots of the regions of LatentIM with different ϵ parameters. The plots are smoothed with time windows ($\alpha = 16$).

Inverse Model Learning

We also examined the performance of our method in inverse model learning where an inverse model ($[I_{enc}, E] \mapsto A$) was used instead of a forward model ($[I_{enc}, A] \mapsto E$). In order to analyze the performance of our intrinsic motivation strategies, we used the prediction error of the inverse model to measure the learning progress for the IM-based active exploration.

Fig. 9 shows the average weighted MSE of the 30 repeated runs for each model. We see that the performances of LatentIM and ActionIM are significantly better than the other three methods, where LatentIM ($\mu = 0.014418$) and ActionIM ($\mu = 0.015498$) are almost tied with each other. The performances of OutcomeIM and ObjectIM are worse because the regions that they form do not differentiate the end-effector configurations *(open, half-open, and close)*. In contrast, the regions of LatentIM and ActionIM are discriminated by the end-effector configurations, i.e., each region consisting of only one end-effector configuration. The differentiation based on end-effector configurations is crucial for the inverse model performance; for example, using the close end-effector configuration for lifting the object can not produce the desired outcome. In conclusion, LatentIM performed better than the other variants on both forward and inverse model learning.

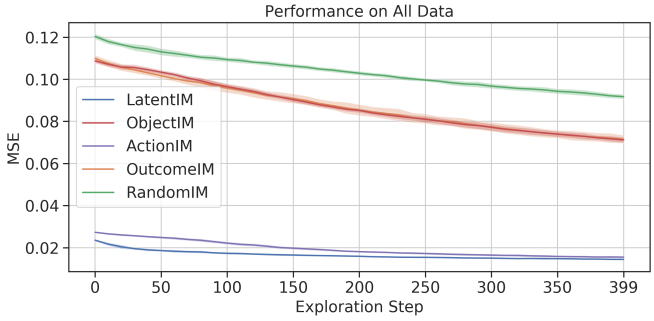

Fig. 9. Comparison of the inverse models' prediction performances of LatentIM, OutcomeIM, ObjectIM, ActionIM, and RandomIM. The plot shows the change in the average MSE during exploration from 30 independent runs. The shaded areas show the standard deviation.

TABLE II
MEAN MSE VALUES FOR LATENTIM, EFFECTIM AND RANDOMIM WITH DIFFERENT NUMBER OF CLUSTERS.

# of Clusters	LatentIM		OutcomeIM		RandomIM		$\mu(L)-\mu(E)$	$\mu(L)-\mu(R)$	$\mu(E)-\mu(R)$
	μ	σ	μ	σ	μ	σ			
2	0.009876	0.000293	**0.009090**	0.000232	0.010308	0.000302	0.00079^{***}	-0.00043^{***}	-0.00122^{***}
3	0.009714	0.000354	**0.008502**	0.000231	0.011000	0.000210	0.00121^{***}	-0.00129^{***}	-0.00250^{***}
4	0.008451	0.000352	0.008391	0.000203	0.011424	0.000258	0.00006^{ns}	-0.00297^{***}	-0.00303^{***}
5	**0.006836**	0.000194	0.009195	0.000255	0.011787	0.000199	-0.00236^{***}	-0.00495^{***}	-0.00259^{***}
6	**0.008892**	0.000260	0.010928	0.000264	0.012116	0.000186	-0.00204^{***}	-0.00322^{***}	-0.00119^{***}
7	**0.007666**	0.000291	0.010956	0.000231	0.012588	0.000186	-0.00329^{***}	-0.00492^{***}	-0.00163^{***}

Different Implementations of the Components

To evaluate the effectiveness of the model components, we analyzed prediction errors and LatentIM regions with different implementations of the components than those used in the proposed method. We conducted controlled experiments where the model components were replaced with alternative ones that have different capabilities. Table III presents the prediction errors of each approach with the given combination of implementations. The first row shows the prediction errors (40 experiments) of the proposed method presented in the main text. The underlined components in the following rows indicate the replacements. Each experiment consisted of 10 independent runs.

For the PCA case (Table III, 2^{nd} row), we used three principal components to reduce the dimensionality of the concatenation of the object, action, and outcome vector. For the Autoencoder (3^{rd} row), again, we used a three dimensional latent space. The encoder had an 18 dimensional input layer followed by dense layers with 32, 16, 8, 4 hidden units and a tanh activation function. The decoder of the AE was symmetric to the encoder. It was observed that with Autoencoder and PCA methods, the regions of LatentIM were similar to the regions presented in the main text. Thus, the prediction performance did not change significantly.

For the forward models, we used linear activation function (Table III, 4^{th} row) instead of ReLU. We observed that using linear activation made the MSE values for all the approaches higher, i.e., prediction performances of all the approaches became more inadequate compared to the ReLU case. Thus, we conclude that, with our experiment setup, the forward models need to approximate non-linear relationships.

As a clustering method, K-Means (Table III, 5^{th} row) was used instead of GMM. Following the proposed method, we set the number of clusters as five. The regions of LatentIM, as well as the prediction errors, were quite similar to the GMM case. Furthermore, we also analyzed the regions of other approaches and found only slight changes in the percentages of the object types in each region. Thus, their performances also did not change significantly.

TABLE III
MSE AND STANDARD DEVIATION VALUES FOR LATENTIM,OUTCOMEIM, RANDOMIM, OBJECTIM, ACTIONIM WITH DIFFERENT IMPLEMENTATIONS OF THE COMPONENTS.

Dim. Reduc.	Clustering	Forward Model	Approach	MSE	STD
VAE	GMM	NN ReLU 512 Units	**LatentIM**	**0.0068**	0.000237
			OutcomeIM	0.0091	0.000234
			RandomIM	0.0118	0.000222
			ObjectIM	0.0089	0.000230
			ActionIM	0.0107	0.000248
PCA	GMM	NN ReLU 512 Units	**LatentIM**	**0.0083**	0.000321
			OutcomeIM	0.0091	0.000197
			RandomIM	0.0117	0.000125
			ObjectIM	0.0090	0.000245
			ActionIM	0.0108	0.000201
AE	GMM	NN ReLU 512 Units	**LatentIM**	**0.0078**	0.000212
			OutcomeIM	0.0090	0.000249
			RandomIM	0.0119	0.000184
			ObjectIM	0.0089	0.000272
			ActionIM	0.0108	0.000151
VAE	GMM	NN Linear 512 Units	LatentIM	0.0284	0.000225
			OutcomeIM	0.0213	0.000114
			RandomIM	0.0232	0.000245
			ObjectIM	0.0298	0.000178
			ActionIM	0.0259	0.000201
VAE	K-Means	NN ReLU 512 Units	**LatentIM**	**0.0066**	0.000213
			OutcomeIM	0.0094	0.000262
			RandomIM	0.0119	0.000382
			ObjectIM	0.0084	0.000326
			ActionIM	0.0106	0.000235