

Appendices for Convergent Functions, Divergent Forms

The following items are provided in the Appendix:

- *Benefits of clustering the design space in the **learned latent space** instead of using raw morphology parameters* (App.[A](#))
- *Analysis of the locomotion **learning speed** of our evolved agents (fast and slow learners)* (App.[B](#))
- *Procedure for **sampling random morphologies** within the UNIMAL space* (App.[C](#))
- *Detailed description of the different **test tasks*** (App.[D](#))
- *Detailed description of the **QD-score** metric* (App.[E](#))
- *Description of how **efficiency metrics** in Tab.[I](#) are calculated* (App.[F](#))
- *Pseudo-code for the **Map-Elites** baseline* (App.[G](#))
- *Full **hyperparameter** details for the experiments* (App.[H](#))
- *Broader Societal Impacts* (App.[I](#))

On our website (loki-codesign.github.io), we have

- *Videos showing the diverse locomotion behaviors of LOKI’s evolved agents*
- *Videos of the evolved agents transferred to new test tasks (Obstacle, Bump, Incline, Push box incline, Manipulation ball, Exploration)*

A Benefits of Clustering in the Learned Latent Space

To investigate the role of latent-space clustering versus clustering based on raw morphology parameters, we conduct an ablation study: LOKI (w/ RAW-PARAM-CLUSTER), a variant of our method in which latent-space clusters are replaced with clusters computed from raw morphology parameters, while keeping the number of clusters fixed at $N_c = 40$.

Fig.[8](#) overlays the cumulative mean reward of the top 10 agents from LOKI (w/ RAW-PARAM-CLUSTER) and MAP-ELITES (w/ RAW-PARAM-CLUSTER) on top of Fig.[6](#) in the main paper. Notably, LOKI outperforms LOKI (w/ RAW-PARAM-CLUSTER) across all tasks, indicating that our co-evolution framework benefits from structuring the morphology space via a learned latent representation. In contrast, clustering in the raw parameter space fails to capture structural and behavioral similarities, leading to poor generalization of multi-design policies within each cluster. Tab.[3](#) also compares LOKI against LOKI (w/ RAW-PARAM-CLUSTER) across the quality-diversity metrics.

B Learning Speed of the Evolved Morphologies (Fast & Slow Learners)

We examine the correlation between learning speed and the final performance of evolved agents. To evaluate performance across different training durations, we train single-agent MLP policies [\[4\]](#) for both 5M and 15M steps on flat terrain. This evaluation excludes agents from clusters with consistently low training rewards or predominantly simple morphologies. As shown in Fig.[9](#), the top-performing agents under short- and long-term training are largely disjoint, revealing the presence of both *fast*- and *slow*-learning morphologies. Notably, high short-term performance does not necessarily indicate high long-term performance—some slow learners achieve superior results after 15M steps despite underperforming at 5M steps.

Prior approaches such as DERL train single-agent policies for a fixed number of steps (e.g., 5M), which biases evolution toward fast learners—often resulting in morphologies dominated by cheetah-like forms. In contrast, LOKI does not rely on short-term training. It leverages multi-design transformer policies trained over significantly longer durations, enabling the discovery of both fast and slow learners across the morphology clusters.

Table 3: LOKI benefits both **quality** and **diversity** from clustering in a structured morphology latent space. *One new baseline*, LOKI (w/ RAW-PARAM-CLUSTER), marked with (*), is added to Tab. 2 to assess the impact of clustering in raw parameter space. (\pm denotes standard error across 4 training seeds.)

Method	Max Fitness	QD-score	Coverage(%)
RANDOM	3418.9 \pm 390.8	32.1	90
DERL	5760.8 \pm 248.2	26.2	37.5
MAP-ELITES (w/ RAW-PARAM-CLUSTER)	5807.3 \pm 196.5	43.5	65.0
LATENT-MAP-ELITES	5257.0 \pm 491.3	38.1	100
LOKI (w/o CLUSTER)	4825.8 \pm 76.1	40.0	75
LOKI ($N_c = 10$)	4008.0 \pm 363.7	21.6	47.5
LOKI ($N_c = 20$)	5544.0 \pm 268.1	26.6	45.0
LOKI (w/ RAW-PARAM-CLUSTER) (*)	4055.0 \pm 323.7	27.5	62.5
LOKI 🦊 ($N_c = 40$)	5671.9 \pm 360.1	60.9	100

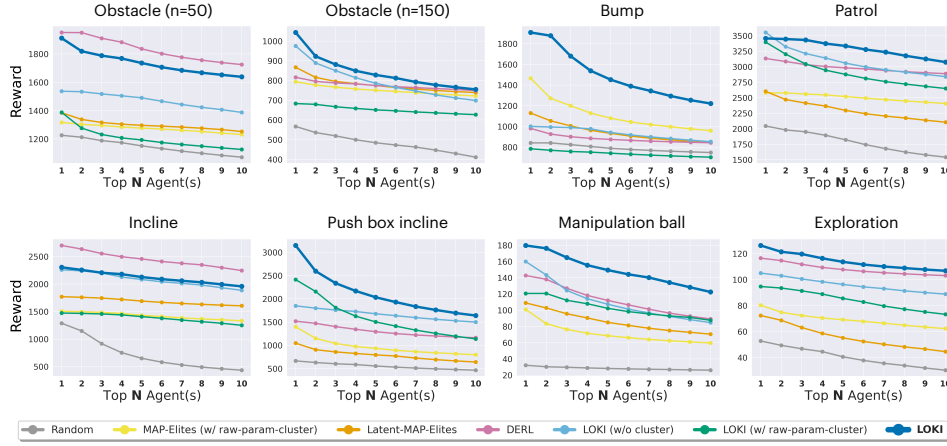


Figure 8: **Morphology-level task adaptability.** Two baselines (MAP-ELITES (w/ RAW-PARAM-CLUSTER), LOKI (w/ RAW-PARAM-CLUSTER)) are added to Fig. 6 to assess the impact of clustering in raw parameter space.

C Sampling Random Morphologies within the UNIMAL Space

The process of sampling random morphologies follows the procedure introduced in prior work [4]. During the population initialization phase, a new morphology is created by first sampling the total number of limbs to grow, followed by a series of mutation operations until the desired number of limbs is reached. These mutations include: *growing or deleting limbs*, *mutating limb parameters*, *density*, *degrees of freedom (DoF)*, *gear ratios*, and *joint angles*. For each mutation, the parameters are uniformly sampled from predefined ranges specified in [4]. The key difference is that DERL [4] samples parameter values from discrete sets, while LOKI samples continuously within each range, enabling a denser and more comprehensive coverage of the morphology space. Table 4 presents the parameter ranges used to create our morphologies. The notation $range(a, b)$ denotes a continuous range from a to b .

D Test Task Descriptions

We describe the task specifications and required skills for the eight test tasks used in our morphology-level evaluation. All tasks, except for *Bump* and *Obstacle (n=150)*, were introduced in prior work [4]. We adopt the same task configurations and reward structures as outlined therein.

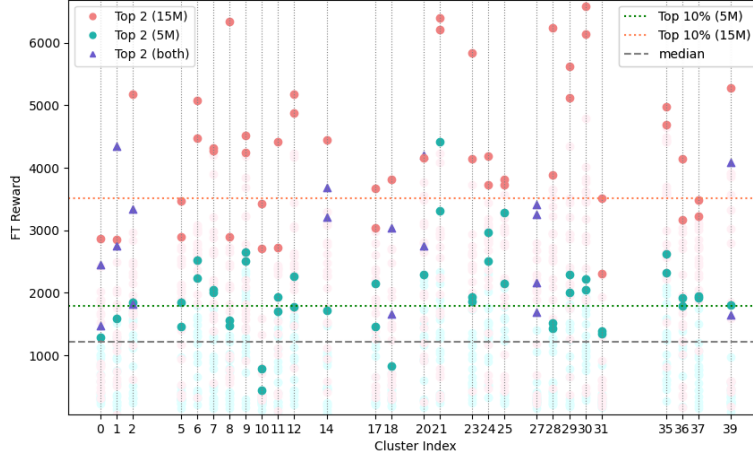


Figure 9: **Training rewards across short- and long-term learning.** Final rewards for LOKI’s agents after 5M and 15M training steps. The top-performing agents at short and long training horizons are largely disjoint. High short-term performance does not guarantee long-term success—some slow learners significantly outperform fast learners after extended training.

Parameters	Sampling Range
Max limbs	11
Limb radius	range(0.02, 0.06)
Limb height	range(0.2, 0.4)
Limb density	range(500, 1000)
Limb orientation θ	[0, 45, 90, 135, 180, 225, 270, 315]
Limb orientation ϕ	[90, 135, 180]
Head radius	0.10
Head density	range(500, 1000)
Joint axis	[x, y, xy]
Motor gear range	range(150, 300)
Joint limits	[(-30, 0), (0, 30), (-30, 30), (-45, 45), (-45, 0), (0, 45), (0, -60), (0, 60), (-60, 60), (-90, 0), (0, 90), (-60, 30), (-30, 60)]

Table 4: Design parameters used for sampling random morphologies in the UNIMAL space.

Patrol. The agent must repeatedly traverse between two target points separated by 10m along the x-axis. High performance in this task requires rapid acceleration, short bursts of speed, and quick directional changes. (Training steps: 5 million)

Incline. The agent operates in a $150 \times 40\text{m}^2$ rectangular arena inclined at a 10-degree angle. The agent is rewarded for moving forward along the +x axis. (Training steps: 5 million)

Push Box Incline. The agent must push a box with a side length of 0.2m up an inclined plane. The environment is a $80 \times 40\text{m}^2$ rectangular arena tilted at a 10-degree angle. The agent starts at one end of the arena and is tasked with propelling the box forward along the slope. (Training steps: 5 million)

Obstacle (n=50, 150). The agent must navigate through a cluttered environment filled with static obstacles to reach the end of the arena. Each box-shaped obstacle has a width and length between 0.5m and 3 m, with a fixed height of 2 m. n denotes the number of randomly distributed obstacles across a flat $150 \times 60\text{m}^2$ terrain. (Training steps: 5 million)

Bump. The agent must traverse an arena filled with 250 low-profile obstacles randomly placed on a flat $150 \times 60\text{m}^2$ terrain. Each obstacle has a width and length between 0.8m and 1.6m, and a height between 0.1m and 0.25m. As obstacle height is comparable to the agent’s body, this task promotes behaviors such as jumping or climbing, adding complexity to locomotion and body coordination. (Training steps: 15 million)

Manipulate Ball. The agent must move a ball from a random source location to a fixed target. A ball of radius 0.2 m is placed at a random location in a flat, square $30 \times 30\text{m}^2$ arena, with the agent initialized at the center. This task requires a fine interplay of locomotion and object manipulation, as the agent must influence the ball’s motion through contact while maintaining its own balance and stability. (Training steps: 20 million)

Exploration. The agent begins at the center of a flat $100 \times 100\text{m}^2$ arena divided into $1 \times 1\text{m}^2$ grid cells. The goal is to maximize the number of unique grid cells visited during an episode. Unlike previous tasks with dense locomotion rewards, this task provides a sparse reward signal. (Training steps: 20 million)

E QD-Score (a quality-diversity metric)

QD-score [64, 19, 55] is a more comprehensive metric than maximum fitness, as it captures not only the performance of the single best agent but also the diversity of high-performing solutions across the search space. Given the vast combinatorial complexity of the UNIMAL space, QD-score is particularly well-suited for evaluating the quality and spread of evolved morphologies.

In Tab. 2, we report the percentile QD-score computed over $N_c = 40$ latent morphology clusters, defined as:

$$\text{QD-score} = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbb{1}_{\|\mathcal{M}_i\|>0} \cdot \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \quad (1)$$

Here, \mathcal{M}_i denotes the set of evolved morphologies allocated to the i -th cluster, and f_i is the mean fitness of the best-performing agent in that cluster. f_{\max} and f_{\min} represent the maximum and minimum mean fitness values across all final population clusters, respectively. This normalization ensures comparability across clusters with different fitness scales.

F Efficiency Metrics in Table 1.

This section defines the efficiency metrics reported in Tab. 1.

Number of interactions. The number of interactions refers to the total number of environment steps taken by all agents throughout the entire evolutionary process.

In our multi-design evolution framework, two types of interactions are counted: (1) trajectories collected from agents in the training pool, which are used to train the shared multi-design policy via PPO, and (2) evaluation episodes conducted during each drop-off round.

The total number of interactions is calculated as:

$$\begin{aligned} \text{Total interactions} &= N_c \cdot (\# \text{ interactions (per cluster)} + \# \text{ evaluated samples} \cdot \text{episode length}) \\ &= N_c \cdot \left(100\text{M} + \left\lfloor \frac{N_{\text{iter}}}{f_{\text{diff}}} \right\rfloor \cdot N_{\text{sample}} \cdot l_{\text{eval}} \right) \\ &\approx 4.62\text{B} \end{aligned} \quad (2)$$

In contrast, DERL trains 4,000 agents independently using a single-agent MLP policy, with each agent trained for 5M environment steps—resulting in a total of 20B interactions. Since DERL uses the training reward directly for agent selection, no separate evaluation phase is required.

Number of searched morphologies. The total number of morphologies explored across the morphology space is calculated as follows:

$$\# \text{ of covered morphologies (per cluster)} \leq \left\lfloor \frac{N_{\text{iter}}}{f_{\text{diff}}} \right\rfloor \cdot N_{\text{sample}} + N_w \approx 78,000 \quad (3)$$

$$\begin{aligned}
\# \text{ of searched morphologies} &= \# \text{ of covered morphologies (per cluster)} \cdot N_c \\
&\approx 78,000 \cdot 40 \\
&= 3.12\text{M}
\end{aligned} \tag{4}$$

Note that each morphology is evaluated approximately 6 times on average. These repeated evaluations are valuable because early-stage policies may be unreliable. Re-evaluating morphologies with progressively improved policies allows for the discovery of higher-performing behaviors.

FLOPs per searched morphology. Given the MLP and Transformer policy model architectures used in prior work [4, 63], we compute the training FLOPs per searched morphology as shown in Eq. 5 with the values in Tab. 5.

$$\begin{aligned}
\text{FLOPs (per step)} &= 2 \times \text{FLOPs (per forward pass)} \times \text{Batch size} \\
\text{FLOPs (per model)} &= \text{FLOPs (per step)} \times \text{PPO epochs} \times (\# \text{ of iterations}) \\
\text{FLOPs (per morphology)} &= \text{FLOPs (per model)} / (\# \text{ of searched morphologies})
\end{aligned} \tag{5}$$

DERL employs a single-agent MLP policy, resulting in a per-model training compute of $2 \times 31.9\text{k} \times 512 \times 4 \times 1220 = 159\text{B}$ FLOPs. In contrast, LOKI uses a multi-agent Transformer policy, which incurs higher compute per forward pass, as well as larger batch sizes and more training epochs. However, this higher cost is offset by the fact that each trained model serves a large number of morphologies. As a result, the total training compute per searched morphology is amortized and given by $\frac{2 \times 79.5\text{M} \times 5120 \times 8 \times 1220}{78,000} \approx 102\text{B}$ FLOPs. Despite the higher overall training cost of the Transformer policy compared to the MLP, its shared usage across a large number of morphologies leads to more sample-efficient training, resulting in approximately 40% lower compute cost per morphology.

Table 5: Comparison of total **FLOPs** required for MLP and Transformer policy architectures.

Model	FLOPs (per forward pass)	Batch size	PPO Epochs	# of evaluated morphologies (per model)
MLP	31.9K	512	4	1
Transformer	79.5M	5120	8	78,000

G MAP-Elites

We provide the algorithm for the Map-Elites [23, 49] baseline in Alg. 2.

Algorithm 2 MAP-ELITES

```
1: Input:  
    $\mathcal{M}$ : MAP-Elites repertoire for agent design  
    $\mathcal{U}$ : UNIMAL morphology space  
    $g$ : Cluster classifier based on morphology parameters  
    $N_{\text{train}}$ : Total number of agents to train  
    $M$ : Number of offspring per generation  
    $S$ : Number of interaction steps for training each MLP policy  
2: // Initialization  
3: Randomly sample  $M$  initial morphologies  $\{u_i^0\}_{i=1}^M \subset \mathcal{U}$   
4: Train single-agent MLP policies  $\{\pi_i^0\}_{i=1}^M$  on their respective morphologies for  $S$  steps  
5: Evaluate fitness  $\{f_i^0\}_{i=1}^M$  via one episode roll-out per trained policy  
6: Insert  $\{u_i^0\}_{i=1}^M$  into  $\mathcal{M}$  using fitness  $\{f_i^0\}_{i=1}^M$  and cluster assignments  $\{g(u_i^0)\}_{i=1}^M$   
7: while  $|\mathcal{M}| < N_{\text{train}}$  do  
8:   // Reproduction via mutation  
9:   Sample  $M$  elite morphologies  $\{u_i^j\}_{i=1}^M$  from  $\mathcal{M}$  without replacement  
10:  Apply random mutation to produce  $M$  offspring morphologies  $\{\tilde{u}_i^j\}_{i=1}^M$   
11:  // Train and evaluate new morphologies  
12:  Train MLP policies  $\{\pi_i^j\}_{i=1}^M$  for  $S$  steps  
13:  Evaluate fitness  $\{f_i^j\}_{i=1}^M$  via one episode per policy  
14:  Insert  $\{\tilde{u}_i^j\}_{i=1}^M$  into  $\mathcal{M}$  using fitness  $\{f_i^j\}_{i=1}^M$  and clusters  $\{g(\tilde{u}_i^j)\}_{i=1}^M$   
15: end while
```

H Implementation Details

Detailed hyperparameters are provided in Tab. 6 and Tab. 7.

Table 6: **Hyperparameters of LOKI.**

	Name	Value
Stochastic Multi-Design Evolution	# of samples for K-means	5×10^6
	# of clusters N_c	10, 20, 40
	N_{iter}	1220
	f_{diff}	2
	N_w	20
	N_{filter}	2
	N_{sample}	128
	# of parallel environments	32
	Total interactions	10^8
	Timesteps per PPO rollout	2560
	PPO epochs	8
	Training episode length l_{train}	1000
	Evaluation episode length l_{eval}	200
Multi-Design Transformer Policy	# of heads	2
	# of layers	5
	Batch size	5120
	Feedforward dimension	1024
	Dropout	0.0
	Initialization range for embedding	[-0.1, 0.1]
	Initialization range for decoder	[-0.01, 0.01]
	Limb embedding size	128
	Joint embedding size	128
Morphology VAE	Continuous feature embedding size	32
	Depth feature embedding size	32
	Latent dimension	32
	# of heads	4
	Feed-forward network hidden dimension	256
	# of layers	4
	Optimizer	Adam
	Initial learning rate	10^{-4}
	Weight decay	10^{-5}
	Learning rate scheduler	ReduceLROnPlateau [65]
	Learning rate reduction factor	0.95
	Learning rate reduction patience	10
	# of epochs	200
	Batch size	4096
	$[\beta_{\min}, \beta_{\max}]$	$[10^{-5}, 10^{-2}]$

Table 7: **Hyperparameters of MAP-ELITES.**

Name	Value
# of samples for K-means	5×10^6
# Clusters	40
N_{train}	4000
M	100
S	5×10^6

I Societal Impacts

Positive impacts: Our work aims to challenge prevailing assumptions about optimal robot morphology, potentially reshaping how robotic design is approached. By promoting diversity and functionality

beyond conventional forms, LOKI encourages exploration of unconventional yet effective morphologies. We believe this contributes toward a more inclusive and biologically inspired understanding of embodiment, potentially serving as a bridge between AI, robotics, and the natural sciences.

Negative impacts: A potential concern is that the ability to autonomously generate a large number of novel and capable morphologies could be misused in contexts that prioritize performance over safety. For instance, this approach could enable rapid prototyping of morphologies for autonomous systems without adequate human oversight, increasing the risk of deploying untested designs in sensitive environments.

References

- [1] Francisco J Varela, Evan Thompson, and Eleanor Rosch. *The embodied mind, revised edition: Cognitive science and human experience*. MIT press, 2017.
- [2] Emma Hjellbrekke Stensby, Kai Olav Ellefsen, and Kyrre Glette. Co-optimising robot morphology and controller in a simulated open-ended environment. In *Applications of Evolutionary Computation: 24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings 24*, pages 34–49. Springer, 2021.
- [3] Julius Wolff. *The law of bone remodelling*. Springer Science & Business Media, 2012.
- [4] Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature communications*, 12(1):5721, 2021.
- [5] Tingwu Wang, Renjie Liao, Jimmy Ba, and Sanja Fidler. Nervenet: Learning structured policy with graph neural networks. In *International conference on learning representations*, 2018.
- [6] Charles Schaff, David Yunis, Ayan Chakrabarti, and Matthew R Walter. Jointly learning to construct and control agents using deep reinforcement learning. In *2019 international conference on robotics and automation (ICRA)*, pages 9798–9805. IEEE, 2019.
- [7] Deepak Pathak, Christopher Lu, Trevor Darrell, Phillip Isola, and Alexei A Efros. Learning to control self-assembling morphologies: a study of generalization via modularity. *Advances in Neural Information Processing Systems*, 32, 2019.
- [8] Kevin Sebastian Luck, Heni Ben Amor, and Roberto Calandra. Data-efficient co-adaptation of morphology and behaviour with deep reinforcement learning. In *Conference on Robot Learning*, pages 854–869. PMLR, 2020.
- [9] Tingwu Wang, Yuhao Zhou, Sanja Fidler, and Jimmy Ba. Neural graph evolution: Towards efficient automatic robot design. *arXiv preprint arXiv:1906.05370*, 2019.
- [10] Wenlong Huang, Igor Mordatch, and Deepak Pathak. One policy to control them all: Shared modular policies for agent-agnostic control. In *International Conference on Machine Learning*, pages 4455–4464. PMLR, 2020.
- [11] Ye Yuan, Yuda Song, Zhengyi Luo, Wen Sun, and Kris Kitani. Transform2act: Learning a transform-and-control policy for efficient agent design. *arXiv preprint arXiv:2110.03659*, 2021.
- [12] Vitaly Kurin, Maximilian Igl, Tim Rocktäschel, Wendelin Boehmer, and Shimon Whiteson. My body is a cage: the role of morphology in graph-based incompatible control. *arXiv preprint arXiv:2010.01856*, 2020.
- [13] David Ha. Reinforcement learning for improving agent design. *Artificial life*, 25(4):352–365, 2019.
- [14] Karl Sims. Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15–22, 1994.
- [15] David E Goldberg. Genetic algorithms and walsh functions: Part 2, deception and its analysis. *Complex systems*, 3:153–171, 1989.
- [16] Gregory S Hornby. Alps: the age-layered population structure for reducing the problem of premature convergence. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 815–822, 2006.
- [17] Leonardo Trujillo, Gustavo Olague, Evelyn Lutton, Francisco Fernandez de Vega, León Dozal, and Eddie Clemente. Speciation in behavioral space for evolutionary robotics. *Journal of Intelligent & Robotic Systems*, 64:323–351, 2011.
- [18] Leonardo Trujillo, Gustavo Olague, Evelyn Lutton, and Francisco Fernández de Vega. Discovering several robot behaviors through speciation. In *Workshops on Applications of Evolutionary Computation*, pages 164–174. Springer, 2008.

- [19] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.
- [20] Joel Lehman and Kenneth O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Annual Conference on Genetic and Evolutionary Computation*, 2011.
- [21] Konstantinos Chatzilygeroudis, Antoine Cully, Vassilis Vassiliades, and Jean-Baptiste Mouret. Quality-diversity optimization: a novel branch of stochastic optimization. In *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*, pages 109–135. Springer, 2021.
- [22] Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. *IEEE Transactions on Evolutionary Computation*, 22:623–630, 2016.
- [23] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *ArXiv*, abs/1504.04909, 2015.
- [24] Jean-Baptiste Mouret. Evolving the behavior of machines: from micro to macroevolution. *Iscience*, 23(11), 2020.
- [25] Holger H. Hoos and Thomas Stützle. *Stochastic Local Search Algorithms: An Overview*, pages 1085–1105. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
- [26] Frank Hutter, Dave A. D. Tompkins, and Holger H. Hoos. Scaling and probabilistic smoothing: Efficient dynamic local search for sat. In *International Conference on Principles and Practice of Constraint Programming*, 2002.
- [27] W. Pullan and H. H. Hoos. Dynamic local search for the maximum clique problem. *Journal of Artificial Intelligence Research*, 25:159–185, February 2006.
- [28] Hans Peter Moravec. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. Stanford University, 1980.
- [29] Artificial Intelligence Center. Shakey the robot. 1984.
- [30] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [31] Nicolas Manfred Otto Heess, TB Dhruva, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyun Wang, S. M. Ali Eslami, Martin A. Riedmiller, and David Silver. Emergence of locomotion behaviours in rich environments. *ArXiv*, abs/1707.02286, 2017.
- [32] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [33] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.
- [34] Heng Dong, Junyu Zhang, Tonghan Wang, and Chongjie Zhang. Symmetry-aware robot design with structured subgroups. In *International Conference on Machine Learning*, pages 8334–8355. PMLR, 2023.
- [35] Muhan Li, Lingji Kong, and Sam Kriegman. Generating freeform endoskeletal robots. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [36] Karl Sims. Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372, 1994.
- [37] Haoifei Lu, Zhe Wu, Junliang Xing, Jianshu Li, Ruoyu Li, Zhe Li, and Yuanchun Shi. Bodygen: Advancing towards efficient embodiment co-design. In *The Thirteenth International Conference on Learning Representations*, 2025.

- [38] Donald J Hejna, Pieter Abbeel, and Lerrel Pinto. Task-agnostic morphology evolution. In *9th International Conference on Learning Representations, ICLR 2021*, 2021.
- [39] Reiko Tanese. *Distributed genetic algorithms for function optimization*. University of Michigan, 1989.
- [40] Theodore C Belding. The distributed genetic algorithm revisited. *arXiv preprint adap-org/9504007*, 1995.
- [41] Darrell Whitley, Soraya Rana, and Robert B Heckendorn. Island model genetic algorithms and linearly separable problems. In *Evolutionary Computing: AISB International Workshop Manchester, UK, April 7–8, 1997 Selected Papers*, pages 109–125. Springer, 1997.
- [42] David E Goldberg, Jon Richardson, et al. Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, volume 4149, pages 414–425. Lawrence Erlbaum, Hillsdale, NJ, 1987.
- [43] Christopher K Oei, David E Goldberg, and Shau-Jin Chang. Tournament selection, niching, and the preservation of diversity. *IlligAL report*, 91011, 1991.
- [44] Bruno Sareni and Laurent Krahenbuhl. Fitness sharing and niching methods revisited. *IEEE transactions on Evolutionary Computation*, 2(3):97–106, 1998.
- [45] John Cartlidge and Seth Bullock. Caring versus sharing: How to maintain engagement and diversity in coevolving populations. In *European Conference on Artificial Life*, pages 299–308. Springer, 2003.
- [46] Melanie Mitchell, Michael D Thomure, and Nathan L Williams. The role of space in the success of coevolutionary learning. In *Artificial life X: proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 118–124. The MIT Press (Bradford Books) Cambridge, 2006.
- [47] Antoine Cully and Y. Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22:245–259, 2017.
- [48] Peter Krčah. Solving deceptive tasks in robot body-brain co-evolution by searching for behavioral novelty. In *2010 10th International Conference on Intelligent Systems Design and Applications*, pages 284–289. IEEE, 2010.
- [49] Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. *IEEE Transactions on Evolutionary Computation*, 22(4):623–630, 2017.
- [50] Vassilis Vassiliades and Jean-Baptiste Mouret. Discovering the elite hypervolume by leveraging interspecies correlation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 149–156, 2018.
- [51] Cédric Colas, Vashisht Madhavan, Joost Huizinga, and Jeff Clune. Scaling map-elites to deep neuroevolution. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 67–75, 2020.
- [52] Thomas PIERROT and Arthur Flajolet. Evolving populations of diverse RL agents with MAP-elites. In *The Eleventh International Conference on Learning Representations*, 2023.
- [53] Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. Data-efficient exploration, optimization, and modeling of diverse designs through surrogate-assisted illumination. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 99–106, 2017.
- [54] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.
- [55] Jørgen Nordmoen, Frank Veenstra, Kai Olav Ellefsen, and Kyrre Glette. Map-elites enables powerful stepping stones and diversity for modular robotics. *Frontiers in Robotics and AI*, 8:639173, 2021.

- [56] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017.
- [58] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [59] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.
- [60] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.
- [61] Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *The Twelfth International Conference on Learning Representations*, 2024.
- [62] Laurens Van der Maaten and Geoffrey Hinton. [Visualizing data using t-SNE.](#) *Journal of machine learning research*, 9(11), 2008.
- [63] Agrim Gupta, Linxi Fan, Surya Ganguli, and Li Fei-Fei. Metamorph: Learning universal controllers with transformers. *arXiv preprint arXiv:2203.11931*, 2022.
- [64] Manon Flageat, Bryan Lim, Luca Grillotti, Maxime Allard, Simón C Smith, and Antoine Cully. Benchmarking quality-diversity algorithms on neuroevolution for reinforcement learning. *arXiv preprint arXiv:2211.02193*, 2022.
- [65] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.