Intrinsic Motivation in Object-Action-Outcome Blending Latent Space

Melisa Idil Sener\textsuperscript{1}, Yukie Nagai\textsuperscript{2}, Erhan Oztop\textsuperscript{3,4} and Emre Ugur\textsuperscript{1}

\textsuperscript{1}Bogazici University, Istanbul, Turkey. \textsuperscript{2}International Research Center for Neurointelligence, The University of Tokyo, Tokyo, Japan. \textsuperscript{3}Osaka University, Osaka, Japan. \textsuperscript{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 adapt 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 robot interacts with different kinds of objects using a set of parameterized actions and learns the outcomes of these interactions. With the proposed approach, the robot organizes its own curriculum of learning as in existing intrinsic motivation approaches and outperforms them in terms of learning speed. Moreover, the learning regime demonstrates features that partially match infant development, in particular, the proposed system learns to predict grasp action outcomes earlier than that of push action.

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]. According to Deci [5], [6], exploration, novelty-seeking behaviors, and play stem from the human need for feeling competent and self-determining, these behaviors are “intrinsically motivated”.

Intrinsically motivated strategies have been used along with various types of robot learning methods and applications such as socially guided learning [7]–[9], affordances [10]–[12], and planning [13]. Given the exploration space of the agent, a particular intrinsic motivation (IM) signal, learning progress [14], [15], 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 [16].

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 [17]. 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 to form predictive models that it can use for better control and decision making. Thus, the challenge is to efficiently and effectively deal with high-dimensional object-action-outcome exploration space of the manipulator robot using IM in learning predictive models.

How animals and humans address this challenge inspires our approach as well as several other computational approaches [17]–[19]. Kawato [20] argues the formation of internal models in humans and animals for learning to control different objects and/or body parts in different dynamics. 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 a desired sensorimotor granularity. In particular, a forward model predicts the sensory outcome of a given motor input, and an inverse model estimates the required motor command to achieve a desired sensory outcome. For motor control, Wolpert and Kawato [19] later 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 world considering the qualitatively different contexts that can be encountered, 2) simultaneous learning of different contexts without interference with each other, and 3) the possibility of learning a more complex context by reusing the knowledge captured in 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 (object, action, outcome) triplets, 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 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 different 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 that utilize only object \[21\], action \[10\], or outcome/goal \[22\] spaces in terms of learning speed, and
- the developmental progress obtained by the proposed approach partially parallels the general nature 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 \[23\].

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 II and IV. Section V demonstrates 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 as an essential problem in robot learning \[24\]. Two different disciplines, reinforcement learning \[25\], \[26\] and developmental robotics \[14\], \[27\]–\[29\], studied intrinsic motivation to address the exploration problem in order to facilitate open-ended and cumulative learning in robotic systems \[24\].

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 \[26\], \[30\], \[31\], whereas others studied how to combine intrinsic and extrinsic rewards in RL settings \[25\], \[32\], \[33\]. Intrinsic motivation was also applied to RL at the different levels of the hierarchies \[34\]–\[36\]. 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. \[14\], 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 \[37\], similar to \[14\], we partitioned the sensorimotor space by considering a single parameter at each partitioning step in order to form exploration regions. Forestier et al. \[38\] developed an algorithmic procedure called “intrinsically motivated goal exploration processes” (IMGEP) that allow 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. \[39\] hypothesized that an agent learns the dynamics of its body by autonomous goal generation regulated by the intrinsic motivations. 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. \[40\] 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 as a result of 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 \[41\], \[42\], planning \[43\], \[44\], control \[44\]–\[46\], and RL \[26\], \[32\], \[33\], \[35\], \[47\]–\[50\] focus on learning representations to foster autonomy. Among these, a number of studies utilized representation learning in IM-based exploration \[26\], \[48\], \[51\]–\[54\]. Bugur et al. \[55\] 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. \[51\] integrated a
representation learning stage on top of IMGEP [38] to create the goal-spaces by encoding raw sensory observations. In that study, the agent first passively observed 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. [32] 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 [35] considered only action and effect space, whereas 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 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)), \( I_{enc} \). 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 semicircular 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 [56]. 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 [56]. The VAE is trained with Adam [57] optimizer with a batch size of 100. The latent space is formed by using the \( \mu(z) \) from the encoder’s output.

**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 empirically chosen number of clusters (Fig. 1(C)), where each cluster corresponds to a “region” \( R_i \) that the robot can build local forward models 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
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.

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 \( \kappa \) is sampled from the FM’s responsibility region. In order to avoid catastrophic forgetting \([58], [59]\), the FM is continued training with all the data it encountered so far. To avoid overfitting, at each step, the FM is trained for only a small number (5) of epochs.

**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 \( \epsilon \)-greedy \([61]\) selection mechanism, and (object, action) pairs are sampled corresponding to the selected region for interaction.

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

\[
P_{n}(t+1) = \gamma_{n}(t+1) - \gamma_{n}(t+1 - \theta),
\]

where \( \gamma_{n} \) indicates the mean error of FM\( n \) at update cycle \( t \), and is calculated as follows:

\[
\gamma_{n}(t+1) = \frac{\sum_{i=0}^{\theta} e_{n}(t+1-i)}{\theta + 1}
\]

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 \( E_{obs} \), and the window parameter \( \theta \) is empirically set to 16. The window parameter \( \theta \) 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.

**IV. Experiment Setup**

The experiment setup was simulated in CoppeliaSim \([62]\), where a six-degrees-of-freedom robot arm with a gripper
(UR10) was chosen as the manipulator to be used in the experiments. In the setup created, the robot could interact with three types of objects through its reach and enclose action. The details of the objects used, the action parameters, and the outcome is 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 \(128 \times 128\) top-down depth image 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 \(r_{\text{path}} = [26, 31]\) cm and a z-orientation within \(\phi_{\text{path}} = [0,2\pi]\) radians. \(\phi_{\text{path}}\) 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 to 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 \((r_{\text{path}}, \phi_{\text{path}}, \text{closed}, \text{half open}, \text{open})\) where the last three parameters are binary and represents the one-hot encoding of the gripper state.

**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, and 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 \(I_{\text{enc}}\), 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 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 interactions, and the selected ones by IM are continued to be trained with the past interactions plus newly sampled \(k = 16\) 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 analyze the results of our latent space based IM approach (LatentIM), and compare it with the alternatives that use only object (ObjectIM), action (ActionIM) and outcome (OutcomeIM) spaces in region partitioning with the same number of regions. 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 developmental order of prediction capabilities? (Section V-C)
3) What is the effect of the different hyper-parameters (number of clusters and ϵ) on overall performance? (Appendix)

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 change in z position; region 4 includes open gripper and objects with change in z position; region 5 includes closed. Considering these characteristics, we name 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 as non-lifting (n_close). Note that these labels are given in order to help the reader, and the system does not use any given labels. Sample snapshots from interactions of (l_pinch), (n_pinch), and (n_close) regions are provided in Fig. 3

B. Comparison of Overall Performances

To investigate the effect of the region formation on the overall performance of the system, we analyze five different models, namely LatentIM, ObjectIM, ActionIM, OutcomeIM, and RandomIM. Fig. 5(a) shows the average weighted MSE of the 40 repeated runs for each model. 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, the OutcomeIM and ObjectIM perform similarly; the only difference for those two is that the OutcomeIM is better at the beginning, but ObjectIM shows a more rapid decrease than the MSE. It seems that among all the methods, the 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 the 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 l_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 consider OutcomeIM as the competitor of our method and RandomIM as the baseline.

In Fig. 5(b), we present the statistical analysis of the differences between LatentIM, OutcomeIM, and RandomIM taken from the different exploration steps. We ran analysis of variance (ANOVA) to check whether the MSE distributions of these three approaches are different, then carried 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. Developmental Order of Skill Prediction

In this subsection, we analyzed the developmental order of regions and skill prediction that is regulated by the IM module. The analysis of the developmental 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 the regions of LatentIM 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.
the strategies, the LP values reach to 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 $l_{\text{pinch}}$, $l_{\text{power}}$, $n_{\text{pinch}}$, $n_{\text{close}}$ and $n_{\text{power}}$ in average. This ordering is reasonable because, when grasped, the orientation change is $\approx 0$. 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.

To evaluate the significance of the ordering presented in Fig. 7, A Kruskal-Wallis test was performed on the learning progresses of the five different regions. The differences between the learning progress distributions of the regions taken from the interval $t = [0, 139]$ were 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 $l_{\text{pinch}}$ has the maximum learning progress value. Taken from that interval at Point A, the LP of $l_{\text{pinch}}$ was significantly greater than of $l_{\text{power}}$, $p < 0.05$ and the LP of $l_{\text{power}}$ was significantly greater than the rest with $p < 0.001$. At the same time-step, $n_{\text{pinch}}$ was not significantly different than $n_{\text{close}}$, while the LP of $n_{\text{close}}$ being significantly greater than of $n_{\text{power}}$, $p < 0.01$. Following the plot, we see the dominance of $l_{\text{power}}$ over $l_{\text{pinch}}$. Taken from that time window, at Point B, LP of $l_{\text{power}}$ was significantly greater than of $l_{\text{pinch}}$, $p < 0.05$. While the LP of $l_{\text{pinch}}$ is significantly greater than of $n_{\text{close}}$ and $n_{\text{pinch}}$ with $p < 0.001$, the difference between $n_{\text{close}}$ and $n_{\text{pinch}}$ was not significant, both being greater than $n_{\text{power}}$ with $p < 0.01$. After the decrease of $l_{\text{pinch}}$ and $l_{\text{power}}$, a significant increase in $n_{\text{pinch}}$ is visible at the Fig. 7. Being within that time window, at Point C, the LP of $n_{\text{pinch}}$ was significantly greater than of $n_{\text{close}}$, $p < 0.05$. While LP of $l_{\text{pinch}}$ is significantly less than of $n_{\text{close}}$ with $p < 0.05$ it was significantly greater than of $n_{\text{power}}$, $p < 0.001$. And finally, there is a short primacy of $n_{\text{close}}$ over the rest, Point D, the LP of $n_{\text{close}}$ was significantly greater than of $n_{\text{pinch}}$, $p < 0.05$.

VI. DISCUSSION AND CONCLUSION

The results of our experiments suggest that the key ingredients of 1) object-action-outcome blended latent space formation and 2) learning progress prioritization of local learning over the blended space yields a curriculum of sensorimotor learning that parallels important features of infant sensorimotor development. 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.

Developmental order of prediction skills. The proposed system developed prediction ability for basic grasp actions and sampling inside the regions. Detailed investigation and statistical analysis of the developmental order formed by LatentIM will be discussed later in this subsection.

In Fig. 6(b), OutcomeIM gives priority to $R_2$, which corresponds 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 developmental order (Fig. 6(c)).

Note that through the end of the exploration phase, for all
before the prediction skill for purposeful push actions, which replicates the basic pattern in infant development [23]. This was an emergent feature realized through the coupling of learning progress based intrinsic motivation with the object-action-outcome blended space. However, the order within the grasp action types could not be replicated. Since the precision pinch grasp requires finer control and precise movements, in infants, it emerges later compared to power grasp [2, 23]. 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 [64]. On the contrary, the robotic gripper is more suitable for a precision pinch grasp. 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.

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 identity [65], [66].

Superiority of the blended latent space based IM over others. From the onset, it is not clear whether using an object-action-outcome blended latent space to define the local models and apply IM in selecting regions formed within this latent space would yield better predictors. Yet, our experiments showed that latent space based IM significantly outperformed IM when using spaces based on individual object, action, or outcome in terms of learning speed and prediction accuracy. Furthermore, the emergence of regions and development order that we discussed above has only been observed clearly with latent space based IM.

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. However, these particular methods are not central in our framework, and VAE, GMM, and FFNN may be replaced by other dimensionality reduction, clustering, and regression methods, respectively, if required.

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 [67] 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. 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.

APPENDIX

In this section, we provide additional experiment results to examine the effect of the hyperparameters, i.e., $\epsilon$, and the number of clusters.

| $\epsilon$ | $\alpha$(LatentIM) | $\alpha$(OutcomeIM) | $\alpha$(RandomIM) |
|-----------|-------------------|-------------------|-------------------|
| 0.0        | 0.007121          | 0.009238          | 0.011895          |
| 0.2        | 0.006153          | 0.009065          | 0.011748          |
| 1.0        | 0.006819          | 0.008034          | 0.011766          |

Effect of $\epsilon$ Parameter

As explained in III-D, the $\epsilon$ parameter controls the ratio of exploration steps with random exploration to the ones with active exploration. We conduct $N = 30$ experiments for each $\epsilon$ value and verify 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 yield 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 $\epsilon$ parameter. For all the $\epsilon$ values, we observe that the LatentIM gives the lowest error among the other two. We 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 $\epsilon$ conditions. Increasing values of $\epsilon$ 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 number
of clusters. For this experiment, we use the given hyperparameters except for the number of clusters, which is 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 use the One-Way ANOVA test followed by Tukey’s HSD post-hoc analysis. We observe with 5, 6 and 7 clusters, LatentIM gives lower prediction error that is statistically significant with \( p < 0.001 \).

**ACKNOWLEDGMENT**

This research was partially supported by JST CREST “Cognitive Mirroring” [grant number: JPMJCR16E2] and TBTAK (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.

**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. 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.
8. Nicolas Duminy, Sao Mai Nguyen, and Dominique Duhaut. Learning a set of interrelated tasks by using a successation of motor policies for a socially guided intrinsically motivated learner. *Frontiers in neurorobotics*, 12:87, 2019.
9. 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.
10. Enure 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.
11. 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.
12. 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.
13. 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.
14. Pierre-Yves Oudeyer, Frédic Kaplan, and Verena V Hafner. Intrinsically motivated systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.
15. Jürgen Schmidhuber. Curious model-building control systems. In *Proc. international joint conference on neural networks*, pages 1458–1463, 1991.
16. 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.
17. Yukie Nagai. Predictive learning: its key role in early cognitive development. *Philosophical Transactions of the Royal Society B*, 374(1771):20180030, 2019.
18. 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.
19. Daniel M Wolpert and Mitsuo Kawato. Multiple paired forward and inverse models for motor control. *Neural networks*, 11(7-8):1317–1329, 1998.
20. Mitsuo Kawato. Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727, 1999.
21. Enure 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.
22. 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.
23. Rebecca J Scharf, Graham J Scharf, and Amanmari Stroutnup. Developmental milestones. *Pediatrics in review*, 37(1):25, 2016.
24. Manuel Lopes and Pierre-Yves Oudeyer. Guest editorial active learning and intrinsically motivated exploration in robots: Advances and
