
Intrinsically Motivated Exploration for Automated Discovery of Patterns in Morphogenetic Systems

Anonymous Author(s)

Affiliation

Address

email

Abstract

Exploration is a cornerstone both for machine learning algorithms and for sciences in general to discover novel solutions, phenomena and behaviors. Intrinsically motivated goal exploration processes (IMGEPs) were shown to enable autonomous agents to efficiently explore 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. 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 IMGP 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 IMGP 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 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 cornerstone 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 not. The goals are defined in a space of representations

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

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

40 So far, IMGEPs have been mainly used in the context of autonomous learning agents and robots.
41 They enabled an efficient exploration of diverse skill repertoires in high-dimensional robots [3, 12].
42 Nonetheless, their exploration capabilities are not constrained to this field and can be used in a
43 variety of application scenarios. In this paper we exemplify their application for automating the
44 discovery of complex behaviours and patterns of high-dimensional complex systems such as studied in
45 developmental (theoretical) biology, chemistry or physics. Based on our results, IMGEPs show a high
46 potential to be efficient tools for helping scientists to discover and analyze novel high-dimensional
47 self-organized structures in these complex systems. In a recent step in that direction, Grizou et al.
48 [17] showed that IMGEPs are capable of making autonomously discoveries in a chemical system
49 based on a simple low-dimensional hand engineered goal space . In this paper we show that the full
50 abilities of IMGEPs can be utilized for such environments by also learning in an unsupervised manner
51 the representations that define the goal space. We show this ability on the example of discovering
52 morphogenetic patterns in Lenia [6], a continuous game-of-life cellular automaton.

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

58 In summary, the paper provides the following new contributions:

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

63 2 Related Work

64 Inspired from the way human children can self-develop a hierarchy of skills in order to make
65 sense of the world, *intrinsically-motivated* learning [2, 3] is a family of computational models that
66 autonomously organize an agents exploration curriculum in order to discover efficiently a maximally
67 diverse set of outcomes that can be produced in an unknown environment. Intrinsically Motivated
68 Goal Exploration Processes (IMGEPs) [3, 12] are a family of curiosity-driven algorithms which have
69 been developed in the context of high-dimensional complex real world systems. Population-based
70 versions of these algorithms, leveraging episodic memory and hindsight learning, have shown to
71 enable robots or artificial agents to acquire diverse repertoires of skills [12, 35] as to bootstrap
72 the exploration capacity for deep reinforcement learning problems with rare or deceptive rewards
73 [8]. Recent work [26, 29] studies how to automatically learn the goal representations with the use
74 of deep variational autoencoders. However, the training is done in an early stage and passively on a
75 precollected set of available images. A related family of algorithms in evolutionary computation is
76 novelty search [27] and quality-diversity algorithms [30], which can be formalized as special kind of
77 population-based IMGEPs with a fixed random goal sampling policy.

78 Intrinsically motivated learning techniques have also been widely developed to handle exploration
79 in reinforcement learning, with diverse approaches ranging from estimating visitation counts [4],
80 measures of empowerment [16], goal exploration approaches [11] with hindsight learning [1] and
81 automated curriculum [9], or related concepts such as auxiliary tasks [34] and general value functions
82 [38]. However, these approaches focused on the problem of sequential decisions in MDPs (incurring
83 a cost on sample efficiency), orthogonal to the automated discovery framework considered here with
84 independent experiments allowing the use of memory-based sampling efficient methods.

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

We are using representation learning methods to identify autonomously the goal spaces for IMGEPS. 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 [7, 18, 19, 21, 25, 41] 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 the representations used as their goal spaces.

3.1 Intrinsically Motivated Goal Exploration Processes

An IMGEPEP is a sequence of experiments that explore the parameters of a system by targeting self-generated goals (Fig. 1). 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 parameter space Θ corresponding to the controllable system 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.

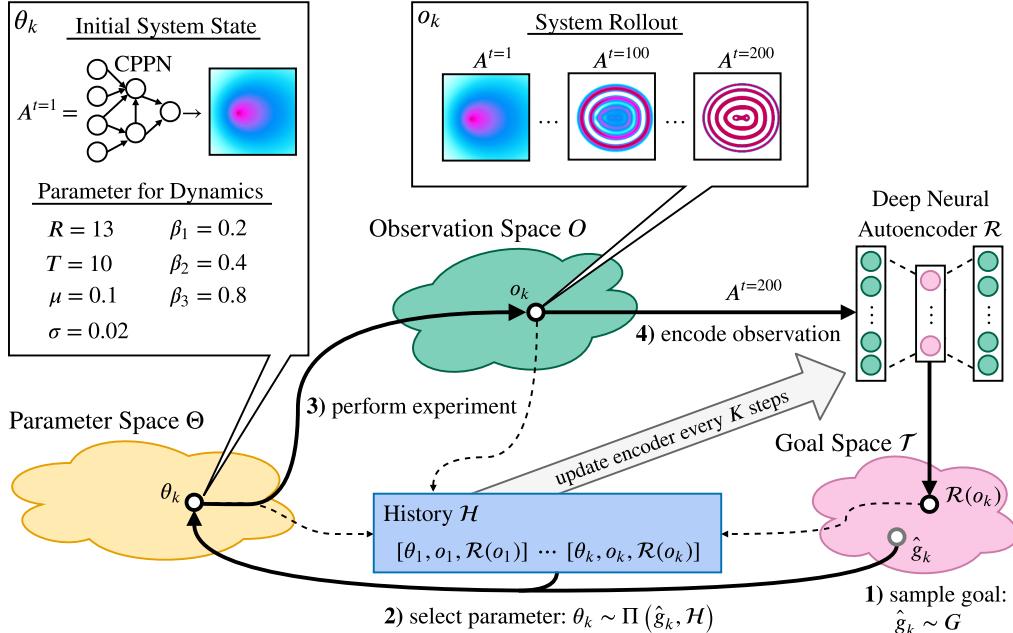


Figure 1: The intrinsically motivated goal exploration process (IMGEPEP) on the example of the IMGEPEP-OGL algorithm and Lenia as target system.

114 To explore a system, an IMGEPE defines a goal space \mathcal{T} that represents relevant features of its
 115 observations. For a robot that has to manipulate objects and observes them with a video camera,
 116 those features could be their positions. From this goal space a goal $g \sim G(\mathcal{H})$ is sampled by a goal
 117 sampling distribution. In the robot example this would correspond to a sampling of positions to
 118 which the robot should move the objects. Then, a parameter θ is chosen that will be explored to reach
 119 goal g . The parameter is chosen according to a parameter sampling policy $\Pi = \Pr(\theta; g, \mathcal{H})$. Usually,
 120 the parameter sampling policy and in some cases the goal sampling distribution take into account
 121 previous explorations which are stored in a history \mathcal{H} . After a parameter is selected it is explored
 122 on the system and the outcome o observed. Based on the observation the actually reached goal is
 123 computed using an encoding function $\hat{g} = \mathcal{R}(o)$. The encoder is either hand-defined or in the case
 124 of our online approach learned via a variational autoencoder. The reached goal is together with its
 125 corresponding parameter and observation stored in a history \mathcal{H} . The exploration process is repeated
 126 until a certain number of steps N or another constraint is reached. Because the sampling of goals and
 127 parameters depend on a history of explored parameters, an initial set of N_{init} parameters are randomly
 128 sampled and explored before the intrinsically motivated goal exploration process starts.
 129 Different goal and parameter sampling mechanisms can be used for this architecture [3, 13]. We
 130 chose for both basic approaches. Goals are sampled from a uniform distribution over the goal space.
 131 Parameters are chosen by selecting for a given goal the parameter from the history whose reached
 132 goal has the shortest distance in the goal space to the given goal. This parameter is then mutated by a
 133 random process.

134 3.2 Learning of Goal Spaces via Online Representation Learning

135 For IMGEPEs the definition of the goal space \mathcal{T} and its corresponding encoder \mathcal{R} are a critical part,
 136 because they define which observations the process tries to identify of the target system. A straight-
 137 forward choice to define a goal space is by selecting features manually, such as by using computer
 138 vision algorithms to detect the positions of objects from video images [12, 17]. A problematic point
 139 of this approach is its requirement of expert knowledge to select helpful features. Moreover, even
 140 experts might not know which features are important or how to formulate them for unknown or
 141 high-dimensional systems.

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

146 Previous approaches already applied successfully the learning of goal spaces with variational autoen-
 147 coders (VAE) [26, 29]. However, the goal spaces were learned before the start of the exploration

Algorithm 1: IMGEPE-OGL

```

1 Initialize goal space representation VAE  $\mathcal{R}$  with random weights
2 for  $i \leftarrow 1$  to  $N$  do
3   if  $i < N_{\text{init}}$  then                                // Initial random iterations to populate  $\mathcal{H}$ 
4      $\downarrow$  Sample  $\theta \sim \mathcal{U}(\Theta)$ 
5   else                                            // Intrinsically motivated iterations
6      $\downarrow$  Sample a goal  $g \sim G(\mathcal{H})$  based on space represented by  $\mathcal{R}$ 
7      $\downarrow$  Choose  $\theta \sim \Pi(g, \mathcal{H})$ 
8   Perform an experiment with  $\theta$  and observe  $o$ 
9   Encode reached goal  $\hat{g} = \mathcal{R}(o)$ 
10  Append  $(\theta, o, \hat{g})$  to the history  $\mathcal{H}$ 
11 if  $i \bmod K == 0$  then                                // Periodically train the network
12   for  $E$  epochs do
13      $\downarrow$  Train  $\mathcal{R}$  on observations in  $\mathcal{H}$  with importance sampling
14   for  $(\theta, o, \hat{g}) \in \mathcal{H}$  do                      // Update the database of reached goals
15      $\downarrow$   $\mathcal{H}[\hat{g}] \leftarrow \mathcal{R}(o)$ 

```

148 from a prerecorded dataset of observations from the target system. During the exploration the learned
149 representations were kept fixed. A problem with this pretraining approach is that training samples
150 may be limited and often biased towards the initial knowledge about the system.

151 In this paper we attempt to address this problem by continuously adapting the learned representation
152 to the observations that are encountered during the exploration process. We believe it is crucial to
153 learn the representation of features for new and unseen observations to further enable the discovery
154 of a diversity of similar observations. To address this challenge, we propose an online goal space
155 learning IMGEP (IMGEP-OGL), which learns the goal space in an incremental manner during the
156 exploration process (Algorithm 1). We evaluated different variants of VAEs for the representation
157 learning part of the algorithm. The Supplementary Material provides further details about the different
158 VAE variants.

159 The training procedure of the VAE is integrated in the goal sampling exploration process by first
160 initializing the VAE with random weights. The VAE network is then trained every K explorations for
161 E epochs on the observation collected in the history \mathcal{H} . Importance sampling is used to give more
162 weight to recently discovered patterns.

163 4 Experiments

164 The usefulness of IMGEPs for the discovery of novel patterns in complex system was evaluated
165 on the Lenia system. The following sections introduce Lenia, the different algorithms that were
166 compared and the experimental procedure. Please refer to the Supplementary Materials for further
167 details about the procedure and additional algorithm variants that have been compared.

168 4.1 Target System: Lenia

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

175 Lenia consists of a two-dimensional grid of cells $A \in [0, 1]^{256 \times 256}$ where the state of each cell is a
176 real-valued scalar activity $A^t(x) \in [0, 1]$. The state of cells evolves over discrete time steps t (Fig. 2,
177 a). The activity change is computed by integrating the activity of neighbouring cells. Lenia’s behavior
178 is controlled by its initial pattern $A^{t=1}$ and several settings that control the dynamics of the activity
179 change ($R, T, \mu, \sigma, \beta_1, \beta_2, \beta_3$).

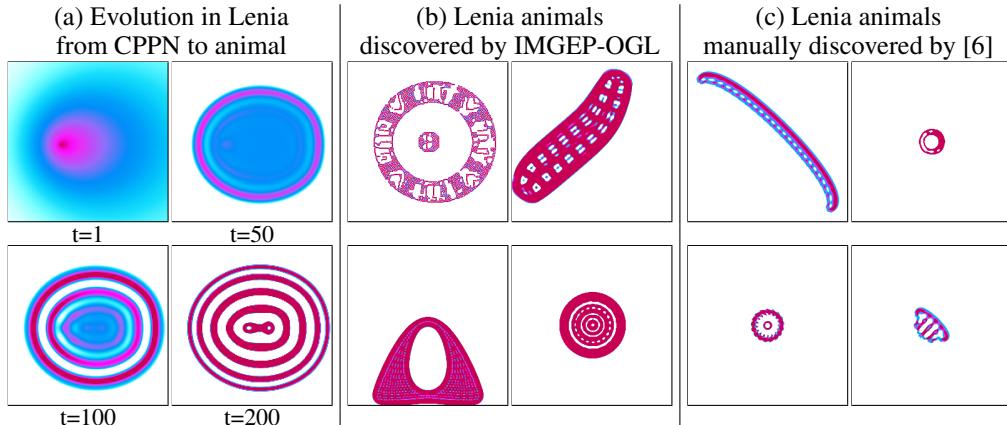


Figure 2: Example patterns produced by the Lenia system. Illustration of the dynamical morphing
from an initial CPPN image to an animal (a). The automated discovery (b) is able to find similar
complex animals as a human-expert manual search (c) by [6].

180 Lenia can be understood as a morphogenetic system where the parameters represent the genes of a
181 developmental process. They control into which final activity pattern the initial pattern morphs. Lenia
182 can produce diverse patterns with different dynamics such as stable, non-stable or chaotic patterns.
183 Most interesting, patterns that resemble microscopic animals can be produced (Fig. 2, b, c). We use
184 Lenia to study if IMGEPs can autonomously discover such patterns.

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

190 4.2 Algorithms

191 The exploration behavior of different IMGEP algorithms were evaluated and compared to a random
192 exploration. The IMGEP variants differ in their way how the goal space is defined or learned.
193 We tested for each algorithm class several variants and selected the optimal ones. Please see the
194 Supplementary Material for more information and results for the different variants.

195 **Random exploration:** The IMGEP variants were compared to a random exploration that sampled
196 randomly for each of the N exploration iterations the parameters θ and the initial state $A^{t=1}$.

197 **IMGEP-HGS - Goal exploration with a hand-defined goal space:** The first IMGEP uses a hand-
198 defined goal space that is composed of 5 features. Each feature measures a certain property of the
199 final pattern $A^{t=200}$ that emerged in Lenia: 1) the sum over the activity of all cells, 2) the number of
200 the activated cells, 3) the density of the activity center, 4) a asymmetry measure of the pattern and 5)
201 a distribution measure of the pattern.

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

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

209 4.3 Experimental Procedure

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

219 The parameters consisted of a compositional pattern producing network (CPPN) [37] that gener-
220 ates the initial state $A^{t=1}$ for Lenia and the settings defining Lenia's dynamics: $\theta = [\text{CPPN} \rightarrow$
221 $A^{t=1}, R, T, \mu, \sigma, \beta_1, \beta_2, \beta_3]$. CPPNs are recurrent neural networks that were originally used to gen-
222 erate and evolve gray scale images, but that can be similarly used to generate Lenia patterns. The
223 networks are initialized and mutated by a random process that defines their structure and connection
224 weights as done in [37]. The random initialization of the other Lenia settings was done by an uniform
225 distribution and their mutation by a Gaussian distribution around the original values. The meta
226 parameters to initialize and mutate parameters θ are equal for all algorithms (see the Supplementary
227 Material). They were manually chosen without optimizing them for a specific algorithm. The
228 parameters of the CPPN networks were set to initialize and mutate networks that generate similar
229 images as in [37].

230 **5 Results**

231 We compared random explorations and IMGEП algorithms on their ability to identify a set of Lenia
 232 patterns with a high diversity. Diversity is measured in an analytic behavior space constructed by
 233 hand-defined and learned features. Furthermore, we compared the goal spaces of hand-defined and
 234 learned IMGEП variants.

235 **5.1 Diversity of Explored Lenia Patterns**

236 Diversity is measured by the spread of the exploration in an analytic behavior space S defined by 13
 237 features. We constructed this space because the observation space O , i.e. the final Lenia patterns, are
 238 too high-dimensional. The features of the new space are: 1) the sum over the activity of all cells, 2)
 239 the number of the activated cells, 3) the density of the activity center, 4) a asymmetry measure of
 240 the pattern, 5) a distribution measure of the pattern and 6-13) a latent representation of a VAE. The
 241 VAE was trained on a dataset of Lenia patterns that were identified during all performed experiments.
 242 Please see the Supplementary Material for details.

243 For each experiment all explored patterns were projected into the analytic behavior space. To measure
 244 the diversity of the found solutions we used a simple measure of the area that the exploration covered
 245 in the analytic behavior space. The measure discretizes the analytic behavior space into bins of equal
 246 size by splitting each dimension into 7 sections resulting in 7^{13} bins. The number of bins in which at
 247 least one explored entity falls is used as a measure for diversity.

248 We also measured the diversity in the space of parameters Θ by constructing an analytic parameter
 249 space. The 15 features of this space consisted of Lenia's parameters ($R, T, \mu, \sigma, \beta_1, \beta_2, \beta_3$) and
 250 the latent representation of a VAE. The VAE was trained on a dataset of initial Lenia states ($A^{t=1}$)
 251 that were used during the experiments. Also for this diversity 7 bins per dimension where used to
 252 discretize the space.

253 Comparing the diversity between the analytic parameter and behavior space reveals the advantage
 254 of IMGEПs over random explorations (Fig. 3, a, b). Although random explorations have reached

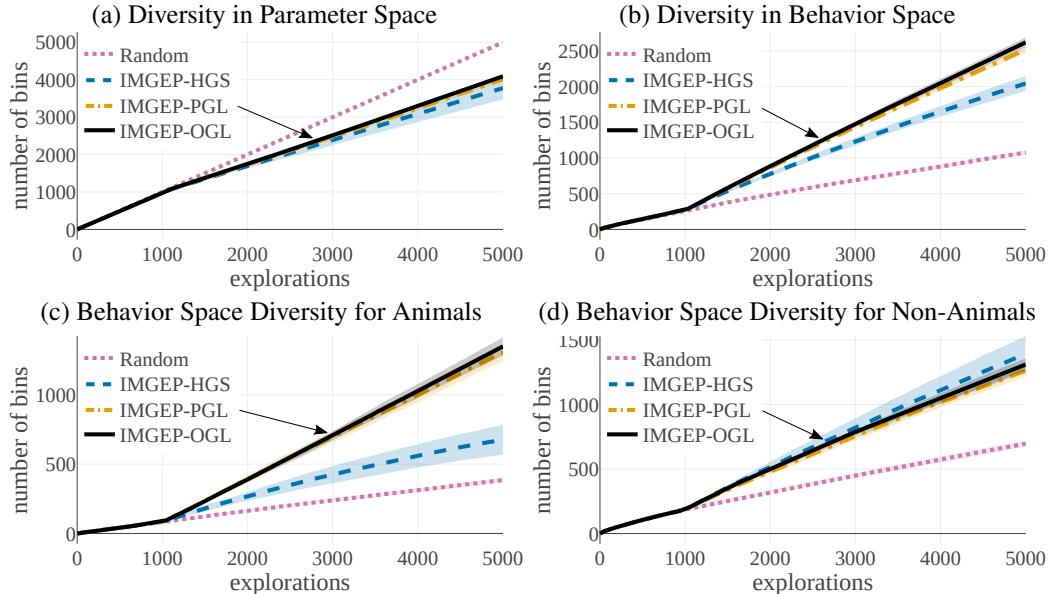


Figure 3: (a) Although random explorations reach the highest diversity in the analytic parameter space,
 (b) IMGEПs reach a higher diversity in the analytic behavior space. They discover a more diverse
 set of Lenia patterns. (c) IMGEПs with a learned goal space discovered a larger diversity of animals
 compared to the IMGEП with a hand-defined goal space. (d) 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).

255 the highest diversity in the space of parameters, they are outperformed in terms of diversity by the
256 IMGEPS approaches in the analytic behavior space. Thus, the IMGEPS approaches are better in the
257 actual objective of our exploration, finding a diverse set of Lenia patterns.

258 Both IMGEPS (PGL and OGL) with learned goal spaces reached a higher diversity over all patterns
259 than the one with a hand-defined goal space (HGS) (Fig. 3, b). Nonetheless, this is not the case
260 for certain subgroups of patterns. In the case of comparing the diversity only over explored animals
261 (Fig. 3, c) the new online approach IMGEP-OGL is finding the highest diversity of animals. It is
262 closely followed by the pretrained IMGEP-PGL approach. The hand-defined goal space approach
263 IMGEP-HGS can identify a diversity of 50% compared to IMGEP-OGL and random explorations
264 only less than 30%. In the case of diversity over non-animal patterns (Fig. 3, d) the IMGEP-HGS
265 reached the highest diversity followed by the IMGEP-OGL and IMGEP-PGL. Random explorations
266 reached the lowest diversity. These results show that the goal-space has a critical influence on the
267 type of patterns that are identified.

268 5.2 Differences in Goal Spaces

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

273 The goal spaces of IMGEP-HGS and IMGEP-OGL show strong differences between each other
274 (Fig. 4) which we believe explain their different abilities to find either a high diversity of non-animals
275 or animals (Fig. 3, c, d). The goal space of the IMGEP-HGS shows large areas and several clusters for
276 non-animal patterns (Fig. 4, a). Animals form only few and nearby clusters. Thus, the hand-defined
277 features seem poor to discriminate and describe animal patterns in Lenia. As a consequence, when
278 goals are uniformly sampled within this goal space during the exploration process, then more goals
279 are generated in regions that describe non-animals. This can explain why IMGEP-HGS explored a
280 higher diversity of non-animal patterns.

281 In contrast, features learned by IMGEP-OGL capture better factors that differentiate animal patterns
282 as can be observed by several clusters of animals that span a wide area in its goal space (Fig. 4, b). We
283 attribute this effect to the difficulty of VAEs to capture sharp details [42]. They therefore represent
284 mainly the general form of the patterns but not their fine-grained structures. Animals differ often
285 in their form whereas non-animals occupy often the whole cell grid and differ in their fine-grained
286 details. The goal spaces learned by VAEs seem therefore better suited for exploring diverse sets of
287 animal patterns.

288 6 Conclusion

289 We presented in this paper the application of intrinsically motivated exploration via IMGEPS towards
290 a new and exciting application area: the discovery of patterns and structures in complex systems.

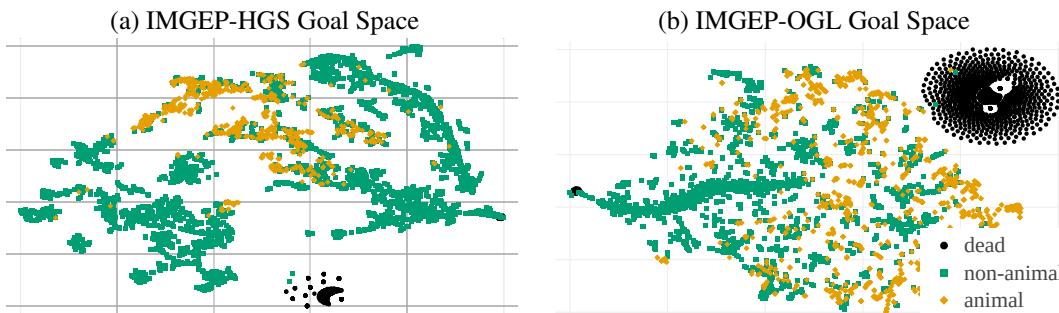


Figure 4: (a) Hand-defined and (b) learned goal spaces have major differences shown here by a t-SNE visualization. The different size of areas covered by animals or non-animals can explain the differences in their resulting diversity between the algorithms (Fig. 3).

291 All evaluated IMGEPS variants were able to discover a diverse set of patterns for Lenia, a cellular
292 automaton, by directly exploring its high-dimensional parameters and observing its high-dimensional
293 output patterns. We could demonstrate that goal spaces for such systems can be successfully learned
294 via deep VAEs which allow the identification of animal-like patterns similar to those identified
295 by human experts (Fig. 2). Moreover, our new approach of learning goal spaces online via data
296 collected during the exploration process could outperform a pretrained and fixed goal space in terms
297 of identifying a diverse set of animal-like patterns.
298 We believe that IMGEPS are able to facilitate the study of similar complex and high-dimensional
299 systems in different fields of engineering and science such as physics and chemistry. IMGEPS allow
300 to explore unknown systems efficiently to discover their interesting behaviors and patterns which can
301 help to understand the systems better or to find new solutions for problems in them.

302 References

- 303 [1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder,
304 Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience
305 replay. In *Advances in Neural Information Processing Systems*, pages 5048–5058, 2017.
- 306 [2] Gianluca Baldassarre and Marco Mirolli. *Intrinsically motivated learning in natural and
307 artificial systems*. Springer, 2013.
- 308 [3] Adrien Baranes and Pierre-Yves Oudeyer. Active learning of inverse models with intrinsically
309 motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013.
- 310 [4] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi
311 Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural
312 Information Processing Systems*, pages 1471–1479, 2016.
- 313 [5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and
314 new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–
315 1828, 2013.
- 316 [6] Bert Wang-Chak Chan. Lenia-biology of artificial life. *arXiv preprint arXiv:1812.05433*, 2018.
- 317 [7] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of
318 disentanglement in variational autoencoders. In *Advances in Neural Information Processing
319 Systems*, pages 2610–2620, 2018.
- 320 [8] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. Gep-pg: Decoupling exploration and
321 exploitation in deep reinforcement learning algorithms. *arXiv preprint arXiv:1802.05054*, 2018.
- 322 [9] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. Curious: Intrinsically motivated multi-
323 task, multi-goal reinforcement learning. In *International Conference on Machine Learning
324 (ICML)*, 2019.
- 325 [10] Vasilios Duros, Jonathan Grizou, Weimin Xuan, Zied Hosni, De-Liang Long, Haralampos N
326 Miras, and Leroy Cronin. Human versus robots in the discovery and crystallization of gigantic
327 polyoxometalates. *Angewandte Chemie*, 129(36):10955–10960, 2017.
- 328 [11] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse
329 curriculum generation for reinforcement learning. *arXiv preprint arXiv:1707.05300*, 2017.
- 330 [12] Sébastien Forestier, Yoan Mollard, and Pierre-Yves Oudeyer. Intrinsically motivated goal
331 exploration processes with automatic curriculum learning. *arXiv preprint arXiv:1708.02190*,
332 2017.
- 333 [13] Sébastien Forestier and Pierre-Yves Oudeyer. Modular active curiosity-driven discovery of tool
334 use. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*,
335 pages 3965–3972. IEEE, 2016.
- 336 [14] Martin Gardner. Mathematical games: The fantastic combinations of john conway’s new
337 solitaire game "life.". *Scientific American*, 223:120–123, 1970.

- 338 [15] Jarosław M Granda, Liva Donina, Vincenza Dragone, De-Liang Long, and Leroy Cronin.
339 Controlling an organic synthesis robot with machine learning to search for new reactivity.
340 *Nature*, 559(7714):377, 2018.
- 341 [16] Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv
342 preprint arXiv:1611.07507*, 2016.
- 343 [17] Jonathan Grizou, Laurie J Points, Abhishek Sharma, and Leroy Cronin. Exploration of self-
344 propelling droplets using a curiosity driven robotic assistant. *arXiv preprint arXiv:1904.12635*,
345 2019.
- 346 [18] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David
347 Vazquez, and Aaron Courville. Pixelvae: A latent variable model for natural images. *arXiv
348 preprint arXiv:1611.05013*, 2016.
- 349 [19] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick,
350 Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a
351 constrained variational framework. In *International Conference on Learning Representations*,
352 volume 3, 2017.
- 353 [20] Claudia Houben and Alexei A Lapkin. Automatic discovery and optimization of chemical
354 processes. *Current opinion in chemical engineering*, 9:1–7, 2015.
- 355 [21] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*,
356 2018.
- 357 [22] Ross D King, Jem Rowland, Stephen G Oliver, Michael Young, Wayne Aubrey, Emma Byrne,
358 Maria Liakata, Magdalena Markham, Pinar Pir, Larisa N Soldatova, et al. The automation of
359 science. *Science*, 324(5923):85–89, 2009.
- 360 [23] Ross D King, Kenneth E Whelan, Ffion M Jones, Philip GK Reiser, Christopher H Bryant,
361 Stephen H Muggleton, Douglas B Kell, and Stephen G Oliver. Functional genomic hypothesis
362 generation and experimentation by a robot scientist. *Nature*, 427(6971):247, 2004.
- 363 [24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint
364 arXiv:1312.6114*, 2013.
- 365 [25] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of
366 disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*,
367 2017.
- 368 [26] Adrien Laversanne-Finot, Alexandre Pere, and Pierre-Yves Oudeyer. Curiosity driven explo-
369 ration of learned disentangled goal spaces. In Aude Billard, Anca Dragan, Jan Peters, and
370 Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of
371 *Proceedings of Machine Learning Research*, pages 487–504. PMLR, 29–31 Oct 2018.
- 372 [27] Joel Lehman and Kenneth O Stanley. Exploiting open-endedness to solve problems through the
373 search for novelty. In *ALIFE*, pages 329–336, 2008.
- 374 [28] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine
375 learning research*, 9(Nov):2579–2605, 2008.
- 376 [29] Alexandre Pérez, Sébastien Forestier, Olivier Sigaud, and Pierre-Yves Oudeyer. Unsupervised
377 Learning of Goal Spaces for Intrinsically Motivated Goal Exploration. In *ICLR2018 - 6th
378 International Conference on Learning Representations*, Vancouver, Canada, April 2018.
- 379 [30] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. Quality diversity: A new frontier for
380 evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.
- 381 [31] Paul Raccuglia, Katherine C Elbert, Philip DF Adler, Casey Falk, Malia B Wenny, Aurelio
382 Mollo, Matthias Zeller, Sorelle A Friedler, Joshua Schrier, and Alexander J Norquist. Machine-
383 learning-assisted materials discovery using failed experiments. *Nature*, 533(7601):73, 2016.

- 384 [32] Brandon J Reizman, Yi-Ming Wang, Stephen L Buchwald, and Klavs F Jensen. Suzuki-miyaura
385 cross-coupling optimization enabled by automated feedback. *Reaction chemistry & engineering*,
386 1(6):658–666, 2016.
- 387 [33] Joseph W Richards, Dan L Starr, Henrik Brink, Adam A Miller, Joshua S Bloom, Nathaniel R
388 Butler, J Berian James, James P Long, and John Rice. Active learning to overcome sample
389 selection bias: application to photometric variable star classification. *The Astrophysical Journal*,
390 744(2):192, 2011.
- 391 [34] Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom
392 Van de Wiele, Volodymyr Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by
393 playing-solving sparse reward tasks from scratch. *arXiv preprint arXiv:1802.10567*, 2018.
- 394 [35] Matthias Rolf, Jochen J Steil, and Michael Gienger. Goal babbling permits direct learning of
395 inverse kinematics. *IEEE Transactions on Autonomous Mental Development*, 2(3):216–229,
396 2010.
- 397 [36] Artur M Schweidtmann, Adam D Clayton, Nicholas Holmes, Eric Bradford, Richard A Bourne,
398 and Alexei A Lapkin. Machine learning meets continuous flow chemistry: Automated op-
399 timization towards the pareto front of multiple objectives. *Chemical Engineering Journal*,
400 352:277–282, 2018.
- 401 [37] Kenneth O Stanley. Exploiting regularity without development. In *Proceedings of the AAAI*
402 *Fall Symposium on Developmental Systems*, page 37. AAAI Press Menlo Park, CA, 2006.
- 403 [38] Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam
404 White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge
405 from unsupervised sensorimotor interaction. In *The 10th International Conference on Au-*
406 *tomous Agents and Multiagent Systems-Volume 2*, pages 761–768. International Foundation
407 for Autonomous Agents and Multiagent Systems, 2011.
- 408 [39] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based
409 representation learning. *arXiv preprint arXiv:1812.05069*, 2018.
- 410 [40] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of modern physics*,
411 55(3):601, 1983.
- 412 [41] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational
413 autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.
- 414 [42] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Towards deeper understanding of variational
415 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

1 Overview

- 2 The Supplementary Material for the paper "Intrinsically Motivated Exploration for Automated
- 3 Discovery of Patterns in Morphogenetic Systems" provides implementation details and additional
- 4 results. Section 1 introduces the target system of all experiments, Lenia, in more detail. The sampling
- 5 procedure for the parameters that control Lenia are described in Section 2. Section 3 describes
- 6 in more detail the diversity measure used to compare algorithms and the definition of the analytic
- 7 parameter and behavior space in which it is measured. Afterwards results for one additional random
- 8 exploration and several variations of IMGEPs with hand-defined goal spaces are provided in Section 4.
- 9 In Section 5 a results provided for different VAE variants used to learn goal spaces for IMGEPs.
- 10 Finally, Section 6 lists additional results and analysis for the experiments listed in the main paper.

11 Contents

12	1 Target System: Lenia	3
13	1.1 Implementation Details and Parameters	3
14	1.2 Classifier	3
15	1.3 Statistical Measures for Lenia Patterns	6
16	2 Sampling of Parameters for Lenia	8
17	2.1 Sampling of Start Patterns for Lenia via CPPNs	8
18	2.2 Sampling of Lenia's Dynamic Parameters	11
19	3 Measurement of Diversity in the Analytic Parameter and Behavior Space	12
20	3.1 Diversity Measure	12
21	3.2 Analytic Parameter Space	12
22	3.3 Analytic Behavior Space	12
23	4 Random Exploration and IMGEPs with Hand-Defined Goal Spaces	16
24	4.1 Random Explorations	16
25	4.2 IMGEPs with Hand-Defined Goal Spaces	16
26	4.3 Results	16
27	5 IMGEPs with Learned Goal Spaces via Deep Variational Autoencoders	19
28	5.1 VAE Variants	19
29	5.2 Implementation Details	20
30	5.3 Results	21
31	5.3.1 Diversity	22
32	5.3.2 VAE Pattern Reconstruction	22
33	6 Additional Results	26
34	6.1 Number of Identified Patterns	26
35	6.2 Dependence of the Diversity Measure on the Number of Bins per Dimension	27
36	6.3 Dimension Reduction of the Analytic Parameter and Behavior Space	27
37	6.4 Identified Patterns	29
38	6.5 Visualization of Goal Spaces	29

39 **1 Target System: Lenia**

40 Lenia was used as the target system for all exploration experiments. The following section describes
 41 Lenia and the parameters to control its behavior in detail. It is followed by a description of the
 42 classifiers used to categorize dead, animal and non-animal Lenia patterns. Finally, statistical measures
 43 about the patterns are introduced which were used to define goal and analytic spaces.

44 **1.1 Implementation Details and Parameters**

45 Lenia [2] is a cellular automaton [18]. It consists of a two-dimensional grid of cells $A \in [0, 1]^{L \times L}$
 46 with $L = 256$ for all experiments. The cell grid is similar to the surface of a ball. Cells on the north
 47 border are neighbors to the south border cells. The east and west border are also connected. The
 48 state of each cell is a real-valued scalar activity $A(x) \in [0, 1]$. The states of cells A^t evolve over
 49 discrete time steps $t = [1, \dots, M]$ with $M = 200$ for all experiments. The activity change of a cell
 50 is computed by integrating the previous activity of its neighbouring cells:

$$A^{t+1}(x) = [A^t(x) + \Delta t G(K * A^t(x))]_0^1,$$

51 where G is the *growth mapping*, K is the *kernel*, $\Delta t = \frac{1}{T}$ with $T \in \mathbb{N}$ is the *time step* and
 52 $[n]_b^a = \min(\max(n, a), b)$ is the clip function. For all experiments an exponential growth mapping
 53 was used:

$$G(u; \mu, \sigma) = 2 \exp\left(-\frac{(u - \mu)^2}{2\sigma^2}\right) - 1,$$

54 with $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}$ being paremeters that control its shape.

55 The kernel integrates the activity of the current cell x and its neighbours by a convolution with a
 56 *kernel function* $K(n)$:

$$K * A^t(x) = \sum_{n \in \mathcal{N}(x)} K(n) A^t(x + n) \Delta x^2, \quad (1)$$

57 where \mathcal{N} is the *neighborhood* around the cell x and $\Delta x = \frac{1}{R}$ with $R \in \mathbb{N}$ is the site distance. The
 58 neighborhood is defined by a circle around x with radius R : $\mathcal{N}(x) = \{y : \|x - y\|_2 \leq R\}$. The
 59 kernel is constructed by a *kernel core* function $K_C : [0, 1] \mapsto [0, 1]$ and a *kernel shell* function
 60 $K_S : [0, 1] \mapsto [0, 1]$. The kernel core creates a ring around the center coordinate and is defined by an
 61 exponential:

$$K_C(r) = \exp\left(\alpha - \frac{\alpha}{4r(1-r)}\right) \text{ with } \alpha = 4.$$

62 The kernel shell K_S takes a vector parameter $\beta = (\beta_1, \beta_2, \beta_3) \in [0, 1]^3$ and copies the kernel core
 63 into concentric rings. The rings are of equal thickness with peak heights β_i :

$$K_S(r; \beta) = \beta_{\lfloor Br \rfloor} K_C(Br \bmod 1).$$

64 Finally, the kernel is normalized:

$$K(n) = \frac{K_S(\|n\|_2)}{|K_S|}.$$

65 In total 8 parameters $\theta = \{A^{t=1}, R, T, \mu, \sigma, \beta_1, \beta_2, \beta_3\}$ controlled the behavior of Lenia for all
 66 experiments. $A^{t=1}$ is the starting pattern of the system. R is the radius of the circle around a cell
 67 x whose enclosed cells influence the activity of x . T controls the growth strength update per time
 68 step. The growth mapping is controlled by μ and σ . The form of the kernel function is controlled by
 69 $\beta_1, \beta_2, \beta_3$.

70 We based our Python implementation of Lenia on the code provided by <https://github.com/Chakazul/Lenia>.

72 **1.2 Classifier**

73 We categorized 3 types of patterns that are observed in Lenia. The categories were used to analyze if
 74 the exploration algorithms showed differences in their exploration behaviors by identifying different
 75 types of patterns. The 3 categories are dead, animals and non-animals. For each class is a classifier

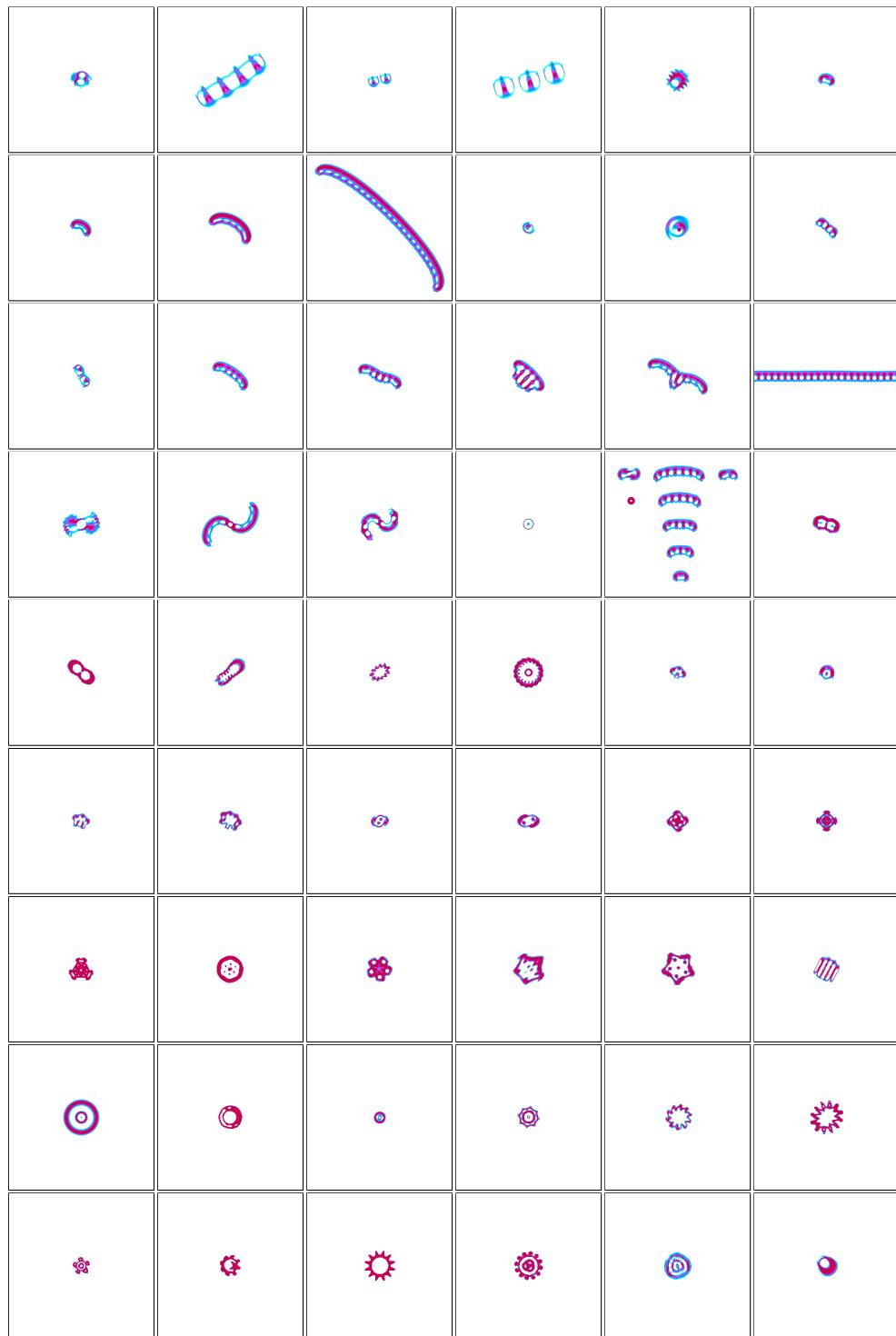


Figure 1: Patterns identified by a human-expert [2].

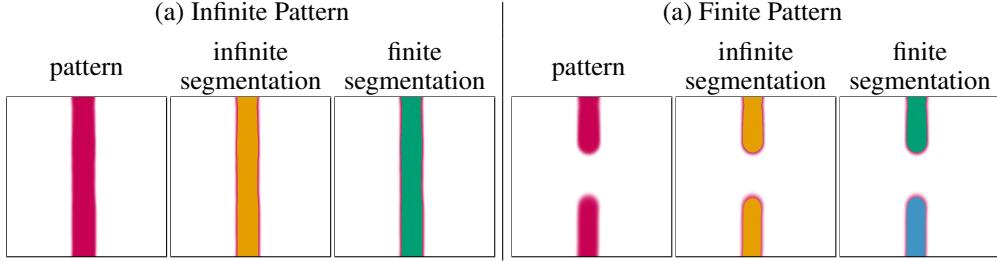


Figure 2: Classification of Lenia patterns into finite and infinite patterns. Infinite patterns form loop between the image borders which are identified if a segment is connected between two borders in the infinite and finite segmentation. Finite patterns form no loops. They have connected segments between borders in the infinite but not finite segmentation. Segments are colorized in yellow, green and blue.

76 defined. The classifiers only classify the final pattern in which the Lenia system morphs after
 77 $M = 200$ time steps.

78 **Dead Classifier:** For dead patterns is the activity of all cells either 0 or 1 in the last time step $t = M$.

79 **Animal Classifier:** The final Lenia pattern is classified as an animal if it is a *finite* and *connected*
 80 pattern of activity. Cells x, y are connected as a pattern if both are active ($A(x) \geq 0.1$ and $A(y) \geq 0.1$)
 81 and if they influence each other. Cells influence each other when they are within their radius of the
 82 kernel K as defined by the parameter R (Eq. 1).

83 Furthermore, the connected pattern must be finite. In Lenia finite and infinite patterns can be
 84 differentiated because the opposite borders of Lenia's cell grid are connected, so that the space is
 85 similar to a ball surface. Thus, a pattern can loop around this surface making it infinite. We identify
 86 infinite patterns by the following approach. First, all connected patterns are identified for the case of
 87 assuming an infinite grid cell, i.e. opposite grid cell borders are connected. Second, all connected
 88 patterns for the case of a finite grid cell, i.e. opposite grid cell borders are not connected, are identified.
 89 Third, for each border pair (north-south and east-west) it is tested if cells within a distance of R from
 90 both borders exists, that are part of a connected pattern for the infinite and finite grid cell case. If
 91 such a pattern exists than it is assumed to be infinite, because it loops around the grid cell surface
 92 of Lenia (Fig. 2, a). All other patterns are considered to be finite (Fig. 2, b). Please note that this
 93 method has a drawback. It can not identify certain infinite patterns that loop over several borders, for
 94 example, if a pattern exists that connects the north to east and then the west to south border (Fig. 3).

95 Moreover, there are two additional constraints that an animal pattern must fulfill. First, the cells of
 96 the connected pattern $P = \{x_1, \dots, x_n\}$ must have at least 80% of all activation, i.e. $\sum_{x \in P} A(x) \geq$
 97 $0.8 \sum_{y \in P} A(y)$. Second, a pattern must exist for the last two time steps ($t = M$ and $t = M - 1$).
 98 Both constraint are used to avoid that too small patterns or chaotic entities which change drastically
 99 between time steps are classified as animals. See Fig. 1, 23, 24, 25 and 26 for examples of animal
 100 patterns.

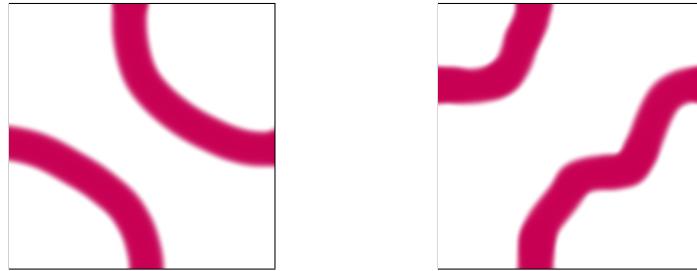


Figure 3: Examples of infinite patterns that are misclassified as finite patterns.

101 **Non-Animal Classifier:** We also classified non-animal patterns which are all entities that were not
 102 dead and not an animal. These patterns spread usually over the whole state space and are connected
 103 via borders. See Fig. 23, 24, 25 and 26 for examples of non-animal patterns.

104 1.3 Statistical Measures for Lenia Patterns

105 We defined five statistical measurements for the final patterns $A^{t=M}$ that emerge in Lenia. The
 106 measures were used as features for hand-defined goal spaces of IMGEPs and to define partly the
 107 analytic behavior space in which the results of the exploration experiments were compared.

108 **Activation mass M_A :** Measures the sum over the total activation of the final pattern and normalizes
 109 it according to the size of the Lenia grid:

$$M_A = \frac{1}{L^2} \sum_x A^{t=M}(x),$$

110 where $L^2 = 256 \cdot 256$ is the number of cells of the Lenia system.

111 **Activation volume V_A :** Measures the number of active cells and normalizes it according to the size
 112 of the Lenia grid:

$$V_A = \frac{1}{L^2} |\{\forall x : A^{t=M}(x) > \epsilon\}| \text{ with } \epsilon = 10^{-4}.$$

113 **Activation density D_A :** Measures how dense the activation is distributed on average over all active
 114 cells:

$$D_A = \frac{M_A}{V_A}.$$

115 **Activation asymmetry A_A :** Measures how symmetrical the activation is distributed according to an
 116 axis that starts in the center of the patterns activation mass and goes along the last movement direction
 117 of this center. This measure was introduced to especially characterize animal patterns such as shown
 118 in Fig. 1. The center of the activity mass is usually also the center of the animals and analyzing the
 119 activity along their movement axis measures how symmetrical they are.

120 As a first step, the center of the activation mass is computed for every time step of the Lenia simulation
 121 and the Lenia pattern recentered to this location. This ensures that the center is all the time correctly
 122 computed in the case the animal moves and reaches one border to appear on the opposite border in
 123 the uncentered pattern. The center $(\bar{x}, \bar{y})_t$ for time step t is calculated by:

$$(\bar{x}, \bar{y})_t = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right) \text{ with } M_{pq} = \sum_x \sum_y x^p y^q A^t(x, y),$$

124 where M_{pq} measures the image moment (or raw moment) of order $(p + q)$ for $p, q \in \mathbb{N}$.

125 Based on the center $(\bar{x}, \bar{y})_t$ the pattern A^t is recentered to A_C^t by shifting the x and y indexes
 126 according to the center:

$$A_C^t(x, y) = A^t((x - \bar{x}) \bmod L, (y - \bar{y}) \bmod L), \quad (2)$$

127 where L is width and length of the Lenia grid and the indexing is $x, y = 0, \dots, L - 1$. After each
 128 time step the center is recomputed and the pattern recentered:

$$A^{t=1} \xrightarrow{\text{recenter}} A_C^{t=1} \xrightarrow{\text{Lenia step}} A^{t=2} \xrightarrow{\text{recenter}} A_C^{t=2} \xrightarrow{\text{Lenia step}} \dots$$

129 Please note, the simulations and all figures of patterns in the paper are done with the uncentered
 130 pattern. The centered version is only computed for the purpose of statistical measurements.

131 The recenter step by $(\bar{x}, \bar{y})_t$ defines also the movement direction of the activity center:

$$(m_x, m_y)_t = (\bar{x}, \bar{y})_t - (x^{\text{mid}}, y^{\text{mid}}) = (\bar{x} - x^{\text{mid}}, \bar{y} - y^{\text{mid}}),$$

132 where $x^{\text{mid}}, y^{\text{mid}} = \frac{L-1}{2}$ are the coordinates for the middle point of the grid. A line can be defined
 133 that starts in the midpoint $(x^{\text{mid}}, y^{\text{mid}})$ of the final centered pattern $A_C^{t=M}$ and goes in and opposite
 134 to the final movement direction of the activity mass center $(m_x, m_y)_{t=M}$. This line separates the

135 grid in two equal areas. The asymmetry is computed by comparing the amount of activity in the grid
 136 right M_A^{right} and left M_A^{left} of the line. The normalized difference between both sides is the final
 137 asymmetry measure:

$$A_A = \frac{1}{M_A} (M_A^{right} - M_A^{left}).$$

138 **Activation centeredness** C_A : Measures how strong the activation is distributed around the activity
 139 mass center:

$$C_A = \frac{1}{M_A} \sum_x \sum_y w_{xy} \cdot A_C^{t=M}(x, y) \quad \text{with} \quad w_{xy} = \left(1 - \frac{d(x, y)}{\max_{y,x} d(x, y)}\right)^2,$$

140 where $d(x, y) = \sqrt{(x - x^{\text{mid}})^2 + (y - y^{\text{mid}})^2}$ is the distance from the point (x, y) to the center
 141 point $(x^{\text{mid}}, y^{\text{mid}})$. $A_C^{t=M}(x, y)$ is the centered activation that is updated every time step as for the
 142 asymmetry measure (Eq. 2). The weights w_{xy} decrease the farther a point is from the center. Thus,
 143 patterns that are concentrated around the center have a high value for C_A close to 1. Whereas, patterns
 144 whos activity is distributed throughout the whole grid have a smaller value. For patterns that are
 145 equally distributed ($\forall_{x,x'} : A(x) = A(x')$) is $C_A = 0$ defined as centeredness measure.

146 **2 Sampling of Parameters for Lenia**

147 All exploration algorithms explore Lenia patterns by sampling the parameters θ that control Lenia.
 148 The parameters are comprised of the initial pattern $A^{t=1}$ and the parameters which control the
 149 dynamic behavior ($R, T, \mu, \sigma, \beta_1, \beta_2, \beta_3$). There are two operations to sample parameters: 1) random
 150 initialization and 2) mutating an existing parameter θ . CPPNs are used for the random initialization
 151 and mutation of the initial pattern $A^{t=1}$. The details of this process are described in the next
 152 section. Afterwards, the initialization and mutation of Lenia's parameter that control its dynamics are
 153 described.

154 **2.1 Sampling of Start Patterns for Lenia via CPPNs**

155 CPPNs (Compositional Pattern Producing Networks) are recurrent neural networks that were developed
 156 for the generation and evolution of gray-scale 2D images [15]. We used CPPNs to generate and
 157 mutate the initial state of Lenia $A^{t=1}$ which resembles an image. CPPNs generate images pixel by
 158 pixel by taking as input a bias value, the x and y coordinate of the pixel in the image and its distance
 159 d to the image center (Fig. 4). Their output is the pixel value as a gray scale between 0 and 1 for the
 160 given (x, y) coordinate. For the generation of initial Lenia patterns is as input the x and y coordinate
 161 of the grid cells used. They were mapped to $x = [-2, 2]$ and $y = [-2, 2]$. The distance to the grid
 162 center is given by $d(x, y) = \sqrt{x^2 + y^2}$. The final activity of a cell is the remapped output p of the
 163 CPPN via $A(x, y) = 1 - |p|$.

164 CPPNs consist of several hidden neurons (typically between 4 to 6 in our experiments) that can
 165 have recurrent connections and self connections. Each CPPN has one output neuron. Two activation
 166 functions were used for the hidden neurons and the output neuron. The first is Gaussian and the
 167 second is sigmoidal:

$$\text{gauss}(x) = 2 \exp(-(2.5x)^2) - 1, \quad (3)$$

$$\text{sigm}(x) = 2 \left(\frac{1}{1 + \exp(-5x)} \right) - 1. \quad (4)$$

168 To randomly initialize a Lenia initial pattern $A^{t=1}$ a CPPN is randomly sampled by sampling the
 169 number of hidden neurons, the connections between inputs and neurons and neurons to neurons,
 170 their connection weights and the activation functions for neurons. Afterwards the initial pattern
 171 is generated by it. In the history \mathcal{H} of the IMGEPs is then the CPPN as part of the parameter
 172 θ added. If the parameter is mutated, then the weights, connections and activation functions of
 173 the CPPN are mutated and the new initial pattern $A^{t=1}$ generated by it. We used the *neat-python*
 174 package (<https://github.com/CodeReclaimers/neat-python>) for the random generation and
 175 mutation of CPPNs. It is based on the NEAT (NeuroEvolution of Augmenting Topologies) algorithm
 176 for the evolution of neural networks [16]. The meta-parameters for the initialization and mutation
 177 of CPPNs are listed in Table 1. The random sampling and mutation of CPPNs allows to generate
 178 complex patterns as illustrated in Fig. 5.
 179

180 The random sampling of a new CPPN is done by the following steps. All CPPNs are initialized
 181 with 4 hidden neurons and 1 output neuron. Their activation functions are randomly assigned. Each

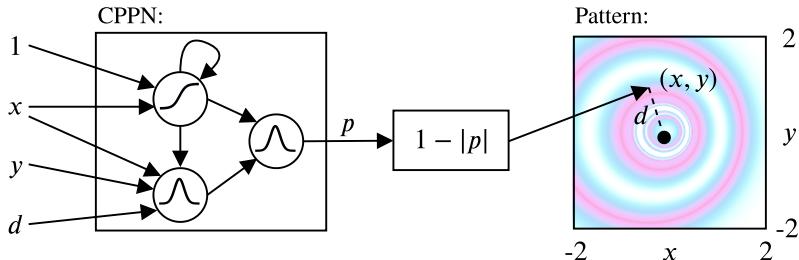


Figure 4: CPPNs are recurrent neural networks which take as input a bias of 1, the x and y coordinate of a point in the generated pattern and its distance r to the center of the pattern. Their output is the pixel value for the point.

Parameter	Value
Initial number of hidden neurons	4
Initial activation functions	gauss, sigm
Initial connections	random connections with probability 0.6
Initial synapse weight	Gaussian distribution with $\mu = 0, \sigma = 0.4$
Synapse weight range	$[-3, 3]$
Mutation neuron add probability	0.02
Mutation neuron delete probability	0.02
Mutation connection add probability	0.05
Mutation connection delete probability	0.01
Mutation rate of activation functions	0.1
Mutation rate of synapse weights	0.05
Mutation replace rate of synapse weights	0.06
Mutation power of synapse weights σ_M	1
Mutation enable/disable rate of synapse weights	0.02

Table 1: Configuration for the initialization and mutation of CPPN networks that generate the initial state for the Lenia system.

182 input-hidden, hidden-hidden and hidden-output neuron pair is connected with a probability of 0.6.
 183 The weights of each connection are sampled via a Gaussian distribution: $w_{ij} \sim \mathcal{N}(\mu = 0, \sigma = 0.4)$.
 184 The maximum and minimum weights for a connection are -3 and 3 .
 185 An existing CPPN is mutated by the following procedure. At first, structural mutations are performed.
 186 With probability 0.02 a new neuron z with a random activation function is added. The neuron is
 187 connected to the network by choosing randomly an existing connection. This connection is deleted.
 188 A connection from the source i of the deleted connection to the new neuron is added with weight
 189 $w_{iz} = 1.0$. Additionally, a new connection from the new neuron to the target j of the deleted
 190 connection is added with the old connection weight $w_{zj} = w_{ij}$, finishing the addition of a new
 191 neuron. With probability 0.02 one of the hidden neurons is deleted. With probability 0.05 a new
 192 connection is added between a random input-hidden, hidden-hidden or hidden-output neuron pair.
 193 The connection weight is sampled by the same method as for the sampling of new CPPNs. With
 194 probability 0.01 one random existing connection is removed. After the structural mutations the
 195 activation functions and weights are mutated. For each neuron the activation function is changed
 196 with probability 0.1 by randomly assigning a new activation function (either gauss or sigm). For each
 197 connection the weight is mutated by the following steps. With probability 0.05 the weight of the
 198 connection is changed according to:

$$w_{ij} \leftarrow [w_{ij} + \mathcal{N}(0, \sigma_M)]_{-3}^3 ,$$

199 where $\sigma_M = 1$ is the mutation power and $[n]_b^a = \min(\max(n, a), b)$ is the clip function. With
 200 probability 0.06 the connection weight is completely replaced by sampling a new one as done for the
 201 sampling of weights of new CPPNs.

202 Please note, the neat-python package allows also the setting and mutation of response and bias
 203 weights for each neuron. Those settings were not used for the experiments. Moreover, we adjusted
 204 the sigmoid and Gaussian function in the neat-python package to the ones defined in Eq. 3 and Eq. 4
 205 to be able to replicate similar images as in [15].

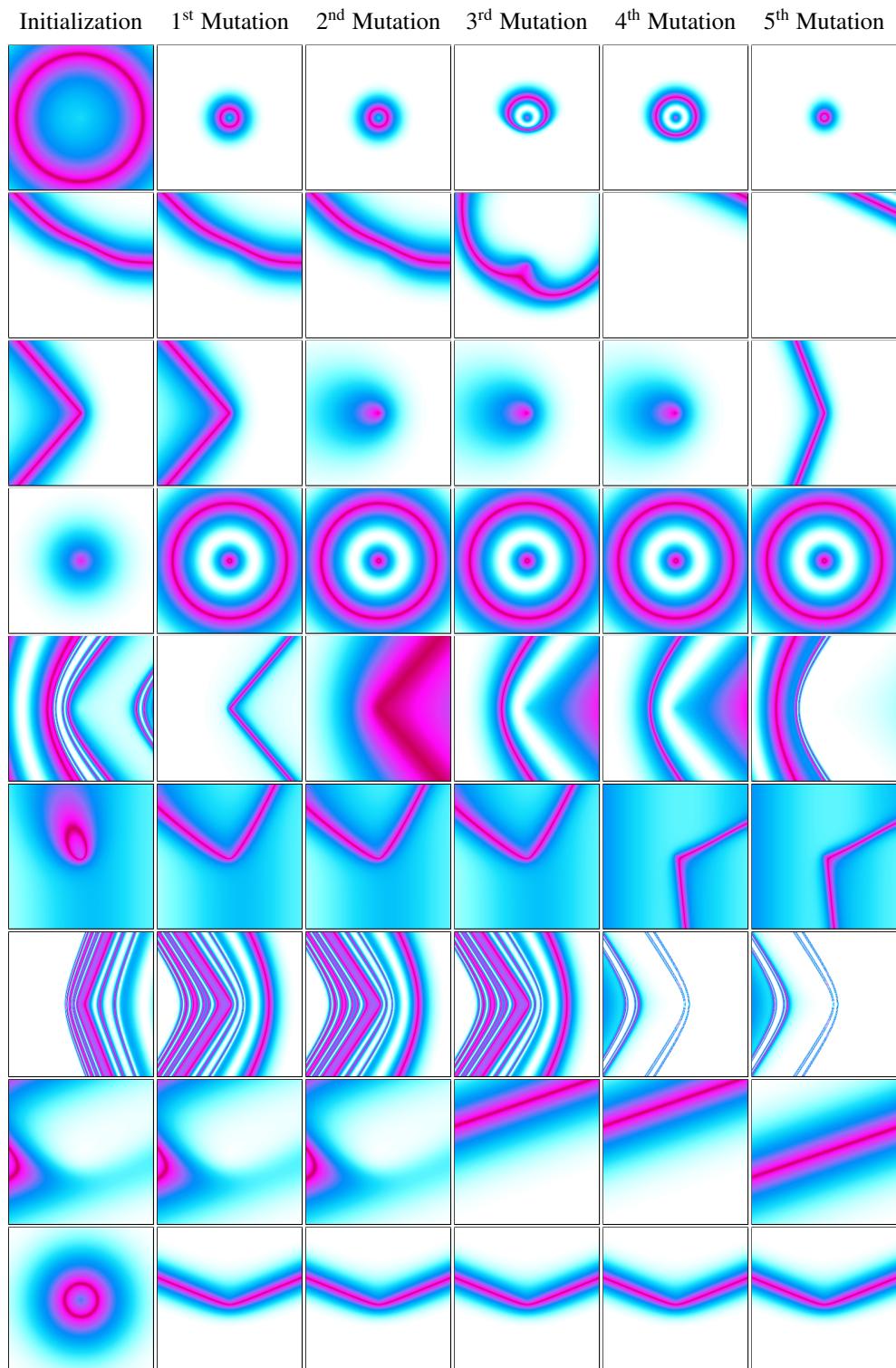


Figure 5: CPPNs can generate complex patterns via their random initialization and successive mutations. Each row shows generated patterns by one CPPN and its mutations.

206 **2.2 Sampling of Lenia's Dynamic Parameters**

207 The parameters that control the dynamics of Lenia ($R, T, \mu, \sigma, \beta_1, \beta_2, \beta_3$) are initialized and mutated
 208 via uniform and Gaussian distributions. Table 2 lists for each parameter the meta-parameters for their
 209 initialization and mutation. Each parameter is initialized by an uniform sampling $\theta_i \sim \mathcal{U}(a, b)$ with a
 210 and b as upper and lower border. An existing parameter θ_i is mutated by the following equation:

$$\theta_i \leftarrow [\theta_i + \mathcal{N}(0, \sigma_M)]_b^a ,$$

211 where σ_M is the mutation power and $[n]_b^a = \min(\max(n, a), b)$ is the clip function with a and b
 212 as upper and lower border. For natural numbers $\theta_i \in \mathbb{N}$ the resulting value is rounded towards the
 213 nearest natural number.

Parameter	Type	Value Range	Mutation σ_M
R	\mathbb{N}	$[2, 20]$	0.5
T	\mathbb{N}	$[1, 20]$	0.5
μ	\mathbb{R}	$[0, 1]$	0.05
σ	\mathbb{R}	$[0.001, 0.3]$	0.01
$\beta[1]$	\mathbb{R}	$[0, 1]$	0.05
$\beta[2]$	\mathbb{R}	$[0, 1]$	0.05
$\beta[3]$	\mathbb{R}	$[0, 1]$	0.05

Table 2: Settings for the initialization and mutation of Lenia system parameters θ .

214 **3 Measurement of Diversity in the Analytic Parameter and Behavior Space**

215 The algorithms are compared on their ability to explore a diverse set of patterns. The next section in-
 216 troduces the diversity measure, followed by sections that introduce the spaces in which the algorithms
 217 are compared.

218 **3.1 Diversity Measure**

219 Diversity is measured by the area that explored parameters cover in the parameter space of Lenia
 220 or that the identified patterns cover in the observation space. For the experiments the parameter
 221 space consisted of the initial start state of Lenia ($A^{t=1} \in [0, 1]^{256 \times 256}$) and the settings for Lenia's
 222 dynamics ($R, T, \mu, \sigma, \beta_1, \beta_2, \beta_3$). The space consists therefore of 256^2 dimensions, each for a single
 223 grid cell of the initial pattern, plus 7 dimensions for the dynamic settings. The observation space
 224 consists of the final patterns $A^{t=200} \in [0, 1]^{256 \times 256}$ resulting in 256^2 dimensions for the space. Each
 225 single exploration results in a new point in those spaces.

226 The diversity measures how much area the algorithms explored in those spaces (Fig. 6). The
 227 measurement is done by discretizing the space with a spatial grid and counting the number of
 228 discretized areas in which at least one point falls. For the discretization each dimension of the space is
 229 given a range, i.e. a minimum and maximum border. Each dimension is then split in a certain number
 230 of equally sized bins between those borders. The areas with values falling below the minimum or
 231 above the maximum border are counted as two additional bins.

232 The number of dimensions of the original parameter and observation space are too large to measure
 233 diversity in a meaningful manner. The initial pattern and the final pattern have 256^2 dimensions. We
 234 constructed therefore an analytic parameter and behavioral space where the latent representations of
 235 a β -VAE were used to reduce the high-dimensional patterns to 8 dimensions. The diversity in those
 236 spaces was compared between the algorithms. 5 bins (7 with the out of range values) per dimension
 237 were used for the discretization of those spaces for all experiments in the paper.

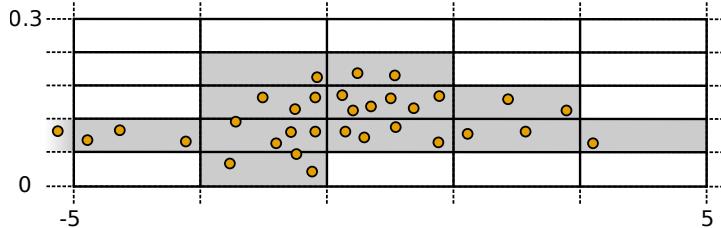


Figure 6: Illustration of the diversity measure in a two-dimensional space. The ranges for the dimensions were set to $[-5, 5]$ and $[0, 0.3]$. The number bins per dimension is 5. Including the outlier areas the number of discretized bins is $7^2 = 49$. The diversity is the number of bins in which points exist (grey areas) which are 12 in this example.

238 **3.2 Analytic Parameter Space**

239 The analytic parameter space was constructed by the 7 Lenia parameters that control its dynamics
 240 and 8 latent representation dimensions of a β -VAE (Table 3). The β -VAE was trained on initial
 241 patterns $A^{t=1}$ used during the experiments. The dataset was constructed by randomly selecting 42500
 242 patterns (37500 as training set, 5000 as validation set) from the experiments of all algorithms and
 243 each of their 10 repetitions. The β -VAE uses the same structure, hyper-parameters, loss function and
 244 learning algorithm as described in Section 5. It was trained for more than 1400 epochs with $\beta = 5$
 245 (Fig. 7). The encoder which resulted in the minimal validation set error during the training was used.
 246 According to its reconstructed patterns it can represent the general form of patterns but often not
 247 individual details such as their texture (Fig. 8).

248 **3.3 Analytic Behavior Space**

249 The analytic behavior space was constructed by combining the 5 statistical measures for final Lenia
 250 patterns (Section 1.3) and 8 latent representation dimensions of a β -VAE (Table 4). The β -VAE

Analytic Parameter Space Definition

Parameter	min	max	Parameter	min	max
R	1	20	β -VAE latent 1	-5	5
T	2	10	β -VAE latent 2	-5	5
μ	0	1	β -VAE latent 3	-5	5
σ	0	0.3	β -VAE latent 4	-5	5
β_1	0	1	β -VAE latent 5	-5	5
β_2	0	1	β -VAE latent 6	-5	5
β_3	0	1	β -VAE latent 7	-5	5
			β -VAE latent 8	-5	5

Table 3: Features of the analytic parameter space and their min and max values

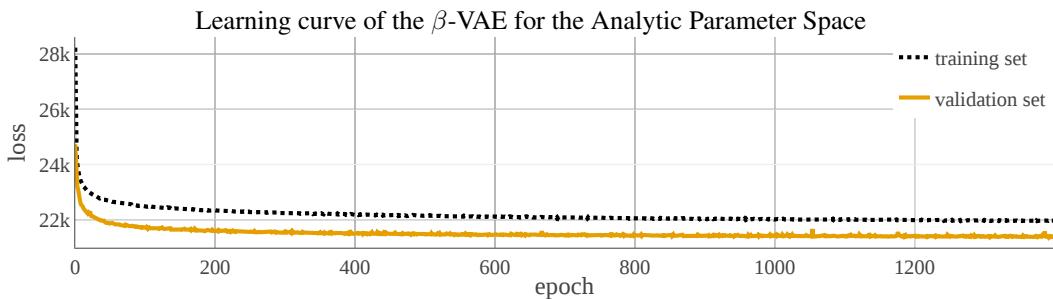


Figure 7: Learning curve of the β -VAE whose latent encoding was used for the analytic parameter space.

Reconstruction Examples of the Analytic Parameter Space β -VAE

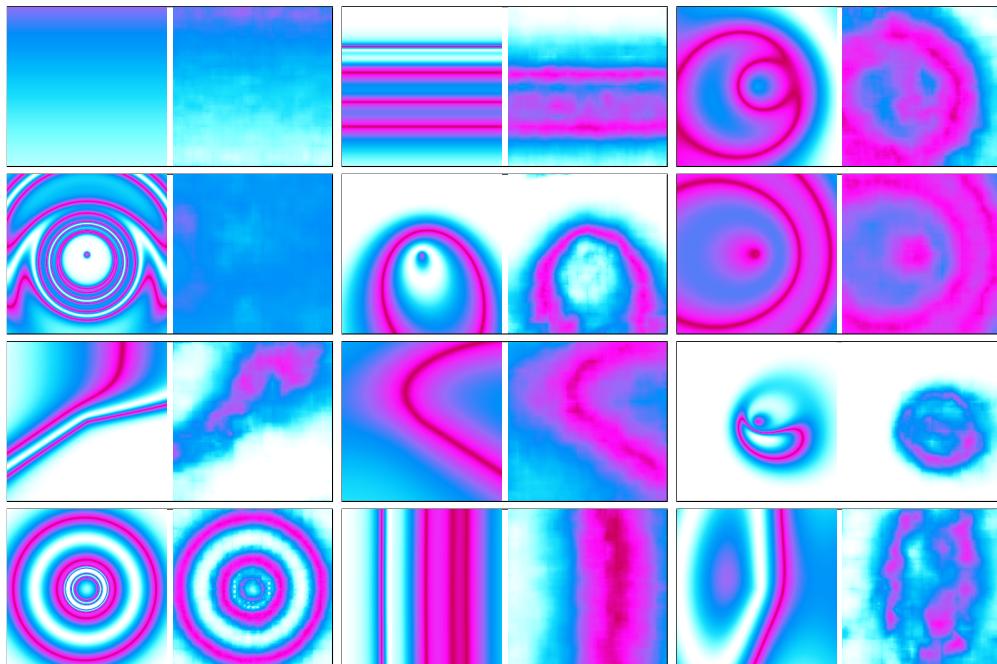


Figure 8: Examples of patterns (left) and their reconstructed output (right) by the β -VAE used for the construction of the analytic parameter space. The patterns are sampled from the validation dataset.

Analytic Parameter Space Definition					
Parameter	min	max	Parameter	min	max
mass M_A	0	1	β -VAE latent 1	-5	5
volume V_A	0	1	β -VAE latent 2	-5	5
density D_A	0	1	β -VAE latent 3	-5	5
asymmetry A_A	-1	1	β -VAE latent 4	-5	5
centeredness C_A	0	1	β -VAE latent 5	-5	5
			β -VAE latent 6	-5	5
			β -VAE latent 7	-5	5
			β -VAE latent 8	-5	5

Table 4: Features of the analytic behavior space and their min and max values

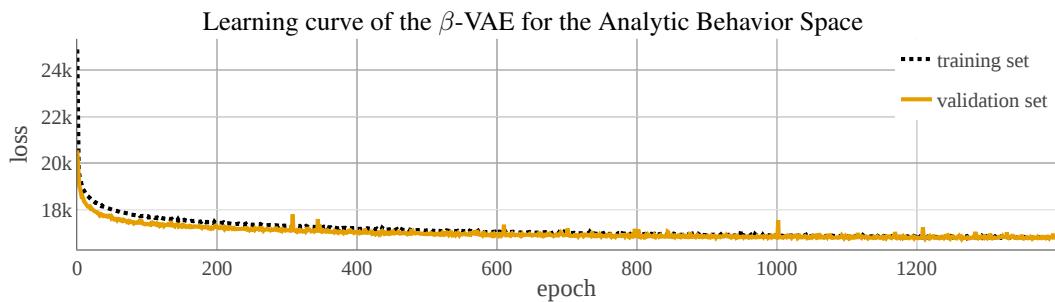


Figure 9: Learning curve of the β -VAE whose latent encoding was used for the analytic behavior space.

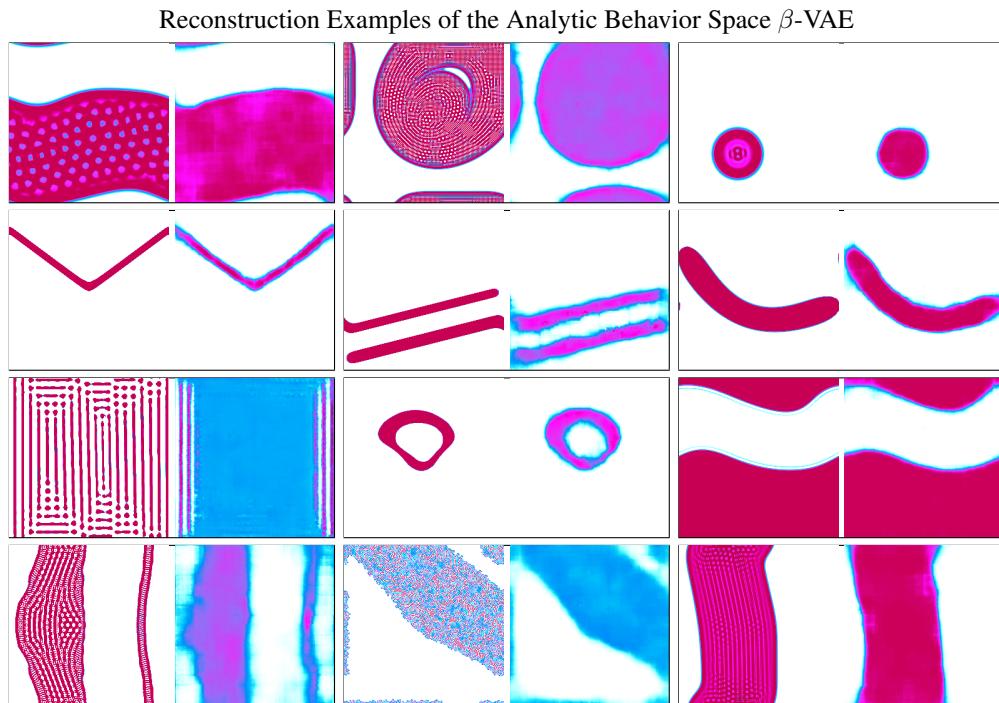


Figure 10: Examples of patterns (left) and their reconstructed output (right) by the β -VAE used for the construction of the analytic behavior space. The patterns are sampled from the validation dataset.

251 was trained on final patterns $A^{t=200}$ observed during experiments. The dataset was constructed by
252 randomly selecting 42500 patterns (37500 as training set, 5000 as validation set) from the experiments
253 of all algorithms and each of their 10 repetitions. The dataset consists of 50% animal and 50% non-
254 animal patterns. The β -VAE uses the same structure, hyper-parameters, loss function and learning
255 algorithm as described in Section 5. It was trained for more than 1400 epochs with $\beta = 5$ (Fig. 9).
256 The encoder which resulted in the minimal validation set error during the training was used. Its
257 reconstructed patterns show that it is able to represent the general form of patterns but often not
258 individual details such as their texture (Fig. 10).

259 **4 Random Exploration and IMGEPs with Hand-Defined Goal Spaces**

260 Two random explorations and several IMGEPs with different hand-defined goal spaces were evaluated
261 and compared. The main paper and the additional results in Section 6 only report the results for
262 the best random exploration and one IMGEP variant with a hand-defined goal space. This section
263 introduces the implementation details and diversity results of all evaluated random explorations and
264 IMGEPs with hand-defined goal spaces.

265 **4.1 Random Explorations**

266 We evaluated two random exploration strategies: Random Initialization and Random Mutation. The
267 main paper and the additional results in Section 6 only discuss the Random Initialization approach.

268 **Random Initialization:** This approach sampled for each of the 5000 explorations a random parameter
269 θ and a random CPPN to generate the initial state $A^{t=1}$. The approach can be replicated by using
270 Algorithm 1 with $N_{init} = 5000$.

271 **Random Mutation:** This approach is closer to the principle of IMGEPs. It first performs $N_{init} =$
272 1000 random explorations and adds each explored parameter θ to a history \mathcal{H} . Afterwards, it randomly
273 samples a parameter from the history and mutates it. The new parameter is also added to history
274 \mathcal{H} . The approach can be replicated by using Algorithm 1 where line 6 is skipped and the parameter
275 sampling distribution $\Pi(g, \mathcal{H})$ is selecting a random parameter from the history and mutating it.

276 **4.2 IMGEPs with Hand-Defined Goal Spaces**

277 We evaluated several IMGEP variants with goal spaces that were hand-defined (IMGEP-HGS). Each
278 space was constructed by a different combination of statistical measures of the final Lenia patterns
279 (Tables 5 and 6) which are described in Section 1.3. The main paper and the additional results in
280 Section 6 only discuss the IMGEP-HGS 9 approach. Algorithm 1 lists the steps of the IMGEP-HGS
281 variants. They begin with $N_{init} = 1000$ random explorations, followed by 4000 explorations based
282 on randomly generated goals. Each goal was sampled from a uniform distribution within the ranges
283 defined in Table 5. Then the parameter from a previous exploration that resulted in the closest
284 outcome to the current goal was mutated and explored.

Algorithm 1: IMGEP-HGS

```
1 Initialize goal space representation  $\mathcal{R}$  by hand-defined features
2 for  $i \leftarrow 1$  to  $N$  do
3   if  $i < N_{init}$  then           // Initial random iterations to populate  $\mathcal{H}$ 
4     Sample  $\theta \sim \mathcal{U}(\Theta)$ 
5   else                      // Intrinsically motivated iterations
6     Sample a goal  $g \sim \mathcal{G}(\mathcal{H})$  based on the space represented by  $\mathcal{R}$ 
7     Choose  $\theta \sim \Pi(g, \mathcal{H})$ 
8   Perform an experiment with  $\theta$  and observe  $o$ 
9   Append  $(o, \theta, \mathcal{R}(o))$  to the history  $\mathcal{H}$ 
```

285 **4.3 Results**

286 The random explorations and IMGEP-HGS variants are compared by their resulting diversity in the
287 analytic parameter and behavior space (Fig. 11). The diversity is measured by the number of reached
288 bins in each space using a binning of 7 bins per dimension.

289 The Random Initialization approach reached for all diversity measures a higher diversity than the
290 Random Mutation approach. Therefore, the Random Initialization approach is used for the comparison
291 to IMGEP approaches in the main paper and the additional results in Section 6.

292 Most IMGEP-HGS variants had a higher diversity in the analytic behavior space compared to random
293 explorations, although their diversity in the analytic parameter space is lower. This shows the
294 advantage of IMGEPs over random searches in discovering a wider range of patterns in the target

295 system. The best overall diversity had IMGEP-HGS 3, 4 and 9. We chose IMGEP-HGS 9 to compare
 296 it with learned goal spaces in the main paper and for the additional results in Section 6. It identified
 297 the highest diversity of non-animals of the three variants (3, 4, 9) reaching a higher diversity for
 298 non-animals than any IMGEP with a learned goal space. It was therefore selected to show that the
 299 choice of the goal space has an influence on the patterns that IMGEPS identify.

300 Depending on the statistical measures used to define the goal space the diversity between the IMGEP-
 301 HGS variants varied. IMGEPs that use the volume measure (HGS 1 - 4) reach in general a higher
 302 overall diversity which can be attributed to their higher diversity of animal patterns than goal spaces
 303 with the density measure (HGS 5 - 8) (Fig. 11, b, c). In terms of diversity of identified animals
 304 showed the inclusion of several measures the best performance (HGS 4 and HGS 8 in Fig. 11, c) In
 305 terms of diversity of identified non-animals showed the inclusion of several measures besides the
 306 centeredness C_A measure the best performance (HGS 3 and HGS 7 in Fig. 11, d). The results show
 307 that the choice of the goal space has an important influence on the diversity of identified patterns and
 308 their type (animal or non-animal).

Feature	min	max
mass M_A	0	1
volume V_A	0	1
density D_A	0	1
asymmetry A_A	-1	1
centeredness C_A	0	1

Table 5: HGS Goal Space Ranges

Feature	HGS-Variants								
	1	2	3	4	5	6	7	8	9
mass M_A	×	×	×	×	×	×	×	×	×
volume V_A	×	×	×	×					×
density D_A					×	×	×	×	×
centeredness C_A		×		×		×		×	×
asymmetry A_A			×	×			×	×	×

Table 6: IMGEP-HGS Variants

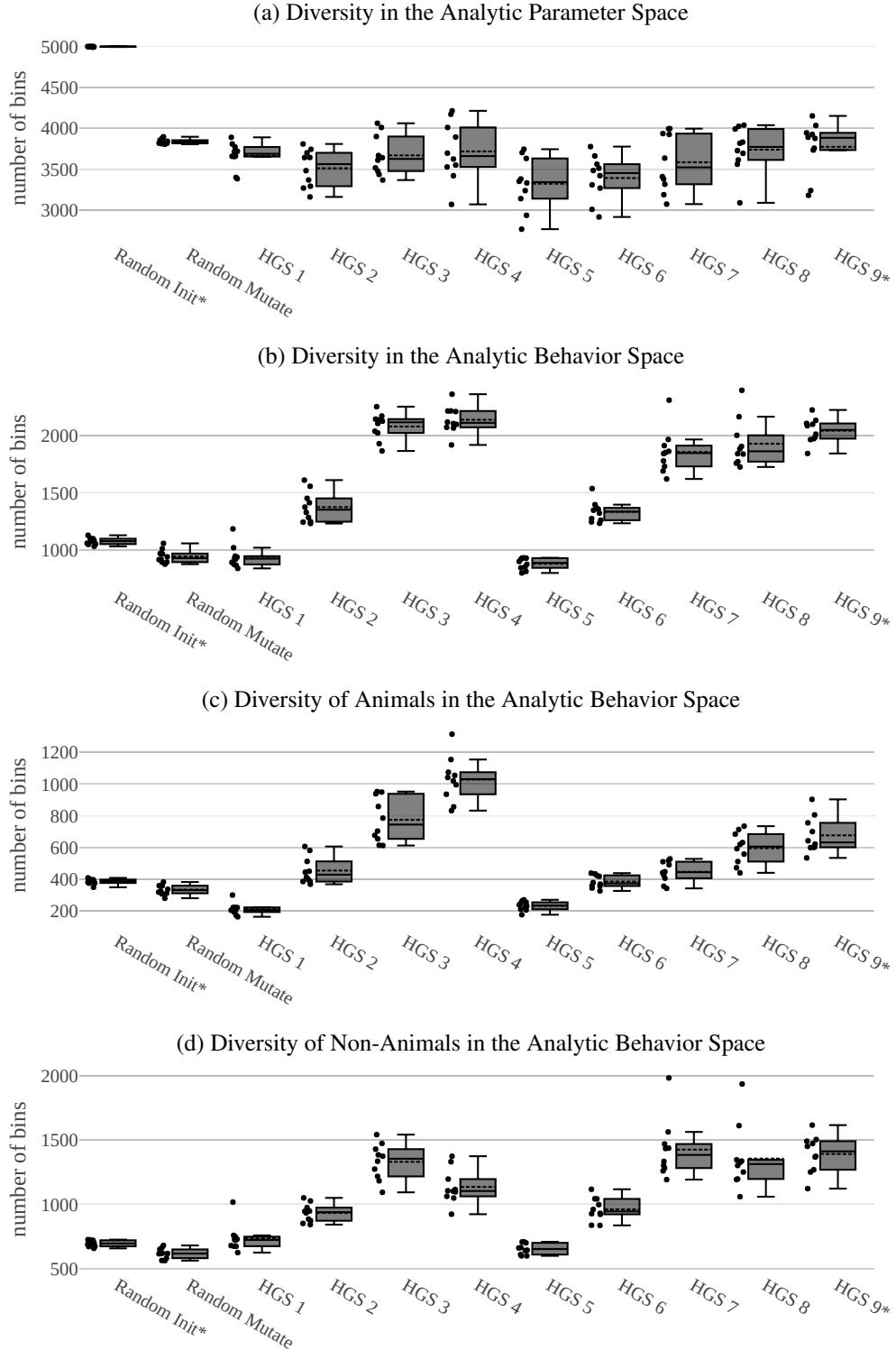


Figure 11: Although all IMGEP-HGS variants have lower diversity in the analytic parameter space compared to the Random Initialization approach, most of them have a higher diversity in the analytic behavior space. Each dot besides the boxplot shows the diversity of found patterns for each repetition ($n = 10$). The box ranges from the upper to the lower quartile. The whiskers represent the upper and lower fence. The mean is indicated by the dashed line and the median by the solid line.

309 **5 IMGEPS with Learned Goal Spaces via Deep Variational Autoencoders**

310 For the IMGEPS using learned goal spaces (IMGEP-PGL and IMGEP-OGL) several deep VAEs
 311 were evaluated and compared. VAEs are artificial neural networks that learn a latent representation of
 312 data. The latent representation has a reduced number of dimensions and should capture the important
 313 features of the input data. We use VAEs to learn the important features that describe Lenia patterns.
 314 The features are then used to define goal spaces for IMGEPS. Section 5.1 explains the different
 315 variants that were implemented and Section 5.2 describes their implementation details.

316 **5.1 VAE Variants**

317 Variational Autoencoders (VAE) [10] have two components: a neural *encoder* and *decoder*. The
 318 encoder $q(\mathbf{z}|\mathbf{x}, \chi)$ represents a given data point \mathbf{x} in a latent representation \mathbf{z} . In variational ap-
 319 proaches the encoder describes a data point by a representative distribution in the latent space of
 320 reduced dimension d . A standard Gaussian prior $p(\mathbf{z}) = \mathcal{N}(0, I)$ and a diagonal Gaussian posterior
 321 $q(\mathbf{z}|\mathbf{x}, \chi) = \mathcal{N}(\mu, \sigma)$ is used for this purpose. Given a data point x , the encoder outputs the mean μ
 322 and variance σ of the representative distribution in the latent space. The decoder $p(\mathbf{x}|\mathbf{z}, \psi)$ tries to
 323 reconstruct the original data x from a sampled latent representation \mathbf{z} for the distribution given by
 324 the encoder. Under these assumptions, training is done by maximizing the computationally tractable
 325 evidence lower bound (with $\beta = 1$):

$$\mathcal{L}(\chi, \psi) = \underbrace{\mathbb{E}_{\mathbf{z} \sim q_\chi(\mathbf{z}|\mathbf{x})} [\log p_\psi(\mathbf{x}|\mathbf{z})]}_a - \beta \times \underbrace{\mathbb{D}_{KL}[q_\chi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})]}_b. \quad (5)$$

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

$$b = \mathbb{D}_{KL}[\mathcal{N}(\mu(\mathbf{x}), \Sigma(\mathbf{x})) \| \mathcal{N}(0, I)] = \sum_{i=1}^d \underbrace{\mathbb{D}_{KL}[\mathcal{N}(\mu(\mathbf{x})_i, \sigma(\mathbf{x})_i) \| \mathcal{N}(0, 1)]}_{b_i}. \quad (6)$$

328 Many current state-of-the-art approaches [3, 6, 8, 12] build on the VAE framework and augment the
 329 VAE objective to enhance interpretability and disentanglement of the latent variables. In this paper,
 330 we couple the VAE architecture with three different objectives: the classical VAE objective [10]
 331 (Equation (5) with $\beta = 1$), the β -VAE objective [6] (Equation (5) with $\beta > 1$) and an augmented
 332 β -VAE objective (Equation (7)).

333 The β -VAE objective re-weights the b term by a factor $\beta > 1$, aiming to enhance the disentangling
 334 properties of the learned latent factors. We are interested in such properties as it has been shown that
 335 it can benefit exploration [13]. However, heavily penalizing b can result in the network learning to
 336 “sacrifice” one or more of the learned latent variables in order to nullify their contribution b_i to this
 337 term (6). Those dimensions become completely uninformative and useless for further exploration in
 338 the learned latent space.

339 To prevent this phenomenon to happen, we augmented b with a new term that encourages the network
 340 to decrease *together* the individual contributions b_i of the different latent variables. This augmented
 341 loss term not only minimizes the averaged contribution (sum) but also the variance of the individual
 342 contributions:

$$b_{aug} = \sum_{i=1}^d b_i + \text{Var}([b_1, \dots, b_d]). \quad (7)$$

343 Prior work also reports this phenomenon [1, 4, 5, 11, 17, 19] and other modifications of the training
 344 objective have been proposed [17, 19].

345 We tested the standard VAE, β -VAE and the augmented objective in the context of IMGEP to learn
 346 their goal spaces. The VAE networks were trained on the loss function given in Equation (8) with the
 347 hyper-parameters $\{\beta = 1, \gamma = 0\}$, $\{\beta = 5, \gamma = 0\}$ and $\{\beta = 5, \gamma = 1\}$ corresponding to the three
 348 variants outlined above.

$$L = -a + \beta(b + \gamma \times c) \quad (8)$$

349 In the main paper paper, the results shown for the IMGEP-PGL and IMGEP-OGL approaches have
 350 been obtained with representations trained using the second variant (β -VAE objective). Its results
 351 were slightly better compared to the other VAE approaches, even though the results do not exhibit
 352 significant trends promoting one method over the other.

353 **5.2 Implementation Details**

354 This section describes the IMGEPE approaches and the network architecture, training procedure,
 355 hyper-parameters and datasets for the training of their VAEs.

356 All VAEs use the same architecture (Table 7). The encoder network has as input the Lenia pattern
 357 and as outputs for each latent variable \mathbf{z}_i the mean μ_i and log-variance $\log(\sigma_i^2)$. The decoder takes as
 358 input during the training for each latent variable a sampled value $z_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. For validation runs
 359 and the generation of all reconstructed patterns shown in figures the decoder takes the mean $z_i = \mu_i$
 360 as input. Its output is the reconstructed pattern.

361 The training objectives of all three variants are given in section 5.1. The resulting loss function (Eq. 8)
 362 of all VAE variants for a batch is:

$$\text{Loss}(x, \hat{x}, \mu, \sigma) = -a + \beta \left(\sum_{i=1}^d b_i + \gamma \text{Var}([b_1, \dots, b_d]) \right),$$

363 where x are the input patterns, \hat{x} are the reconstructed patterns, μ, σ are the outputs of the decoder
 364 network and d is the number of latent dimensions. The reconstruction accuracy part a of the loss is
 365 given by a binary cross entropy with logits:

$$a = \frac{1}{N} \sum_{n=1}^N \sum_{j=1}^{L^2} (x_{j,n} \cdot \log \sigma(\hat{x}_{j,n}) + (1 - x_{j,n}) \cdot \log(1 - \sigma(\hat{x}_{j,n}))),$$

366 where the index j is for the single cells (pixel) of the pattern and n for the datapoint in the current
 367 batch, N is the batch size and $\sigma(x) = \frac{1}{1+e^{-x}}$. The KL divergence terms b_i are given by:

$$b_i = \frac{1}{2 \cdot N} \sum_{n=1}^N (\sigma_{i,n}^2 + \mu_{i,n}^2 - \log(\sigma_{i,n}^2) - 1).$$

368 All VAEs were trained for 2000 epochs. We used the Adam optimizer [9] ($lr = 1e-3$, $\beta_1 = 0.9$,
 369 $\beta_2 = 0.999$, $\epsilon = 1e-8$, weight decay= $1e-5$) with a batch size of 64.

370 The patterns from the datasets were augmented by random x and y translations (up to half the pattern
 371 size and with probability 0.3), rotation (up to 40 degrees and with probability 0.3), horizontal and
 372 vertical flipping (with probability 0.2). The translations and rotations were preceded by spherical
 373 padding to preserve Lenia spherical continuity.

374 Two types of IMGEPEs were evaluated:

375 **IMGEPE-PGL (prelearned goal space):** IMGEPE (Algorithm 2) with a goal space defined by a VAE
 376 that was trained before the exploration starts. The VAE is trained on a dataset with precollected Lenia
 377 patterns. The best VAE model obtained during the training phase, i.e. the one with the highest
 378 accuracy on the validation data, is used for the exploration.

379 The dataset used to train the VAE has 558 patterns which are distributed into a training (75%),
 380 validation (10%) and testing (15%) datasets. Half of the patterns (279) were manually identified
 381 animal patterns by [2] (Fig. 1). The other half (279) are randomly initialized CPPN patterns as
 382 described in Section 2.1 (Fig. 5).

Encoder	Decoder
Input pattern A: $256 \times 256 \times 1$	Input latent vector $z: 8 \times 1$
Conv layer: 32 kernels 4×4 , stride 2, 1-padding + ReLU	FC layers : $256 + \text{ReLU}$, $16 \times 16 \times 32 + \text{ReLU}$
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 : $256 + \text{ReLU}$, $256 + \text{ReLU}$, FC: 2×8	TransposeConv layer: 32 kernels 4×4 , stride 2, 1-padding

Table 7: VAE architecture for the pretrained and online experiments.

Algorithm 2: IMGEP-PGL

```
1 Initialize goal space representation  $\mathcal{R}$  with the pretrained VAE
2 for  $i \leftarrow 1$  to  $N$  do
3   if  $i < N_{init}$  then           // Initial random iterations to populate  $\mathcal{H}$ 
4     Sample  $\theta \sim \mathcal{U}(\Theta)$ 
5   else                         // Intrinsically motivated iterations
6     Sample a goal  $g \sim \mathcal{G}(\mathcal{H})$  based on space represented by  $\mathcal{R}$ 
7     Choose  $\theta \sim \Pi(g, \mathcal{H})$ 
8   Perform an experiment with  $\theta$  and observe  $o$ 
9   Append  $(o, \theta, \mathcal{R}(o))$  to the history  $\mathcal{H}$ 
```

383 **IMGEP-OGL (online learned goal space):** IMGEP (Algorithm 1 in the main paper) that trains
384 the VAE which defines the goal space during the exploration. The VAE is trained on Lenia patterns
385 discovered by the algorithm. Every $K = 100$ explorations the VAE model is trained for 40 epochs
386 resulting in 2000 epochs in total (less if there is not enough data after the first T runs to start the
387 training).

388 Importance sampling is used to give the patterns in the training dataset a different weight during
389 the training. A weighted random sampler is used that samples newly discovered patterns from the
390 training dataset half of the time. Each pattern that has been added to the training dataset during the
391 last period of 100 explorations has a probability of $\frac{0.5}{N}$ to be sampled (N is the total number of new
392 patterns in the dataset). Older patterns are also sampled half of the time each one with probability
393 $\frac{0.5}{|D_T| - N}$. As a result, newer discovered patterns have a higher weight and a stronger influence on the
394 training of the VAE model.

395 The datasets were constructed incrementally during the exploration by gathering non-dead patterns.
396 One pattern every ten is added to the validation set (10%) and the rest is used in the training set. At
397 the initial period of training, the training dataset amounts approximately 50 patterns and at the last
398 period of training the dataset amounts approximately 3425 patterns (Fig. 12). The validation dataset
399 only serves for checking purposes and has no influence on the learned goal space.

400 **5.3 Results**

401 We compared the different IMGEP variants (PGL and OGL) and VAEs (VAE, β -VAE and augmented
402 β -VAE) with each other on the basis of the diversity of their identified patterns. Furthermore, the
403 pattern reconstruction ability of the VAEs is analyzed.

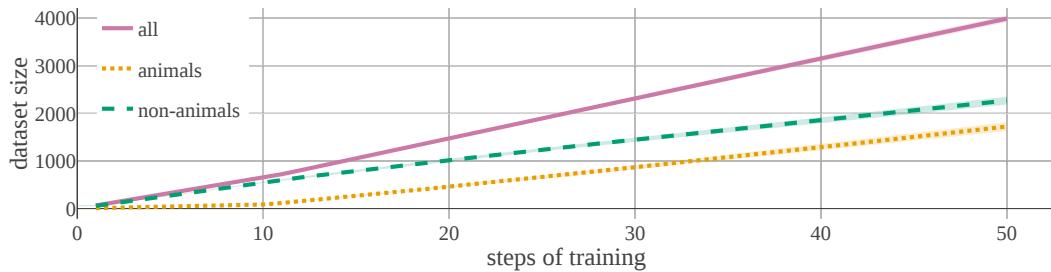


Figure 12: The IMGEP-OGL collects during the exploration animal and non-animal patterns to add them to its dataset for the training of the VAE. The figure shows the development of the averaged dataset size over all repetitions ($n = 10$) of the IMGEP-OGL algorithm with a β -VAE. Standard deviation is depicted as a shaded area but for some not visible because it is too small.

404 **5.3.1 Diversity**

405 The algorithms are compared by their diversity in the analytic parameter and behavior space (Sec-
406 tion 3). Diversity is measured by the number of discretized bins that were explored by the algorithms
407 in each space if each dimension of the space is separated in 7 bins.

408 All IMGEPS and VAE variants reached a higher diversity in the analytic behavior space compared to
409 random explorations (Fig. 13, b), although random explorations have a higher diversity in the analytic
410 parameter space (Fig. 13, a). This result confirms further the advantage of IMGEPS over random
411 explorations in terms of identifying diverse patterns.

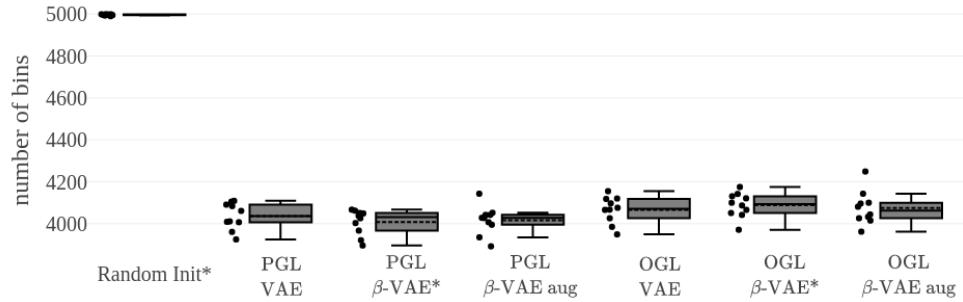
412 The difference between the PGL and OGL variants were small for all diversity measures. The OGL
413 showed a slight advantage over the PGL versions in all diversity measures. Thus, an online version of
414 the IMGEPS can learn an appropriate goal space during the exploration. A precollected dataset as for
415 the PGL is not necessary to successfully use IMGEPS.

416 The difference between the VAE variants (VAE, β -VAE and augmented β -VAE) was very small. The
417 β -VAE was slightly better than the other two variants for the diversity in the analytic parameter space
418 and for both IMGEPS variants. All VAEs seemed to learn similar features for our datasets. It might be
419 possible that the different VAE variants show different behaviors if their parameters are fine-tuned,
420 such as the β parameter, but this was out of the scope of this paper. Because the β -VAE performed
421 slightly the best, it was used for the results in the main paper and in Section 6.

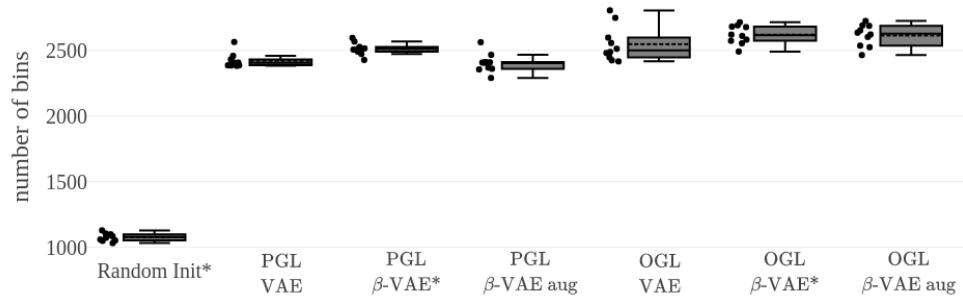
422 **5.3.2 VAE Pattern Reconstruction**

423 All VAE variants showed a similar learning performance on the precollected dataset and the online
424 collected dataset (Fig. 14 and 16). Their ability to reconstruct patterns based on the encoded latent
425 representation is also qualitatively similar. For both datasets are the VAEs able to learn the general
426 form of the activity pattern (Fig. 15 and 17). Nonetheless, the compression of the images to a 8-
427 dimensional vector results in a general blurriness in the reconstructed patterns. As a result, the VAEs
428 are not able to encode finer details and textures of patterns (Fig. 18). We believe this is the reason
429 for their ability to identify more animals compared to the random exploration or the IMGEPS-HGS.
430 Different animals have often a different form, whereas non-animals span often over the whole area of
431 Lenia's grid and differentiate mainly in their textures and small details. Because the VAE seem to
432 encode more the general form a goal space based on them is more appropriate to find patterns with
433 different forms such as the animals and not different textures which are important for non-animals.

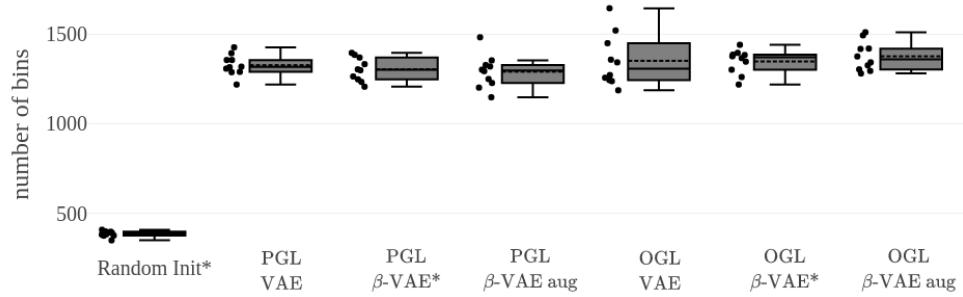
(a) Diversity in the Analytic Parameter Space



(b) Diversity in the Analytic Behavior Space



(c) Diversity of Animals in the Analytic Behavior Space



(d) Diversity of Non-Animals in the Analytic Behavior Space

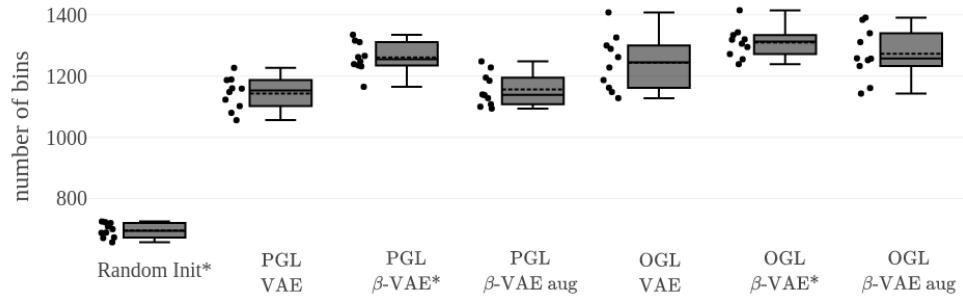


Figure 13: The different VAE algorithms showed only small differences in terms of diversity. The β -VAE had a slightly better diversity for the analytic behavior space for both IMGEV variants (PGL and OGL). Each dot besides the boxplot shows the diversity of found patterns for each repetition ($n = 10$). The box ranges from the upper to the lower quartile. The whiskers represent the upper and lower fence. The mean is indicated by the dashed line and the median by the solid line.

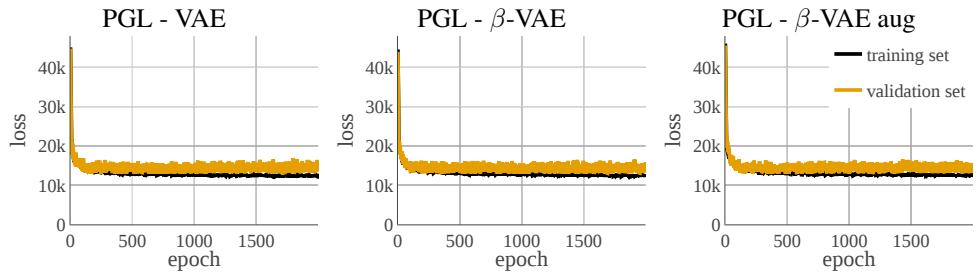


Figure 14: Averaged learning curves ($n = 10$) of the VAEs for the IMGEP-PGL experiments.

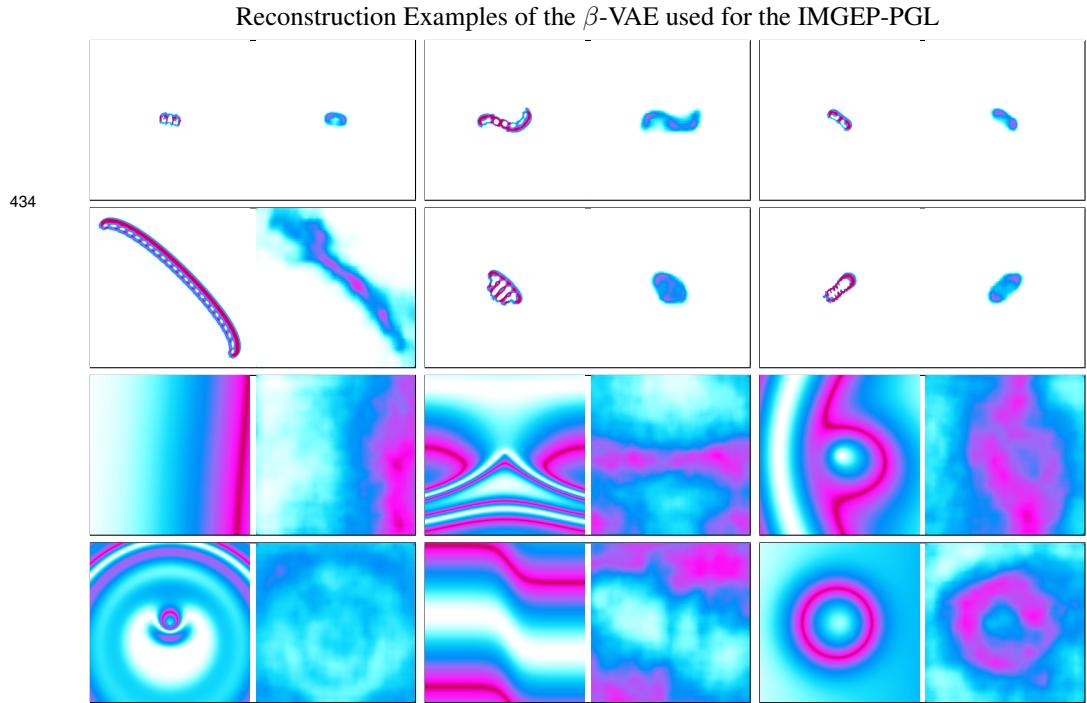


Figure 15: Examples of patterns (left) and their reconstructed output (right) by a VAE network used for the IMGEP-PGL. The patterns are sampled from its validation dataset. The dataset is composed of half animal patterns (rows 1 and 2) and half randomly generated CPPN patterns (rows 3 and 4).

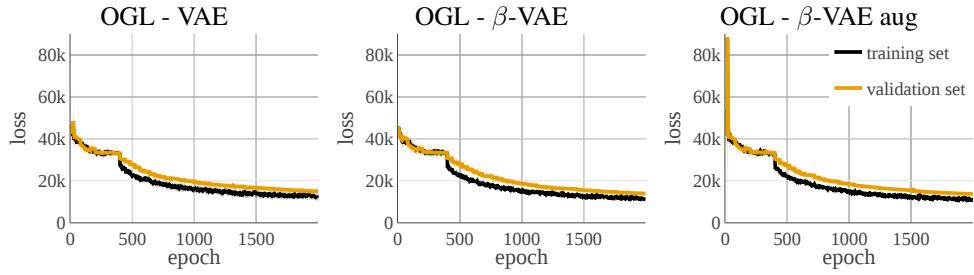


Figure 16: Averaged learning curves ($n = 10$) of the VAEs for the IMGEP-OGL experiments.

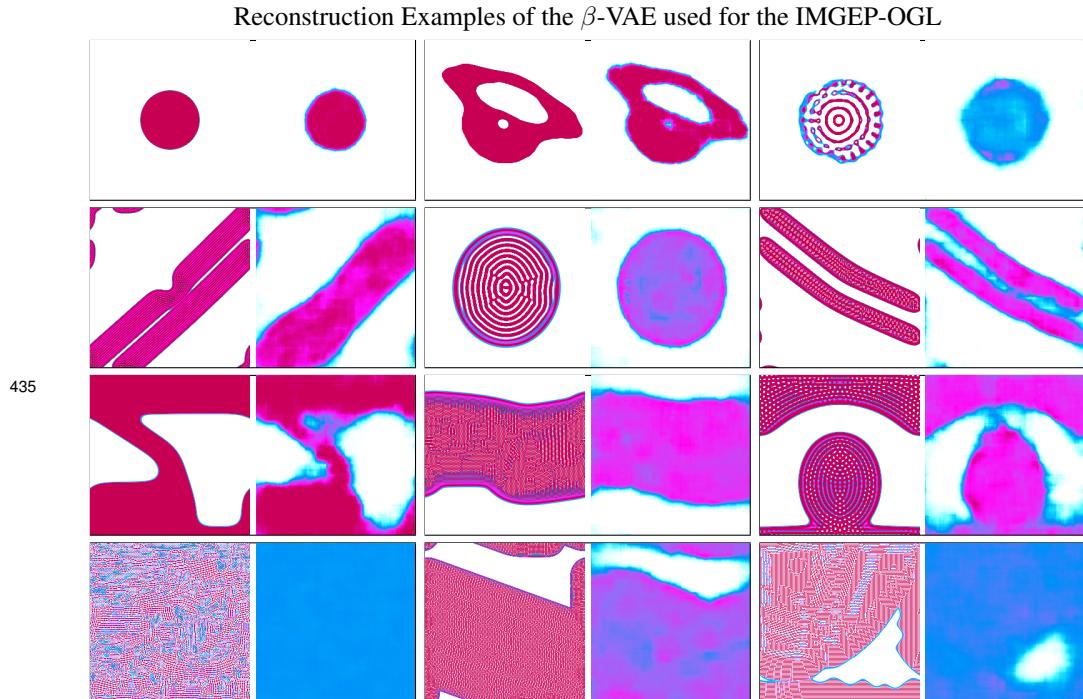


Figure 17: Examples of patterns (left) and their reconstructed output (right) by a VAE network used for the IMGEP-OGL. The patterns are sampled from its validation dataset. Animal patterns (rows 1 and 2) and non-animal patterns (rows 3 and 4) are shown.

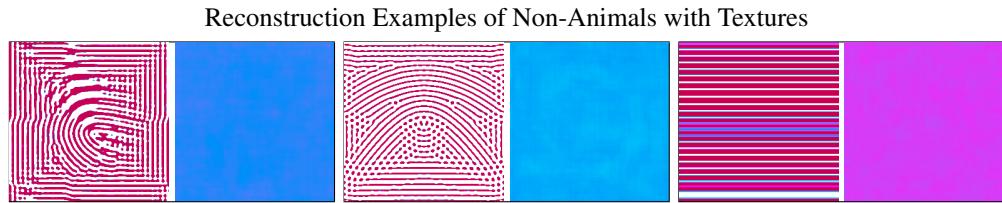


Figure 18: Examples of “textured” patterns that the VAE networks is unable to reconstruct. While a human eye can differentiate the input patterns (spatial frequency, orientation, etc.) the VAE reconstructs the three images identically.

436 6 Additional Results

437 This section lists additional results and analysis for the experimental results in the main paper. The
 438 results are only for a subset of all algorithm variants that have been evaluated. The results correspond
 439 to the following algorithms: Random to Random Initialization (Section 4), IMGEP-HGS to IMGEP-
 440 HGS 9 (Section 4), IMGEP-PGL to IMGEP-PGL with a β -VAE (Section 5) and IMGEP-OGL to
 441 IMGEP-OGL with a β -VAE (Section 5).

442 6.1 Number of Identified Patterns

443 The main paper used the measure of diversity of the found patterns per algorithm to compare their
 444 performance. Another measure to compare the algorithms is the number of the patterns they identified
 445 for each of the three pattern classes: dead, animals, non-animals (Fig. 19).

446 The results deviate slightly from the diversity measures. In terms of identified non-dead patterns
 447 outperform all IMGEP approaches a random exploration by finding between 10 to 20% more patterns.
 448 Although the IMGEP-HGS finds more non-dead patterns than the IMGEPs with learned goal spaces
 449 (OGL, PGL) its overall diversity in the analytic behavior space is smaller (Fig. 3, b of the main paper).

450 In the case of animal patterns, all IMGEP approaches outperform the random exploration (8%).
 451 Within the IMGEP approaches the online learned goal space approach (IMGEP-OGL, 34%) and the
 452 pretrained goal space approach (IMGEP-PGL: 35%) find a similar amount. The hand-defined goal
 453 space approach identified less animal patterns (IMGEP-HGS: 19%). For non-animal patterns the
 454 hand-defined goal space approach identifies most patterns (IMGEP-PGL: 67%), followed by the
 455 random exploration (56%) and both learned goal space approaches (IMGEP-OGL: 45%, IMGEP-
 456 PGL: 43%). Although the number of identified non-animal patterns for the learned goal space
 457 approaches is low, their diversity is higher than for a random exploration and only slightly lower than
 458 for the hand-defined goal space approach (Fig. 3, d of the main paper).

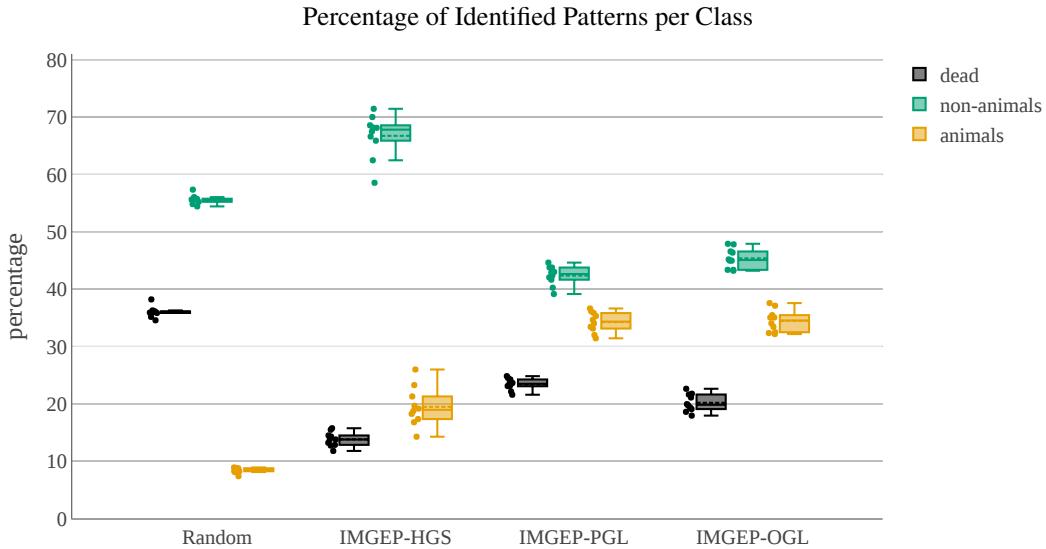


Figure 19: IMGEPs found more non-dead patterns compared to the random exploration. In terms of animals, the learned goal space approaches (IMGEP-PGL and OGL) found most animal patterns. For non-animals, the hand-defined goal space (IMGEP-HGS) found most patterns. The plot illustrates the percentage of found patterns for each class. Each dot besides the boxplot shows the percentage of found patterns for each repetition ($n = 10$). The box ranges from the upper to the lower quartile. The whiskers represent the upper and lower fence. The mean is indicated by the dashed line and the median by the solid line.

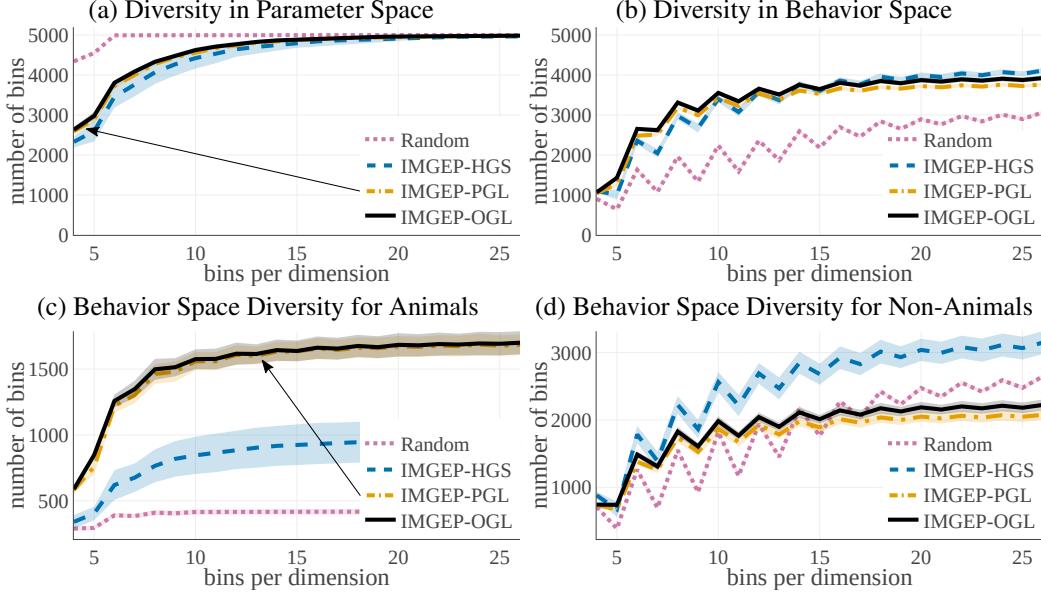


Figure 20: Dependencies of the diversity measure on the number of bins per dimensions for (a) the analytic parameter and (b-d) the behavior space. Depicted is the average diversity ($n = 10$) with the standard deviation as shaded area (for some not visible because it is too small).

459 6.2 Dependence of the Diversity Measure on the Number of Bins per Dimension

460 The diversity of identified patterns measures the spread of the area which the identified patterns cover
 461 in the analytic behavior space (Fig. 3 in the main paper). The measure is defined by dividing the
 462 space in a number of discrete areas or bins (Section 3.1). The diversity is then measured by how
 463 many bins are covered during an exploration. The bins are created by dividing each dimension of
 464 the space into a number of equally-sized bins. We analyzed how the number of bins per dimension
 465 influences the diversity measure (Fig. 20).

466 Although the diversity difference between the algorithms depends on the number of bins per dimension
 467 for each space, the order of the algorithms, i.e. which algorithm has a higher diversity, is generally
 468 constant. Only if the number of bins per dimension grows large (> 10) the order of the algorithms
 469 changes for some spaces and subpatterns. The order starts to follow the order seen for the number
 470 of identified patterns (compare the diversity with 25 bins per dimension in Fig. 20 with the number
 471 of identified patterns in Fig. 19). In this case the discretization of the space becomes too fine and
 472 each pattern falls into its own discretized area. We chose therefore a smaller number of bins per
 473 dimension of 7 (including the out of border bins) for all other diversity plots in the main paper and
 474 the Supplementary Material to compare the algorithms in a meaningful way.

475 6.3 Dimension Reduction of the Analytic Parameter and Behavior Space

476 A two-dimensional reduction of the identified patterns in the analytic parameter and behavior space
 477 (Section 3) visualizes the diversity of the parameters and identified patterns. The dimension reduction
 478 of the parameter space is based on all explored parameters encoded in the analytic parameter space
 479 from the first repetition experiment of all 4 algorithms. All encoded points were normalized so that
 480 the overall minimum value became 0 and the maximum value 1 for each dimension. Afterwards a
 481 principle component analysis (PCA) was performed to detect the 2 principle components [7]. The
 482 found patterns for each algorithm are plotted according to those components.

483 The results show that the random exploration has a stronger uniform distribution than any of the
 484 IMGEPE algorithms in the analytic parameter space (Fig. 21). The IMGEPE algorithms show concen-
 485 trations of explorations in specific regions of the parameter space. The visualization shows also that
 486 it is not possible to define distinct regions in the parameter space that allow to differentiate between
 487 dead, animal and non-animal patterns.

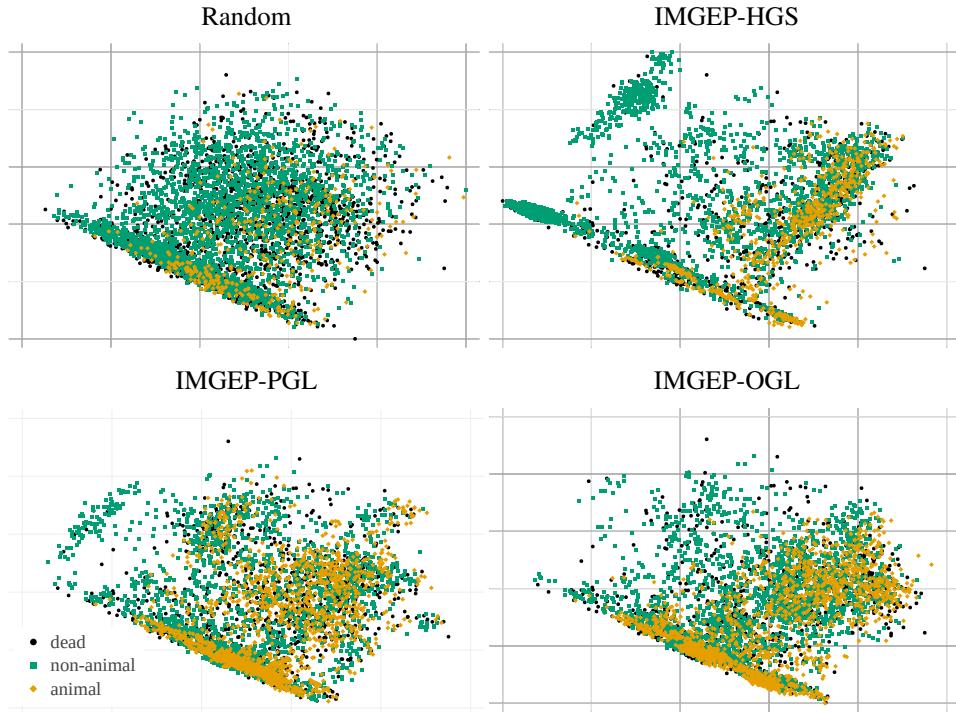


Figure 21: A random exploration covers the analytic parameter space more uniformly than IMGEP algorithms which form clusters at certain areas. PCA dimension reduction of the analytic parameter space which illustrates all explored parameters by the first repetition experiment per algorithm.

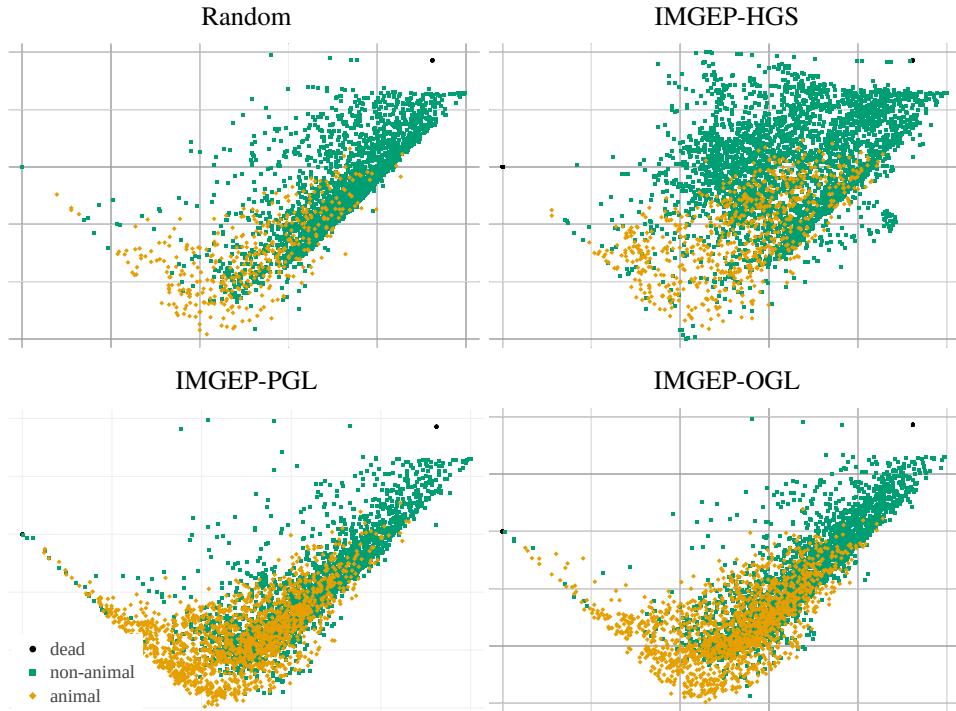


Figure 22: In the analytic behavior space IMGEPs reach a higher diversity compared to a random exploration. The HGS approach explores more non-animal areas and the PGL and OGL more animal areas. PCA dimension reduction of the analytic behavior space which illustrates all identified patterns by the first repetition experiment per algorithm.

488 The same analysis was performed for the identified patterns of each algorithm encoded in the analytic
489 behavior space (Fig. 22). It is visible that the random exploration is more concentrated compared
490 to the IMGEP approaches, especially in a region with many non-animal patterns (north-east). The
491 IMGEP-HGS has a wider spread in the non-animal area. The IMGEPs with a learned goal space
492 (PGL and OGL) show a stronger distribution in an area that encodes mostly animals (south).

493 **6.4 Identified Patterns**

494 Fig. 23, 24, 25 and 26 illustrate examples of identified pattern per class (animal, non-animal, dead)
495 and their ratio for the random exploration, IMGEP-HGS, IMGEP-PGL and IMGEP-OGL. The
496 patterns have been randomly sampled from the results of the first exploration repetition experiment
497 of each algorithm.

498 **6.5 Visualization of Goal Spaces**

499 The goal space of IMGEPs is their most important element because it defines which type of patterns
500 are set as goals for the exploration. This section provides complementary material for the analysis
501 made in Section 5.2 of the main paper and shows further visualizations of goal spaces. The goal
502 spaces of all IMGEP algorithms are visualized via a two-dimensional reduction of each goal space.
503 Two techniques for dimensionality reduction were applied: PCA [7] and t-Distributed Stochastic
504 Neighbor Embedding (t-SNE) [14].

505 The visualization was constructed by using for each exploration algorithm its goal space representa-
506 tions of all patterns it explored from a single repetition experiment. All goal representations were
507 normalized so that the overall minimum value became 0 and the maximum value 1 for each goal
508 space dimension. Afterwards the PCA was performed to detect the 2 principle components. T-SNE
509 was executed by using the default standard Euclidean distance metric and default hyper-parameters
510 (perplexity set to 50).

511 The resulting two-dimensional visualizations of the goal spaces make the differences between the
512 algorithms visible (Fig. 27). For both aproaches (PCA, t-SNE) has the hand-defined goal space (HGS)
513 only a small area and a few clusters for animal patterns. In contrast, the learned goal spaces based
514 on β -VAEs (PGL and OGL) have larger areas and more clusters for animal patterns. As a result,
515 the learned goal spaces explore more animal patterns and find a higher diversity of them (Fig. 3, c)
516 compared to the hand-defined goal space. The reason for this effect seems to be that the β -VAE
517 which defines the goal space for the PGL and OGL is learning to represent the shape of patterns. The
518 shape is an important feature of animals. Whereas, non-animals often cover the whole Lenia grid and
519 differ mainly in their textures which the β -VAE does not represent well (Section 5).

520 The visualization serves as a support in qualitatively evaluating and comparing the efficiency of each
521 algorithm in extracting a diversity of patterns from the data. Integrated into an interactive interface,
522 these graphs are also useful for a potential human end-user to easily explore and visualize the different
523 type of found patterns during the exploration phase. Videos and demonstrations of the interface can
524 be found on the website <https://intrinsically-motivated-discovery.github.io/>.

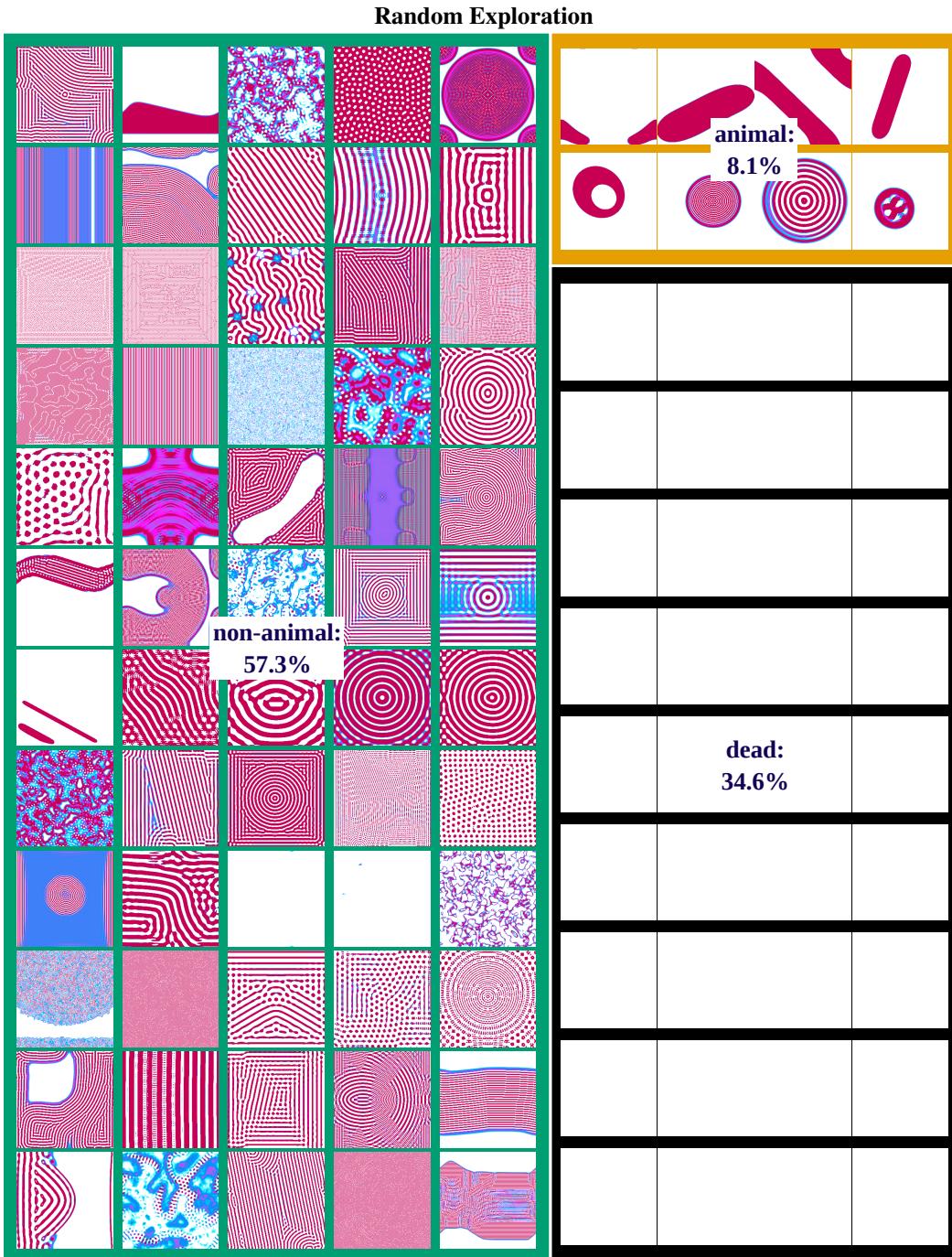


Figure 23: Examples of identified patterns for the random exploration algorithm from the first repetition of experiments.

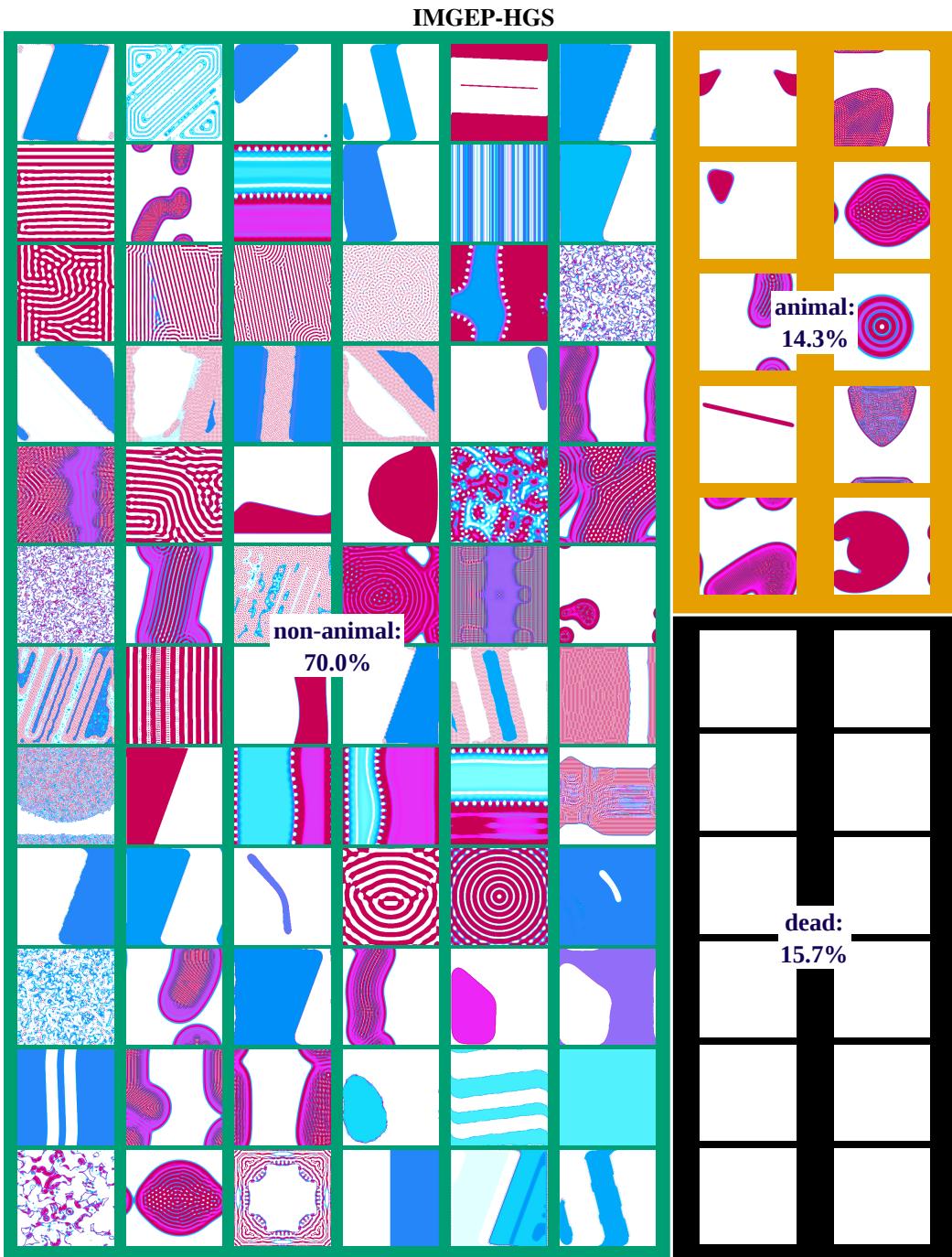


Figure 24: Examples of identified patterns for the IMGEP-HGS algorithm from the first repetition of experiments.

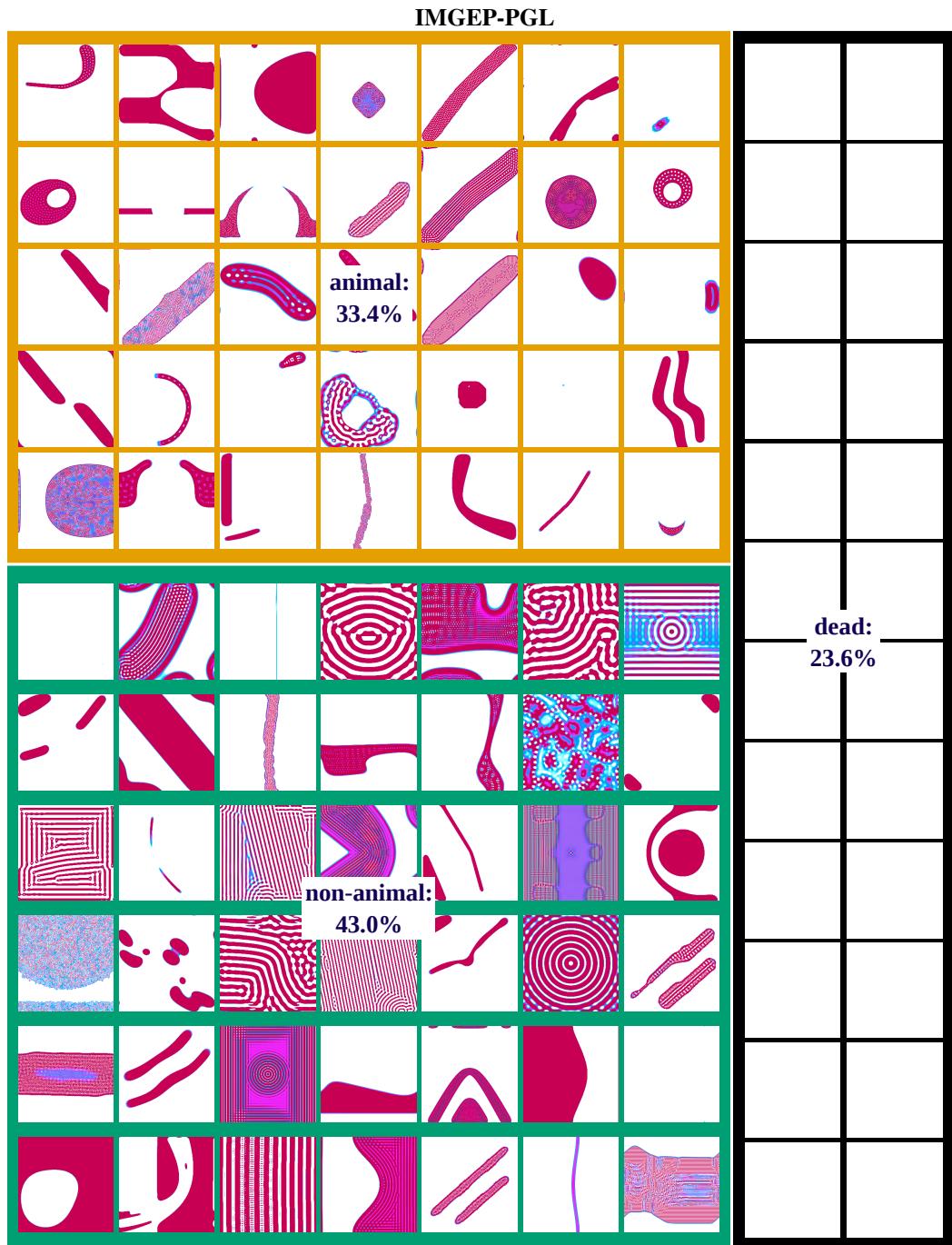


Figure 25: Examples of identified patterns for the IMGEP-PGL algorithm from the first repetition of experiments.

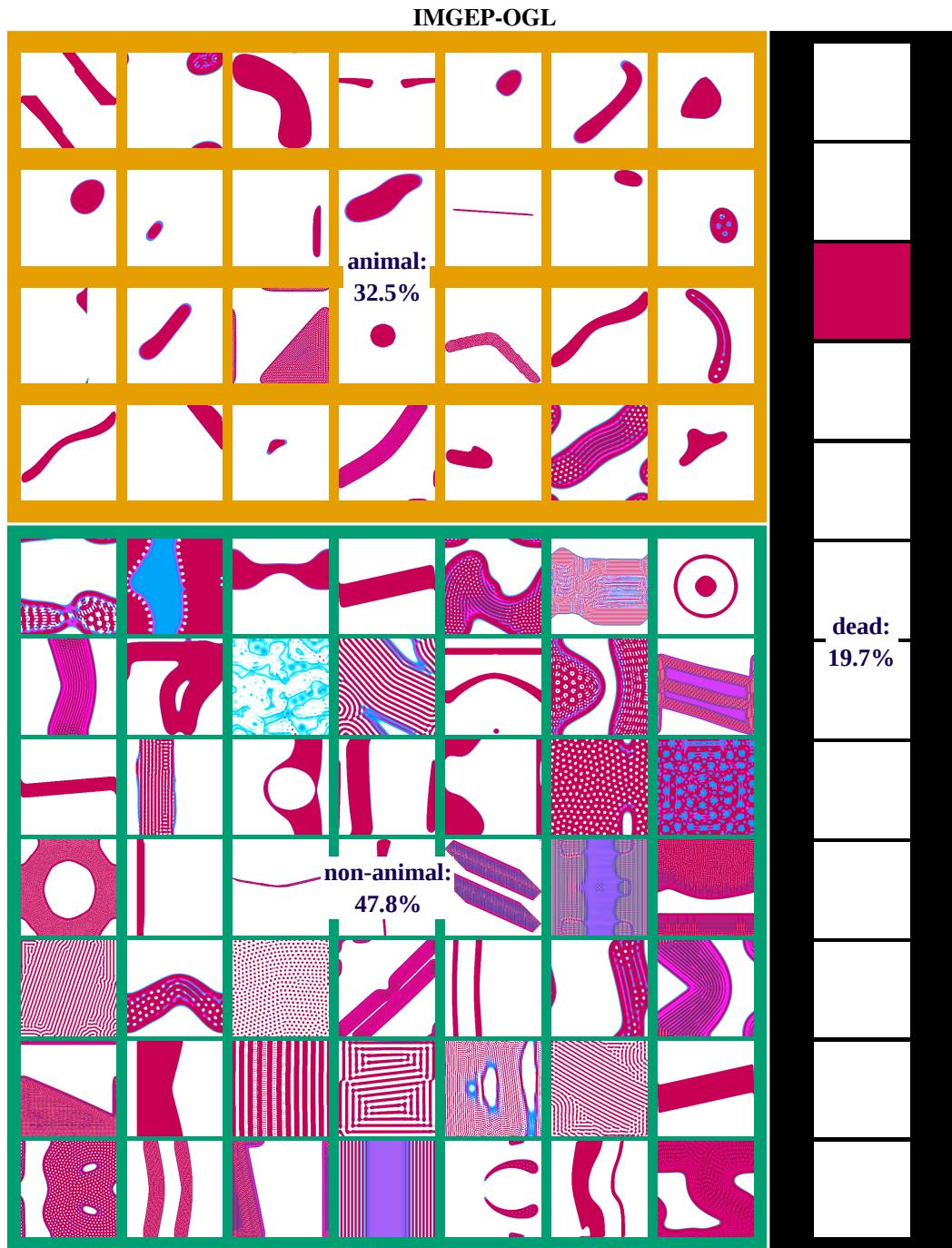


Figure 26: Examples of identified patterns for the IMGEP-OGL algorithm from the first repetition of experiments.

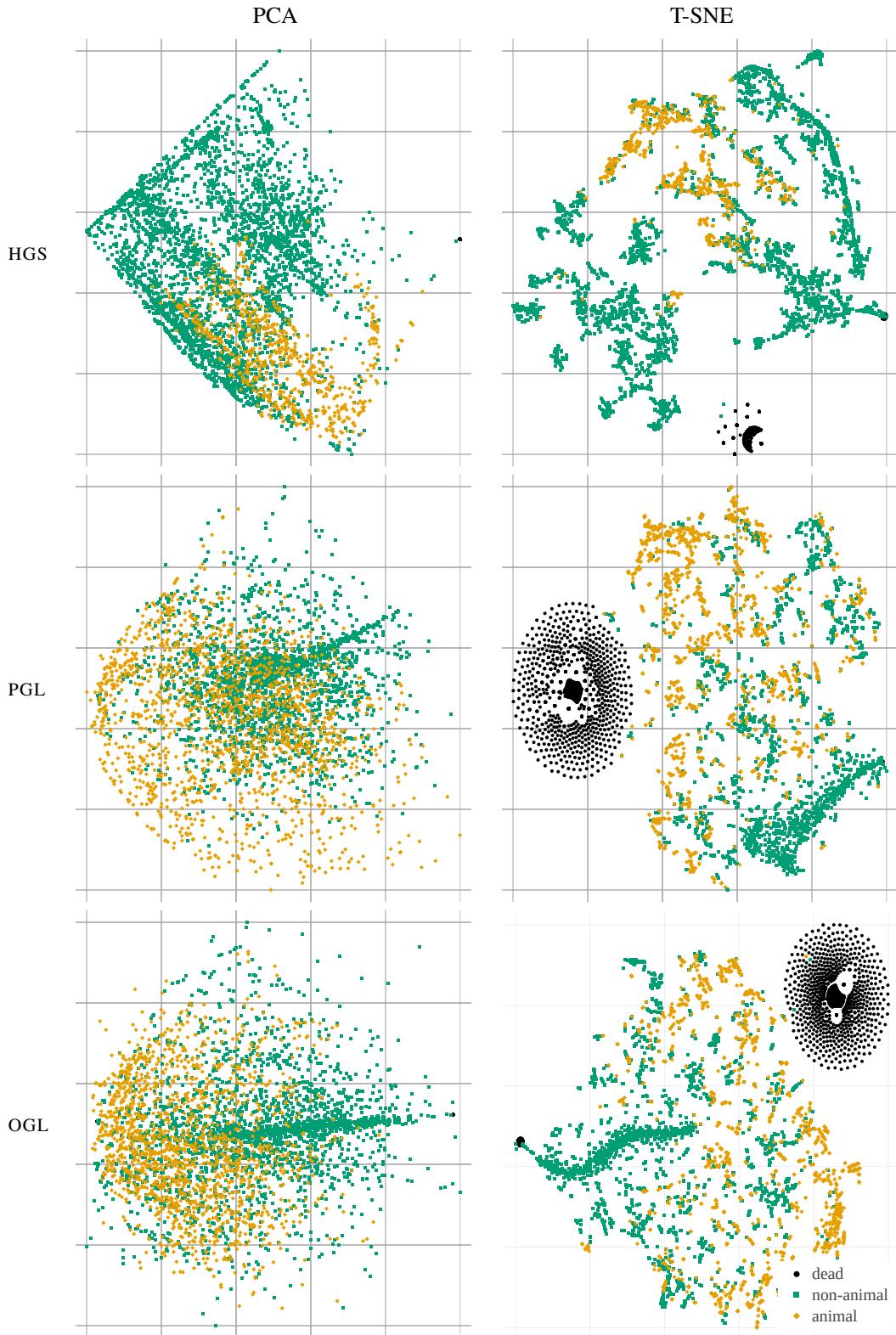


Figure 27: The PCA and t-SNE visualization of the goal spaces for the IMGEP variants shows that the hand-defined goal space (HGS) has more area and clusters for non-animals compared to learned goal spaces (PGL and OGL). They have more area and clusters for animals.

525 **References**

- 526 [1] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy
527 Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*,
528 2015.
- 529 [2] Bert Wang-Chak Chan. Lenia-biology of artificial life. *arXiv preprint arXiv:1812.05433*, 2018.
- 530 [3] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of
531 disentanglement in variational autoencoders. In *Advances in Neural Information Processing
532 Systems*, pages 2610–2620, 2018.
- 533 [4] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya
534 Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*,
535 2016.
- 536 [5] Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference
537 networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*,
538 2019.
- 539 [6] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick,
540 Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a
541 constrained variational framework. In *International Conference on Learning Representations*,
542 volume 3, 2017.
- 543 [7] I. T. Jolliffe. *Principal Component Analysis and Factor Analysis*, pages 115–128. Springer
544 New York, New York, NY, 1986.
- 545 [8] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*,
546 2018.
- 547 [9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint
548 arXiv:1412.6980*, 2014.
- 549 [10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint
550 arXiv:1312.6114*, 2013.
- 551 [11] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling.
552 Improved variational inference with inverse autoregressive flow. In *Advances in neural informa-
553 tion processing systems*, pages 4743–4751, 2016.
- 554 [12] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of
555 disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*,
556 2017.
- 557 [13] Adrien Laversanne-Finot, Alexandre Pere, and Pierre-Yves Oudeyer. Curiosity driven explo-
558 ration of learned disentangled goal spaces. In Aude Billard, Anca Dragan, Jan Peters, and
559 Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of
560 *Proceedings of Machine Learning Research*, pages 487–504. PMLR, 29–31 Oct 2018.
- 561 [14] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine
562 learning research*, 9(Nov):2579–2605, 2008.
- 563 [15] Kenneth O Stanley. Exploiting regularity without development. In *Proceedings of the AAAI
564 Fall Symposium on Developmental Systems*, page 37. AAAI Press Menlo Park, CA, 2006.
- 565 [16] Kenneth O Stanley and Risto Miikkulainen. Efficient evolution of neural network topologies. In
566 *Proceedings of the 2002 Congress on Evolutionary Computation.*, volume 2, pages 1757–1762.
567 IEEE, 2002.
- 568 [17] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-
569 encoders. *arXiv preprint arXiv:1711.01558*, 2017.
- 570 [18] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of modern physics*,
571 55(3):601, 1983.

- 572 [19] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational
573 autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.