Intrinsically Motivated Exploration for Automated Discovery of Patterns in Morphogenetic Systems

Anonymous Author(s)
Affiliation
Address
email

Abstract

Exploration is a corner stone of machine learning algorithms used to discover novel strategies or patterns. Intrinsically motivated goal exploration processes (IMGEPs) were shown to enable autonomous agents to efficiently explore and map the diversity of the effects they can produce on their environment. With IMGEPs, agents self-define their own experiments by imagining goals, then try to achieve them by leveraging their past discoveries. Progressively they learn which goals are achievable and which are not. IMGEPs were shown to enable efficient discovery and learning of diverse repertoires of skills in high-dimensional robots. In this article, we show that the IMGEP framework can also be used in an entirely different application area: automated discovery of self-organized patterns in complex morphogenetic systems. We also introduce a new IMGEP algorithm where goal representations are learned online and incrementally (past approaches used precollected training data with batch learning). For experimentation, we use Lenia, a continuous game-of-life cellular automaton. We study how IMGEPs algorithms enable to discover a variety of complex self-organized visual patterns. We compare random search and goal exploration methods with hand-defined, pretrained and online learned goal spaces. The results show that goal exploration methods identify more diverse patterns compared to random exploration. Moreover, the online learned goal spaces allow to successfully discover interesting patterns similar to the ones manually identified by human experts. Our results exemplify the ability of IMGEPs to aid the discovery of novel structures and patterns in complex systems. We are optimistic that their application will aid the understanding and discovery of new knowledge in various domains of science and engineering.

1 Introduction

Exploration is the process of searching new solutions for a problem or new information about a system. It is a corner stone of many machine learning algorithms. For example, a robotic reinforcement learning agent may have to explore for discovering new objects and effects it can produce on them. Moreover, exploration is an important part of knowledge discovery in science and engineering. In order to understand or optimize a system, one must explore to discover what are its potential behaviours and how to represent them.

Intrinsically motivated goal exploration processes (IMGEPs) have shown to be efficient exploration strategies for autonomous agents to discover and map the diversity of effects they can produce in their environment [3, 12, 26]. With IMGEPs, agents self-define their own experiments by imagining goals, then try to achieve them by leveraging their past discoveries (Fig. 1). Progressively they learn which goals are achievable and which are are not. The goals are defined in a space of representations

Code and additional videos at `https://intrinsically-motivated-discovery.github.io/`

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

which describe the important features of the raw observation space. For a robot that interacts with objects the locations and properties of those objects could be such features [12]. With deep neural networks, the goal representations can be directly learned from raw pixel perception by training the latent layers of autoencoders with pre-collected data [29, 26].

So far, IMGEPs have been mainly used in the context of autonomous learning agents/robots. They enabled an efficient exploration of diverse skill repertoires in high-dimensional robots [3, 12]. Nonetheless, their exploration capabilities are not constrained to this field and can be used in a variety of application scenarios. In this paper we exemplify their application for automating the discovery of complex behaviours and patterns of high-dimensional complex systems such as studied in developmental (theoretical) biology, chemistry or physics. Based on our results, IMGEPs show a high potential to be efficient tools for helping scientists to discover and analyze novel high-dimensional self-organized structures in these complex systems. In a recent step in that direction, Grizou et al. [17] showed that IMGEPs are capable of making autonomously discoveries in a chemical system based on a simple low-dimensional hand engineered goal space . In this paper we show that the full abilities of IMGEPs can be utilized for such environments by also learning in an unsupervised manner the representations that define the goal space. We show this ability on the example of discovering morphogenetic patterns in Lenia [6], a continuous game-of-life cellular automaton.

Moreover, we introduce with this paper a new IMGEP algorithm able to learn online the representations for its goal space. In previous IMGEP approaches, the goal representation either had to be defined by hand [12] or learned by a batch process with prerecorded data [29]. This required either expert knowledge about the system or existing data that might also bias the exploration. The online learning approach can be directly applied without expert knowledge or preexisting data.

In summary, the paper provides the following new contributions:

- The application of IMGEPs in a new domain: The discovery of structures and patterns in high-dimensional complex systems with autonomous learning of goal representations.
- A novel online representation learning algorithm for IMGEPs that does not rely on expert knowledge or precollected data.

2 Related Work

Inspired from the way human children can self-develop a hierarchy of skills in order to make sense of the world, *intrinsically-motivated* learning [2, 3] is a family of computational models that autonomously organize an agent exploration curriculum in order to discover efficiently a maximally diverse set of outcomes that can be produced in an unknown environment. Intrinsically Motivated Goal Exploration Processes (IMGEPs) [3, 12] is a family of curiosity-driven algorithm which have been developed in the context of high-dimensional complex real world systems. Population-based versions of these algorithms, leveraging episodic memory and hindsight learning, have shown to enable robots or artificial agents to acquire diverse repertoires of skills [35, 12] as to bootstrap the exploration capacity for deep reinforcement learning problems with rare or deceptive rewards [8]. Recent work [26, 29] studies how to automatically learn the goal representations with the use deep variational autoencoders. However, the training is done in an early stage and passively on a precollected set of available images. A related family of algorithms in evolutionary computation is novelty search [27] and quality-diversity algorithms [30], which can be formalized as special kind of population-based IMGEPs with a fixed random goal sampling policy.

Intrinsically motivated learning techniques have also been widely developed to handle exploration in reinforcement learning, with diverse approaches ranging from estimating visitation counts [4], measures of empowerment [16], to goal exploration approaches [11] with hindsight learning [1] and automated curriculum [9], or related concepts such as auxiliary tasks [34] and general value functions [38]. However, these approaches focused on the problem of sequential decisions in MDPs (incurring a cost on sample efficiency), orthogonal to the automated discovery framework considered here with independent experiments (allowing the use of memory-based sample efficient methods).

Active inquiry-based learning strategies have been used in biology [23, 22], chemistry [10] and astrophysics [33] to autonomously query which set of experiments to perform in order to improve the overall model of the system. These data-driven approaches considerably reduce the experimental costs but still require a database background of representative experiments. Recently, machine learning

algorithms [31, 32, 36] have been integrated into the experimental laboratory and often combined to the use of robotics and automation platforms [15, 20]. These methods open a brand new perspective to the way scientific experiments are conducted, but most of them rely on *expert* knowledge and optimize specific target properties. Rather than trying to find the optimal physico-chemical model from a database of collected experiments, we are interested to automatically discover a diversity of unseen patterns without requiring prior knowledge of the system.

Representation learning aims at finding a set of low-dimensional explanatory factors representing high-dimensional input data [5] and is a key problem in a lot of areas in order to understand the underlying structure of complex observations. Deep variational *autoencoders* (VAE) [24] are one of the most popular approaches and many state-of-the-art methods [41, 18, 25, 19, 21, 7] build on top of it using varying objectives and network architectures. See Tschannen et al. [39] for an in-depth review.

3 Method

This section describes intrinsically motivated goal exploration and the online learning approach for representation of their goal spaces.

3.1 Intrinsically Motivated Goal Exploration Processes

An IMGEP (Fig. 1) is a sequence of experiments that explore the parameters of a system by targeting self-generated goals. It aims to maximize the diversity of observations from that system within a budget of N experiments.

The systems are defined by three components. A parameterization space Θ corresponding to their controllable parameters θ. An observation space O where an observation o is a vector representing all the signals captured from the system. For this paper, the observations are a time series of images which depict the morphogenesis of activity patterns. Finally, an unknown environment dynamic D: $\theta \rightarrow O$ which maps parameters to observations.

To explore a system, an IMGEP defines a goal space $\mathcal{T}$ that represents relevant features of its observations. For a robot that has to manipulate objects and observes them with a video camera, those features could be the object positions. From this goal space a goal $g \sim G(\mathcal{H})$ is sampled by a goal sampling distribution. For the robot example this would correspond to a sampling of positions to where the robot should move the objects. Then, a parameter θ is chosen by a parameter sampling policy $\Pi = \Pr(\theta; g, \mathcal{H})$ that should be explored in order to reach the goal g. Usually the parameter sampling policy and in some cases the goal sampling distribution take into account previous explorations which are stored in a history $\mathcal{H}$. After a parameter is selected it is explored on the system and the outcome o observed. Based on the observation the actually reached goal $\mathcal{T}(o)$ is computed and together with its corresponding parameter and observation stored in history $\mathcal{H}$. The exploration process is repeated until a certain number of steps N or another constraint is reached. Because the sampling of goals and parameters depend on a history of explored parameters, an initial set of N_{init} parameters are randomly sampled and explored before the goal exploration

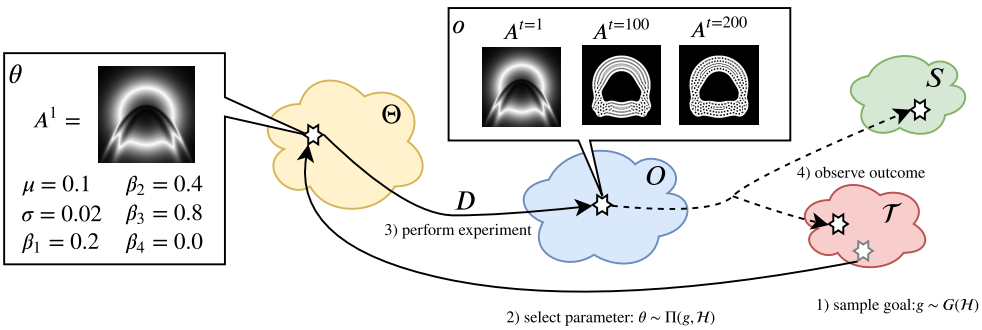

Figure 1: The intrinsically motivated goal exploration process (IMGEP) on the example of Lenia.

process starts. Optional statistics S about the collected observations and can be used to analyze and visualize exploration results, however they are unknown to the discovery algorithm.

Depending on the algorithmic instantiation of this architecture, different goal and parameter sampling mechanisms can be used [3, 13]. For this paper we chose for both simple approaches. Goals are sampled from a uniform distribution over the goal space. Parameters are chosen by selecting for a given goal the parameter from the history whose reached goal has the shortest distance in the goal space to the given goal. This parameter is then mutated to by a random process.

3.2 Learning of Goal Spaces via Online Representation Learning

For IMGEPs the definition of the goal space $\mathcal{T}$ is a critical part, because it defines which behaviors and observations the process tries to find in the target system. A straightforward choice to define a goal space is by selecting features manually, such as by using computer vision algorithms to detect the positions of objects from the recorded video images [12, 17]. The problem of this approach is that expert knowledge is required to select helpful features. Moreover, for unknown and high-dimensional systems even experts might not know which features to use.

A more elaborated approach is to learn goal space features with unsupervised representation learning. Representation learning is able to learn a mapping $\tilde{R_{\mathcal{T}}}(o)$ from the raw sensor observations o to a compact latent vector $\mathbf{z} \in \mathbb{R}^d$. This latent mapping can then be used as a goal space where a latent vector $\mathbf{z} = g$ is interpreted as a goal.

Previous approaches applied already successfully the learning of goal spaces with variational autoencoders (VAE) [26, 29]. However, the goal spaces were learned before the start of the exploration and from a prerecorded dataset of observations from the target system. During the exploration the learned representations were kept fixed. A problem with this pretraining approach is that training samples may be limited and often biased towards the initial knowledge of the system.

In this paper we attempt to address this problem by continuously adapting the learned representation to the different observations of the system encountered during the exploration process. We believe it is crucial to capture the factors of variations of patterns unseen so far by the network to further enable the discovery of a diversity of similar patterns in the system. To address this challenge, we propose an online goal space learning IMGEP (IMGEP-OGL), which can learn the goal space in an incremental manner during the exploration process (Algorithm 1). We are using different variants of VAEs for the representation learning part of the algorithm. For details about the different variants please refer to Section in the Supplementary Materials.

Algorithm 1: IMGEP-OGL

Initialize goal space representation $R_{\mathcal{T}}$ with random weights
Initialize an empty training dataset $D_{\mathcal{T}} = \emptyset$
for $i \leftarrow 1$ **to** N **do**
    if $i < N_{init}$ **then** `// Initial random iterations to populate` $\mathcal{H}$
        Sample $\theta \sim \mathcal{U}(\Theta)$
    else `// Intrinsically motivated iterations`
        Sample a goal $g \sim \mathcal{G}(\mathcal{H})$ based on space represented by $R_{\mathcal{T}}$
        Choose $\theta \sim \Pi(g, \mathcal{H})$
    Perform an experiment with θ and observe o
    Append $(\mathcal{T}(o), \theta, o)$ to the history $\mathcal{H}$
    Append o to $D_{\mathcal{T}}$
    if $i \mod K == 0$ **then** `// Periodically train the network`
        for *E epochs* **do**
            Train $R_{\mathcal{T}}$ on $D_{\mathcal{T}}$ with importance sampling
        for $(g, \theta, o) \in \mathcal{H}$ **do** `// Update the database of reached goals`
            $\mathcal{H}[g] \leftarrow R_{\mathcal{T}}(o)$

Their training procedure is integrated in the goal sampling exploration process by first initializing the VAE network with random weights. After the exploration of each new parameter the resulting observation is added to a dataset of training examples $D_{\mathcal{T}}$. The VAE network is then trained every K explorations for E epochs on the collected training data. Importance sampling is used to give more weight to recently discovered patterns.

4 Experiments

The usefulness of IMGEPs for the discovery of novel patterns in complex system was evaluated on the Lenia system. The following sections introduce Lenia, the different algorithms that were compared and the experimental procedure.

4.1 Target System: Lenia

Lenia [6] is a continuous cellular automaton [40] similar to Conway's Game of Life [14]. Game-of-life systems have been used many times as abstract models for theoretical understanding of how self-organized structures may form in natural morphogenetic systems. Lenia, in particular, represents a high-dimensional complex dynamical system where diverse visual structures can self-organize, including solitons, and yet are hard to find by manual exploration. It is therefore well suited to test the performance of exploration algorithms for unknown and complex systems.

Lenia consists of a two-dimensional grid of cells $A \in \mathbb{R}^{256\times256}$ where the state of each cell is a real-valued scalar activity $A_{x,y} \in [0, 1]$. The state of cells evolves over discrete time steps (Fig. 2, a). The activity change of a cell between time steps is computed by integrating the activity of neighbouring cells. Several parameters $\{R, T, \mu, \sigma, \beta_1, \ldots, \beta_4\}$ control the process.

Lenia can be understood as a morphogenetic system where the parameters θ and the initial state A^1 represent the genes of a developmental process. They control into which final activity pattern the initial pattern morphs (Fig. 2). Lenia can produce diverse patterns with different dynamics such as stable, non-stable or chaotic patterns. Most interesting, patterns that resemble microscopic animals can be produced [6]. We use Lenia to study if IMGEPs can autonomously discover such patterns.

We implemented different pattern classifiers to analyze the exploration results. We differentiate between dead and alive patterns. A pattern is dead if the activity of all cells are either 0 or 1. Alive patterns are classified into animals and non-animals. Animals are a connected areas of positive activity which are finite, i.e. which do not infinitely cross several borders. All other patterns are non-animals which usually spread over the whole state space.

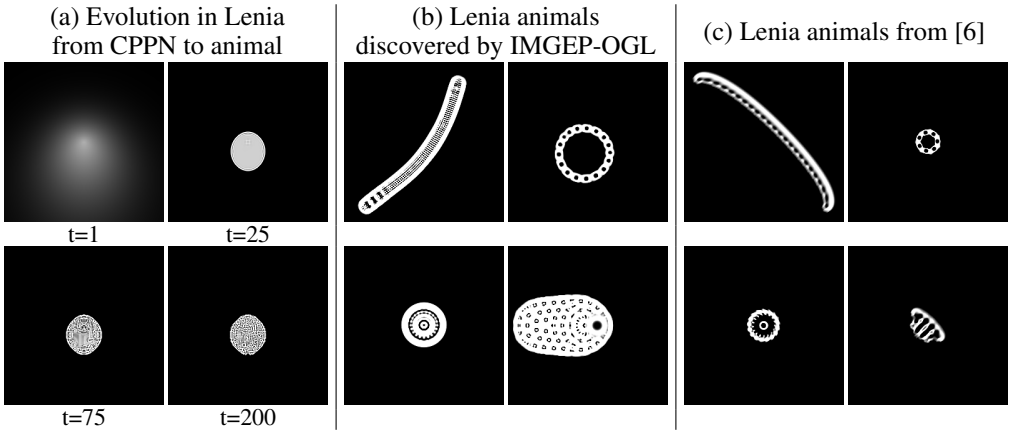

Figure 2: Example patterns produced by the Lenia system. Illustration of the dynamical morphing from an initial CPPN image to an animal (a). The automated discovery (b) is able to find similar complex animals as a human-expert manual search (c) by [6].

4.2 Compared Algorithms

The exploration behavior of different IMGEP algorithms were evaluated and compared to a random exploration. The IMGEP variants differ in their way how the goal space is defined or learned.

Random exploration: The IMGEP variants were compared to a random exploration that sampled randomly for each of the N exploration iterations the parameters θ and the initial state A^1.

IMGEP-HGS - Goal exploration with hand-defined goal space: The first class of IMGEP uses a hand-defined goal space that is constructed by 5 features that were measured about the final pattern that developed in Lenia for each exploration. We tested different combinations of summary statistics and selected the one that was overall best. Its goal space consists of: 1) the sum over the activity of all cells, 2) the density of the activated cells, 3) the velocity of the activity center, 4) a asymmetry measure of the entity and 5) a distribution measure of the entity.

IMGEP-PGL - Goal exploration with a pretrained learned goal space: For this IMGEP variant the goal space was learned with one of several VAE approaches on training data before the exploration process started. The training set consisted of 558 Lenia patterns. Half of the patterns were animals that have been manually identified by [6]. The other half were randomly initialized patterns that were created with the same procedure as described in Section 4.3.

IMGEP-OGL - Goal exploration with online learning of the goal space: The final algorithm is the new online variant for IMGEPs (Algorithm 1).

Please refer to the Supplementary Materials for details about the parameterization of the algorithms.

4.3 Experimental Procedure

For each algorithm 10 exploration experiments were conducted to measure their average performance. Each experiment consisted of $N = 5000$ exploration iterations. For the IMGEP variants the first $N_{\text{init}} = 1000$ iterations were random explorations to populate their histories $\mathcal{H}$. For all algorithms an identical initial set of random explorations was used to allow a better comparison between them. For the following 4000 iterations each IMGEP approach sampled a goal g by a uniform distribution over its goal space. Then, the parameter θ_k from a previous exploration in $\mathcal{H}$ was selected whose reached goal g_k had the minimum distance to the current goal g within the current goal space embedding. This parameter was then mutated by a random process to get the parameter θ that was explored.

The parameters consisted of a compositional pattern producing network (CPPN) [37] that generates the initial state A^1 for Lenia and the settings defining Lenia's kernel function and update rules: $\theta = [\text{CPPN} \rightarrow A^1, R, T, \mu, \sigma, \beta_1, \ldots, \beta_4]$. CPPNs are recurrent neural networks that were originally used to generate and evolve gray scale images, but that can be similarly used for the initial activity patterns of Lenia. The networks are randomly initialized and mutated by a random process that defines their structure and connection weights as done in [37]. The random initialization of the other Lenia settings was done by an uniform distribution and their mutation by a Gaussian distribution around the original values. The meta parameters to initialize and mutate parameters θ are equal for all algorithms (SM, Section 1.4). They were manually chosen without optimizing them for a specific algorithm. The parameters of the CPPN networks were set to initialize and mutate networks that generate similar images as in [37].

5 Results

We compared random explorations and different IMGEP algorithms on their ability to produce diverse Lenia patterns. Diversity is measured in a statistic space constructed by hand-defined features. Furthermore, we compared the goal spaces of hand-defined and learned IMGEP variants.

5.1 Diversity of Explored Lenia Patterns

Diversity is measured by the spread of the exploration in a statistic space S defined by 8 features. The features are: 1) sum over the activity of all cells, 2) number active cells, 3) density measure of the activated cells, 4) velocity of the activity center, 5) average movement angle of the activity center, 6) angular rotation of the activity center, 7) a asymmetry measure of the entity and 8) a distribution

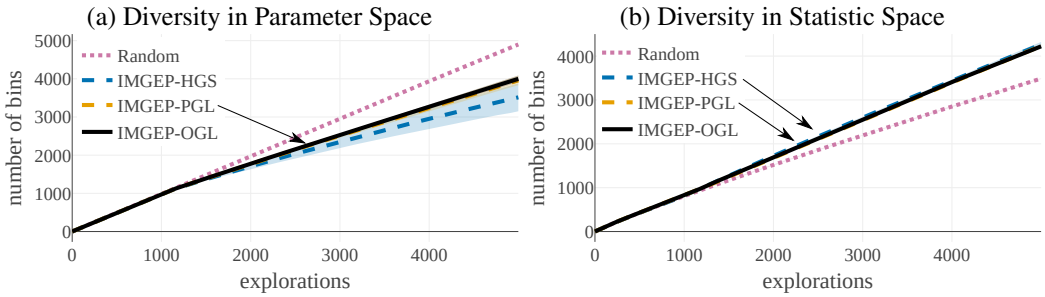

Figure 3: (a) Although random explorations reach the highest diversity in the space of parameters Θ, (b) IMGEPs reach a higher diversity in the statistic space. They discover a more diverse set of Lenia patterns. Depicted is the average diversity ($n = 10$) with the standard deviation as shaded area (for some not visible because it is too small).

measure of the entity. See the Supplementary Material for more details. Please note that the choice of this space was arbitrary and results might change depending on the used features.

For each experiment all explored patterns were projected into the statistic space. To measure the diversity of the found solutions we used a simple measure of the area that the exploration covered in the statistic space. The measure discretizes the statistic space into bins of equal size by splitting each dimension into 10 sections resulting in 10^8 bins. The number of bins in which at least one explored entity falls is counted and used as a measure for diversity.

We also measured the diversity in the space of parameters Θ. This measure excluded the high-dimensional initial state A^1 and used only the other parameters as dimensions: R, T, μ, σ, $b_1, \ldots, b_4$. Also for this space 10 bins per dimension where used to discretize the space.

Comparing the diversity between the parameter space and the statistic space reveals the advantage of IMGEPs over random explorations (Fig. 3). Although random explorations have reached the highest diversity in the space of parameters, they are outperformed in terms of diversity by the IMGEP approaches in the statistic space. Thus, the IMGEP approaches are better in the actual objective of our exploration, finding a diverse set of Lenia patterns.

All IMGEP variants reached a similar diversity measure if measured over all explored patterns (Fig. 3, b). This is not the case if the diversity within certain subgroups of patterns is considered. In the case of the diversity only over explored animals (Fig. 4, a) the new online approach IMGEP-OGL is finding the highest diversity of animals. It is closely followed by the pretrained IMGEP-PGL approach. The hand-defined goal space approach IMGEP-HGS can identify a diversity of 50% compared to IMGEP-OGL, and random explorations only less than 30%. In the case of diversity over non-animal patterns the IMGEP-HGS reached the highest diversity followed by the random

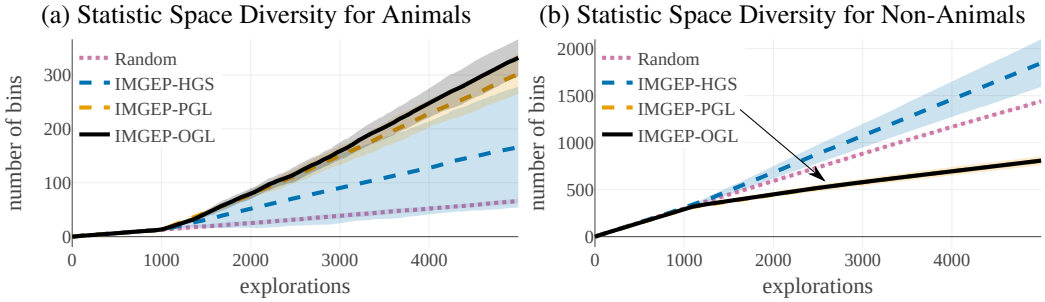

Figure 4: (a) IMGEPs with a learned goal space discovered a larger diversity of animals compared to the IMGEP with a hand-defined goal space. (b) For non-animals the hand-defined goal space discovered the highest diversity. Depicted is the average diversity ($n = 10$) with the standard deviation as shaded area (for some not visible because it is too small).

(a) IMGEP-HGS Goal Space (b) IMGEP-OGL Goal Space

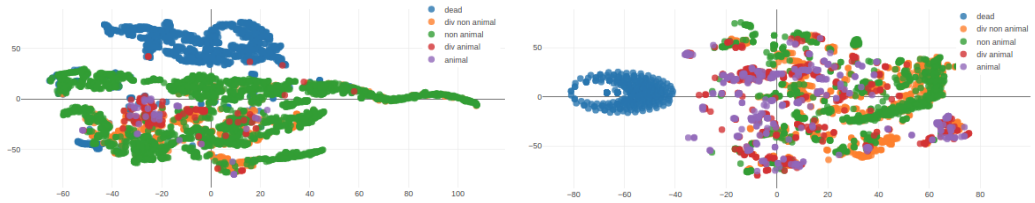

Figure 5: (a) Hand-defined and (b) learned goal space have major differences (shown here by a t-SNE visualization) which explain the difference in their resulting diversity of animals and non-animals (Fig. 4).

explorations (Fig. 4, b). The two approaches that learned the goal space (IMGEP-OGL, IMGEP-PGL) reached the lowest diversity for non-animal patterns.

5.2 Differences in Goal Spaces

We analyzed the goal spaces of the different IMGEP variants to understand their behavior by visualizing their reached goals in a two-dimensional space. t-SNE [28] was used to reduce the high-dimensional goal spaces. It puts points that were nearby in the high-dimensional space also close to each other in the two-dimensional visualization.

The goal spaces of IMGEP-HGS and IMGEP-OGL show strong differences between each other which we believe explain their different abilities to find either a high diversity of non-animals or animals (Fig. 4). The goal space of the IMGEP-HGS (Fig. 5, a) shows large areas and several clusters for non-animal patterns. Animals form only two small and nearby clusters. Thus, the hand-defined features seem poor to discriminate and describe animal patterns in Lenia. As a consequence, when goals are uniformly sampled within this goal space during the exploration process, then more goals are generated in regions that describe non-animals. This can explain why IMGEP-HGS explored a higher diversity of non-animal patterns.

In contrast, features learned by IMGEP-OGL capture better factors that differentiate animal patterns as can be observed by several clusters of animals that span a wide area in its goal space (Fig. 5, b). We attribute this effect to the difficulty of VAEs to capture sharp details [42]. They therefore can only represent the general form of the patterns. Non-animals usually span over the whole state space and differ in their fine-grained structures. VAEs have therefore difficulty to learn their important discriminators. Animals differ also often in their form and extension and can be easier differentiated. The VAE learned goal-spaces seem therefore better suited for exploration of diverse animal patterns.

6 Conclusion

We presented in this paper the application of intrinsically motivated exploration via IMGEPs towards a new and exciting application area: the discovery of patterns and structures in complex systems. All evaluated IMGEP variants were able to discover a diverse set of patterns for Lenia, a cellular automaton, by directly exploring its high-dimensional parameters and observing its high-dimensional output patterns. We could demonstrate that goal spaces for such systems can be successfully learned via deep VAEs which allow the identification of animal-like patterns similar to those identified by human experts (Fig. 2). Moreover, our new approach of learning goal spaces online via data collected during the exploration process could outperform a pretrained and fixed goal space in terms of identifying a diverse set of animal-like patterns.

We believe that IMGEPs are able to facilitate the study of similar complex and high-dimensional systems in different fields of engineering and science such as physics and chemistry. IMGEPs allow to explore unknown systems efficiently to discover their interesting behaviors and patterns which can help to understand the systems better or to find new solutions for problems in them.

References

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pages 5048–5058, 2017.

[2] Gianluca Baldassarre and Marco Mirolli. *Intrinsically motivated learning in natural and artificial systems*. Springer, 2013.

[3] 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.

[4] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016.

[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[6] Bert Wang-Chak Chan. Lenia-biology of artificial life. *arXiv preprint arXiv:1812.05433*, 2018.

[7] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018.

[8] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. Gep-pg: Decoupling exploration and exploitation in deep reinforcement learning algorithms. *arXiv preprint arXiv:1802.05054*, 2018.

[9] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. Curious: Intrinsically motivated multi-task, multi-goal reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2019.

[10] Vasilios Duros, Jonathan Grizou, Weimin Xuan, Zied Hosni, De-Liang Long, Haralampos N Miras, and Leroy Cronin. Human versus robots in the discovery and crystallization of gigantic polyoxometalates. *Angewandte Chemie*, 129(36):10955–10960, 2017.

[11] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. *arXiv preprint arXiv:1707.05300*, 2017.

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

[13] 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.

[14] Martin Gardener. Mathematical games: The fantastic combinations of john conway's new solitaire game" life,". *Scientific American*, 223:120–123, 1970.

[15] Jarosław M Granda, Liva Donina, Vincenza Dragone, De-Liang Long, and Leroy Cronin. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature*, 559(7714):377, 2018.

[16] Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

[17] Jonathan Grizou, Laurie J Points, Abhishek Sharma, and Leroy Cronin. Exploration of self-propelling droplets using a curiosity driven robotic assistant. *arXiv preprint arXiv:1904.12635*, 2019.

[18] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vazquez, and Aaron Courville. Pixelvae: A latent variable model for natural images. *arXiv preprint arXiv:1611.05013*, 2016.

[19] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, volume 3, 2017.

[20] Claudia Houben and Alexei A Lapkin. Automatic discovery and optimization of chemical processes. *Current opinion in chemical engineering*, 9:1–7, 2015.

[21] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[22] Ross D King, Jem Rowland, Stephen G Oliver, Michael Young, Wayne Aubrey, Emma Byrne, Maria Liakata, Magdalena Markham, Pinar Pir, Larisa N Soldatova, et al. The automation of science. *Science*, 324(5923):85–89, 2009.

[23] Ross D King, Kenneth E Whelan, Ffion M Jones, Philip GK Reiser, Christopher H Bryant, Stephen H Muggleton, Douglas B Kell, and Stephen G Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247, 2004.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[25] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

[26] Adrien Laversanne-Finot, Alexandre Pere, and Pierre-Yves Oudeyer. Curiosity driven exploration of learned disentangled goal spaces. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 487–504. PMLR, 29–31 Oct 2018.

[27] Joel Lehman and Kenneth O Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336, 2008.

[28] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[29] Alexandre Péré, Sébastien Forestier, Olivier Sigaud, and Pierre-Yves Oudeyer. Unsupervised Learning of Goal Spaces for Intrinsically Motivated Goal Exploration. In *ICLR2018 - 6th International Conference on Learning Representations*, Vancouver, Canada, April 2018.

[30] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.

[31] Paul Raccuglia, Katherine C Elbert, Philip DF Adler, Casey Falk, Malia B Wenny, Aurelio Mollo, Matthias Zeller, Sorelle A Friedler, Joshua Schrier, and Alexander J Norquist. Machine-learning-assisted materials discovery using failed experiments. *Nature*, 533(7601):73, 2016.

[32] Brandon J Reizman, Yi-Ming Wang, Stephen L Buchwald, and Klavs F Jensen. Suzuki–miyaura cross-coupling optimization enabled by automated feedback. *Reaction chemistry & engineering*, 1(6):658–666, 2016.

[33] Joseph W Richards, Dan L Starr, Henrik Brink, Adam A Miller, Joshua S Bloom, Nathaniel R Butler, J Berian James, James P Long, and John Rice. Active learning to overcome sample selection bias: application to photometric variable star classification. *The Astrophysical Journal*, 744(2):192, 2011.

[34] Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom Van de Wiele, Volodymyr Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing-solving sparse reward tasks from scratch. *arXiv preprint arXiv:1802.10567*, 2018.

[35] Matthias Rolf, Jochen J Steil, and Michael Gienger. Goal babbling permits direct learning of inverse kinematics. *IEEE Transactions on Autonomous Mental Development*, 2(3):216–229, 2010.

[36] Artur M Schweidtmann, Adam D Clayton, Nicholas Holmes, Eric Bradford, Richard A Bourne, and Alexei A Lapkin. Machine learning meets continuous flow chemistry: Automated optimization towards the pareto front of multiple objectives. *Chemical Engineering Journal*, 352:277–282, 2018.

[37] Kenneth O Stanley. Exploiting regularity without development. In *Proceedings of the AAAI Fall Symposium on Developmental Systems*, page 37. AAAI Press Menlo Park, CA, 2006.

[38] Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[39] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[40] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of modern physics*, 55(3):601, 1983.

[41] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.

[42] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658*, 2017.

Supplementary Material for "Intrinsically Motivated Exploration for Automated Discovery of Patterns in Morphogenetic Systems"

Anonymous Author(s)
Affiliation
Address
email

Contents

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

1 Lenia

1.1 System and Parameters

We used the Lenia implementation as described in [2] for all experiments. We based our code on the python implmentation provided at `https://github.com/Chakazul/Lenia`. All Lenia experiments used the following configurations:

- State size: 256×256 ($A \in \mathbb{R}^{256 \times 256}$)
- Number of steps: $T = 200$
- Growth mapping: exponential with $G(u; \mu, \sigma) = 2 \exp\left(-\frac{(u-\mu)^2}{2\sigma^2}\right) - 1$
- Kernel function: exponential with $K_C(r) = \exp\left(\alpha - \frac{\alpha}{4r(1-r)}\right)$, with $\alpha = 4$
- Kernel shell: $K_S(r; \beta) = \beta_{\lfloor Br \rfloor} K_C(Br \ mod \ 1)$, with $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)$

All other settings are part of the parameter space $\theta = [R, T, \mu, \sigma, \beta_1, \ldots, \beta_4]$, where R is the radius of the circle around a cell whose cells influence its activity. T controls the growth strength update per time step. The form of the kernel function is controlled by $\mu, \sigma, \beta_1, \ldots, \beta_4$. The cell grid of Lenia is similar to the surface of a ball, because the cells on the top border are neighbors to the bottom cells and the left and right border are also connected.

1.2 Classifier

We developed classifier for the patterns that are observed in Lenia. This section describes the classifiers in detail.

Dead Classifier: Dead patterns are all patterns where the activity of all cells are either 0 or 1.

Animal Classifier: The final pattern of a Lenia entity is classified as an animal if it is finite. In Lenia the borders of opposite sites are connected, so that the space is similar to a ball surface. We define an animal as a finite connected pattern of activity. Cells are connected as a pattern if they are active, i.e. their activity is larger than 0.1, and if they influence each other, i.e. if the activity of onle cell is in the radius of the influence of the kernel function K_S which is controlled by the parameter R. Moreover, the connected pattern must be at least 80% of all the activation in the space and a pattern must exists for the last two time steps. These constraint are used to avoid that too small patterns or chaotic entities are classified as animals.

Non-Animal Classifier: We also classified non-animal patterns which are all entities that were not dead and not an animal. These are patterns that usually spread over the whole state space or are connected via borders and therefore not finite.

Diverging Classifier: We classified further if patterns are diverging and most probably would become dead if Lenia would run for more time steps. Diverging entities are classified by measuring the average sum over the activity over the middle third of all time steps and compared it to the last third of all time steps. If the two differed by 5%, the pattern was classified as diverging and most likely would be either go fully activated or fully non-activated:

$$0.95 < \frac{\bar{A}^{\left[\frac{1}{3}T, \frac{2}{3}T\right]}}{\bar{A}^{\left[\frac{2}{3}T, T\right]}} < 1.05 \text{ with } \bar{A}^{[t_1, t_2]} = \frac{1}{t_2 - t_1} \sum_{t=t_1}^{t_2} \bar{A}^t \text{ with } \bar{A}^t = \frac{1}{N} \sum_{i=1}^{N} A_i^t$$

1.3 CPPNs for the generation of Lenia start states

To define the initial activation pattern A^1 compositional pattern producing networks (CPPN) [15], which were originally used to evolve images, are used. CPPNs are recurrent neural networks with the x and y coordinate of a grid cell and its distance r to the grid center as input. Their output is the activation for the cell. Randomly initiated CPPNs can produce various patterns. Moreover, it is possible to mutate the networks to evolve their output.

We used the following package to implement CPPNs: `https://github.com/CodeReclaimers/neat-python`. We had to adjust the sigmoid and gaussian function from to the ones used in [15] to replicate similar images.

1.4 Random initialization and mutation of parameters

Tables 1 and 2 list the parameters used to initialize and mutate the Lenia and CPPN parameters during an exploration. Please see [15] for details about the parameters of the CPPN evolution and mutation.

Table 1: Configuration for the initialization and mutation of Lenia system parameters θ for all experiments.

Parameter	Type	initial min	initial max	mutation σ	mutation min	mutation max
R	$\mathbb{N}$	2	20	0.5	2	20
T	$\mathbb{N}$	1	20	0.5	1	20
μ	$\mathbb{R}$	0	1	0.05	0	1
σ	$\mathbb{R}$	0.001	0.2	0.01	0.01	1
$b[1]$	$\mathbb{R}$	0	1	0.05	0	1
$b[2]$	$\mathbb{R}$	0	1	0.05	0	1
$b[3]$	$\mathbb{R}$	0	1	0.05	0	1
$b[4]$	$\mathbb{R}$	0	1	0.05	0	1

Table 2: Configuration for the initialization and mutation of CPPN networks that generate the initial state for the Lenia system.

Parameter	Value
Initial number of hidden neurons	4
Mutation neuron add probability	0.02
Mutation neuron delete probability	0.02
Initial connections	random connections with probability 0.6
Mutation connection add probability	0.05
Mutation connection delete probability	0.01
Initial activation functions	sigmoid, gauss
Mutation rate of activation functions	0.1
Min synapse weight	-3
Max synapse weight	3
Initial synapse weight	Gaussian Distribution with $\mu = 0$, $\sigma = 0.4$
Mutation rate of synapse weights	0.05
Mutation replace rate of synapse weights	0.01
Mutation power of synapse weights	1
Mutation enable/disable rate of synapse weights	0.02

2 Learning to represent goals with deep variational autoencoders

Variational Autoencoder (VAE) [10] is a popular approach for generative modelling. It uses a standard Gaussian prior $p(\mathbf{z}) = \mathcal{N}(0, I)$ and a diagonal Gaussian posterior $q(\mathbf{z}|\mathbf{x}, \chi) = \mathcal{N}(\mu, \sigma)$. The VAE framework couples a neural *encoder* $q(\mathbf{z}|\mathbf{x}, \chi)$ that, given a data point x, outputs the mean μ and variance σ of the representative distribution in the latent space with a neural *decoder* $p(\mathbf{x}|\mathbf{z}, \psi)$ that, given a sampled latent representation z from that distribution, tries to reconstruct the original data x. Under these assumptions, training is done by maximizing the computationally tractable evidence lower bound (with $\beta = 1$):

$$\mathcal{L}(\chi, \psi) = \underbrace{\mathbb{E}_{\mathbf{z}\sim q_\chi(\mathbf{z}|\mathbf{x})}[\log p_\psi(\mathbf{x}|\mathbf{z})]}_{a} - \beta \times \underbrace{\mathbb{D}_{KL}[q_\chi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})]}_{b} \tag{1}$$

The first term (a) represents the expected reconstruction accuracy while the second (b) is the KL divergence of the approximate posterior from the prior.

$$b = \mathbb{D}_{KL}[\mathcal{N}(\mu(\mathbf{x}), \Sigma(\mathbf{x}))\|\mathcal{N}(0, I)] = \sum_{j=1}^{d} \underbrace{\mathbb{D}_{KL}[\mathcal{N}(\mu(\mathbf{x})_j, \sigma(\mathbf{x})_j)\|\mathcal{N}(0, 1)]}_{b_j} \tag{2}$$

Many current state-of-the-art approaches [6, 8, 3, 12] build on the VAE framework and augment the VAE objective to enhance interpretability and disentanglement of the latent variables. In this paper, we couple the VAE architecture with three different objectives: the classical VAE objective (1), the β-VAE objective [6] and an augmented β-VAE objective (3).

The β-VAE objective re-weights the b term by a factor $\beta > 1$, aiming to enhance the disentangling properties of the learned latent factors. We are interested in such properties as it has been shown that it can benefit exploration [13]. However, heavily penalizing b can result in the network learning to "sacrifice" one or more of the learned latent variables in order nullify their contribution b_j to this term (2). Those dimensions become completely uninformative and useless for further exploration in the learned latent space.

$$b_{aug} = \sum_{j=1}^{d} b_j + \underbrace{\text{Var}\left([b_1, \ldots, b_d]\right)}_{c} \tag{3}$$

To prevent this phenomenon to happen, we augmented b with a new term c (3) to force the network to decrease *together* the individual contributions of the different latent variables (by minimizing the variance of the individual contributions) and not only to minimize the averaged contribution (sum). Prior work also reports this phenomenon [1, 11, 4, 5, 16, 17] and other modifications of the training objective have been proposed [16, 17].

We tested these three variants in the context of IMGEP with learned goal spaces. The VAE networks were trained on the loss function given in Equation (4) with the hyper-parameters $\{\beta = 1, \gamma = false\}$, $\{\beta = 5, \gamma = false\}$ and $\{\beta = 5, \gamma = true\}$ corresponding to the three variants outlined above. For the last two variants, $\beta = 5$ was chosen manually and was not optimized.

$$L = -a + \beta\left(b + \mathbf{1}_{\gamma} \times c\right) \tag{4}$$

In the main paper paper, the results shown for the IMGEP-PGL and IMGEP-OGL approaches have been obtained with representations trained using the last variant (β-VAE augmented objective). We present this variant for clarity purpose but it does not seem to improve the results in the context of IMGEPs with learned goal spaces. Actually, when comparing the results we could not find significant trends promoting one method over the other.

2.1 Implementation Details

Network architecture

The VAE architecture used in this paper to learn the different goal spaces can be seen in Table 3.

Table 3: Variational autoencoder architecture for the pretrained and online experiments.

Encoder	**Decoder**
Input image x: $256 \times 256 \times 1$	Input latent vector z: 8×1
Conv layer: 32 kernels 4×4, stride 2, 1-padding + ReLU	FC layers : 256 + ReLU, $16 \times 16 \times 32$ + ReLU
Conv layer: 32 kernels 4×4, stride 2, 1-padding + ReLU	TransposeConv layer: 32 kernels 4×4, stride 2, 1-padding + ReLU
Conv layer: 32 kernels 4×4, stride 2, 1-padding + ReLU	TransposeConv layer: 32 kernels 4×4, stride 2, 1-padding + ReLU
Conv layer: 32 kernels 4×4, stride 2, 1-padding + ReLU	TransposeConv layer: 32 kernels 4×4, stride 2, 1-padding + ReLU
FC layers : 256 + ReLU, 256 + ReLU, FC: 2×8	TransposeConv layer: 32 kernels 4×4, stride 2, 1-padding

Training procedure and hyper-parameters

The models are trained for 2000 epochs. The training objectives (three variants) are given in section 2 and binary cross entropy is used as reconstruction loss. We use Adam optimizer [9] ($lr = 1\text{e}{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1\text{e}{-8}$, weight decay=$1\text{e}{-5}$) with a batch size of 64.

For image data augmentation we use random x and y-translations (up to half the image size and with probability 0.3), rotation (up to 40 degrees and with probability 0.3), horizontal and vertical flipping (with probability 0.2). The translations and rotations are preceded by spherical padding to preserve Lenia spherical continuity.

In the IMGEP-PGL experiments, we keep the best model (with the higher accuracy on the validation set) that was obtained during the pretraining phase for the later exploration .

For the IMGEP-OGL models, the period of training is set to 100, meaning that every 100 runs we train the network for 40 epochs (2000 epochs in total in maximum, less if there is not enough data after the first T runs to start the training). For the importance sampling, we use a weighted random sampler that samples that samples newly-discovered images from the training data set half of the time (with a a probability of $\frac{0.5}{N}$ where N is the number of new images coming from the last 100 runs).

Data sets

For the IMGEP-PGL experiments, the data set used to train the VAE has 558 images which are distributed into a training (75%), validation (10%) and testing (15%) data sets. The images are Lenia patterns, half of them (279) are the ones manually identified by [2] as *animals* and the other half (279) are randomly initialized CPPN patterns (see Section 4.3 of the main paper).

For the IMGEP-OGL experiments, the data sets are constructed incrementally with the non-dead observations o encountered during the goal sampling phase. One image every ten is added to the validation set (10%) and the rest is used in the training set.

2.2 Visualization of the learned Goal Spaces

This section provides complementary material for the analysis made in Section 5.2 of the main paper. The goal spaces learned in the IMGEP-OGL algorithm (Figure 1 (b)) are compared to the hand-defined goal spaces from the IMGEP-HFS algorithm (Figure 1 (a)). The five-dimensional (a) and eight-dimensional (b) goal spaces are visualized with t-SNE [14] and PCA [7] dimensionality reduction techniques.

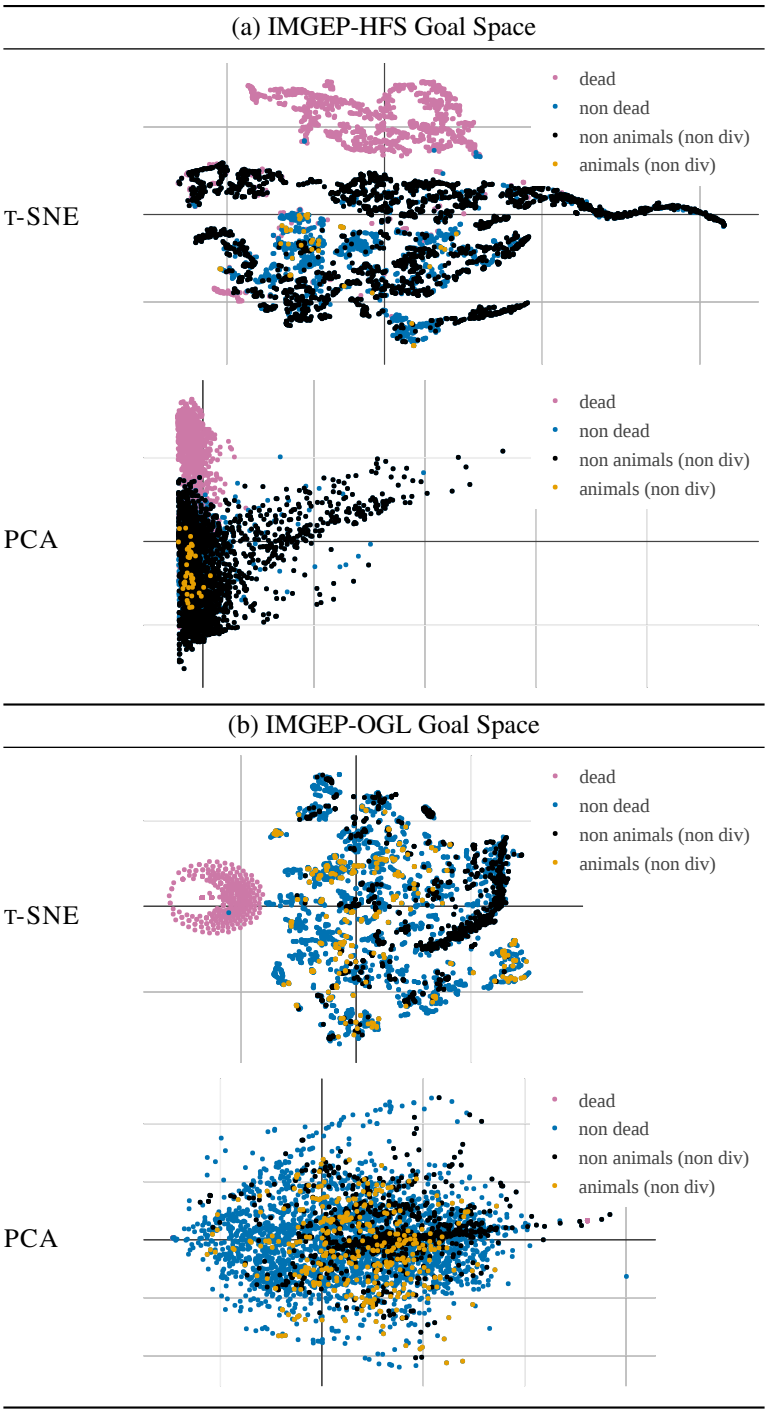

Figure 1: T-SNE and PCA visualization of the (a) hand-defined and (b) learned goal spaces

References

[1] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[2] Bert Wang-Chak Chan. Lenia-biology of artificial life. *arXiv preprint arXiv:1812.05433*, 2018.

[3] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018.

[4] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

[5] Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*, 2019.

[6] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, volume 3, 2017.

[7] I. T. Jolliffe. *Principal Component Analysis and Factor Analysis*, pages 115–128. Springer New York, New York, NY, 1986.

[8] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[11] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

[12] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

[13] Adrien Laversanne-Finot, Alexandre Pere, and Pierre-Yves Oudeyer. Curiosity driven exploration of learned disentangled goal spaces. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 487–504. PMLR, 29–31 Oct 2018.

[14] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[15] Kenneth O Stanley. Exploiting regularity without development. In *Proceedings of the AAAI Fall Symposium on Developmental Systems*, page 37. AAAI Press Menlo Park, CA, 2006.

[16] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[17] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.