MorphVAE: Advancing Morphological Design of Voxel-Based Soft Robots with Variational Autoencoders

Technical Appendix

A Variational Inference of MorphVAE

The ELBO in our work is constructed as follows:

$$
\begin{aligned}
& E_{y\sim\mathcal{Y}}E_{x\sim\mathcal{X}_y}\log p_\theta(x|y) \\
&= E_{y\sim\mathcal{Y}}E_{x\sim\mathcal{X}_y}\log\int p_\theta(h|y)p_\theta(x|h)\mathrm{d}h \\
&= E_{y\sim\mathcal{Y}}E_{x\sim\mathcal{X}_y}\log\int \frac{p_\theta(x|h)p_\theta(h|y)}{q_\phi(h|x,y)}\cdot q_\phi(h|x,y)\mathrm{d}h \\
&\geq E_{y\sim\mathcal{Y}}E_{x\sim\mathcal{X}_y}E_{h\sim q_\phi(h|x,y)}\log\frac{p_\theta(x|h)p_\theta(h|y)}{q_\phi(h|x,y)} \\
&= E_{y\sim\mathcal{Y}}E_{x\sim\mathcal{X}_y}[E_{h\sim q_\phi(h|x,y)}\log p_\theta(x|h)] - D_{\mathrm{KL}}(p_\theta(h|y)||q_\phi(h|x,y)) \\
&\equiv \mathrm{ELBO},
\end{aligned}
$$

where $\mathcal{Y}$ denotes a uniform distribution over all tasks, $\mathcal{X}_y$ denotes the distribution of advantageous morphology corresponding to task y, which would be introduced in Section 4.2, $q_\phi(h|x,y)$ denotes the aforementioned approximate posterior, and "$\geq$" follows from Jensen's inequality. It can be shown that

$$
E_{y\sim\mathcal{Y}}E_{x\sim\mathcal{X}_y}\log p_\theta(x|y) = \mathrm{ELBO} + E_{y\sim\mathcal{Y}}E_{x\sim\mathcal{X}_y}D_{\mathrm{KL}}(p_\theta(h|x,y)||q_\phi(h|x,y)),
$$

and therefore assuming q_ϕ is of arbitrarily high capacity, we are able to maximize the likelihood function by maximizing ELBO. In this work, we parametrize q_ϕ as a Gaussian distribution as well, whose mean vector and diagonal variance-covariance matrix are computed by MLPs that both take x and y as input.

B Environment Details

In this section, we provide a brief introduction to the tasks we adopted in Evolution Gym, which is heavily borrowed from their original paper (Bhatia et al. 2021).

Let us first define some notations that would later be used.

- **Position:** Denote with p^o the position of the center of mass of an object o, which consists of two components p^o_x and p^o_y, i.e. the positions on x and y axis. p^o is derived by averaging the positions of all the point-masses that make up object o;
- **Velocity:** Denote with v^o the velocities of the center of mass of an object o, which consists of two components v^o_x and v^o_y, i.e. the velocity on x and y axis. v^o is computed by averaging the velocities of all point masses that make up object o;

- **Orientation:** Denote with θ^o, a vector of length one, the orientation of an object o. Denote the position of point mass i of object o as p_i, and θ^o is computed by averaging over all i the angle between the vector $p_i - p^o$ at current time and the initial state. This average is weighted by $||p_i - p^o||$ in the initial state.
- **Other observations:** Let c^o be a vector of length $2n$ that describes the relative positions of all n point masses of object o to the center of mass. Let $h_b^o(d)$ characterize the terrain information around a robot below its center of mass. More specifically, for some integer $x \leq d$, the corresponding entry in vector $h_b^o(d)$ will be the highest point of the terrain which is lower than p_y^o between a range of $[x, x+1]$ voxels from p_x^o in the x-direction.
- Besides, we would denote the robot as object r, the box that it is trying to manipulate as object b, the number of point masses in r as n, the observation vector as $\mathcal{S}$, and the reward function as R.

B.1 Carrier-v0

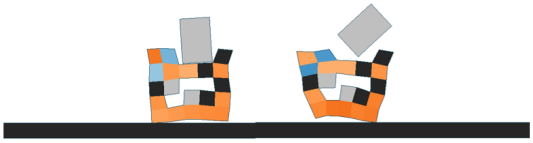

Figure 1: Carrier-v0

In this task, the robot is required to catch a box initialized above it and carries it as far as possible. $\mathcal{S} \in \mathbb{R}^{n+6}$ consists of v^b, $p^b - p^r$, v^r and c^r with lengths 2, n, 2 and 2 respectively. $R = R_1 + R_2$, where $R_1 = 0.5 \cdot \Delta p_x^r + 0.5 \cdot \Delta p_x^b$ rewards the robot and the box for moving in the positive x-direction, and $R_2 = 0$ if $p_y^b \geq t_y$ and otherwise $10 \cdot \Delta p_y^b$ penalizes the robot for dropping the box below a threshold height t_y. The robot is also given a one-time reward of 1 for reaching the end of the terrain.

B.2 Pusher-v0

Figure 2: Pusher-v0

In this task, the robot is required to push a box initialized in front of it. $\mathcal{S} \in \mathbb{R}^{n+6}$ consists of v^b, $p^b - p^r$, v_r and c^r with lengths 2, n, 2 and 2 respectively. $R = R_1 + R_2$, where $R_1 = 0.5 \cdot \Delta p_x^r + 0.75 \cdot \Delta p_x^b$ rewards the robot and the box for moving in

the positive x-direction, and $R_2 = -\Delta|p_x^b - p_x^r|$ penalizes the robot and the box for separating in the x-direction. The robot is also given a one-time reward of 1 for reaching the end of the terrain.

B.3 Climber-v0

Figure 3: Climber-v0

In this task, the robot is required to climb as high as possible through a flat, vertical channel. $\mathcal{S} \in \mathbb{R}^{n+2}$ consists of v^r and c^r with lengths 2 and n, respectively. $R = \Delta p_y^r$ rewards the robot for moving in the positive y-direction. The robot is also given a one-time reward of 1 for reaching the end of the channel.

B.4 UpStepper-v0

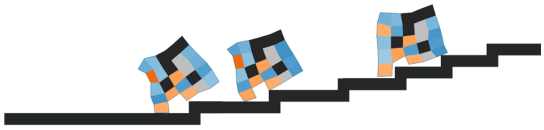

Figure 4: UpStepper-v0

In this task, the robot is required to mount stairs of varying lengths. $\mathcal{S} \in \mathbb{R}^{n+14}$ consists of v^r, θ^r, c^r and $h_b^r(5)$ with lengths 2, 1, n and 11, respectively. $R = \Delta p_x^r$ rewards the robot for moving in the positive x-direction. The robot is given a one-time reward of 2 for reaching the end of the terrain, and a one-time penalty of -3 for rotating more than 75 degrees from its original orientation in either direction (after which the environment is reset).

B.5 Walker-v0

In this task, the robot is required to walk as far as possible on flat terrain. $\mathcal{S} \in \mathbb{R}^{n+2}$ consists of v^r and c^r with lengths 2 and n. $R = \Delta p_x^r$ rewards the robot for moving in the positive x-direction. The robot is also given a one-time reward of 1 for reaching the end of the terrain.

Figure 5: Walker-v0

B.6 Catcher-v0

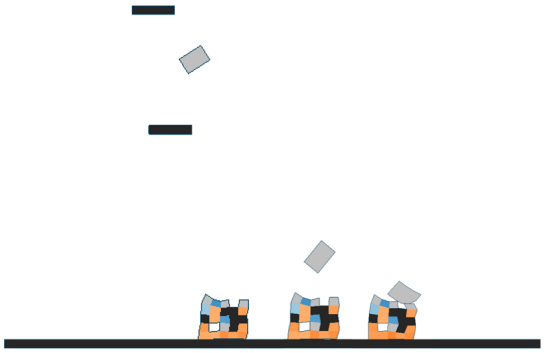

Figure 6: Catcher-v0

In this task, the robot is required to catch a fast-moving and rotating box. The observation space $\mathcal{S} \in \mathbb{R}^{n+7}$ consists of $p^b - p^r$, v^r, v^b, θ^b and c^r, with lengths 2, 2, 2, 1 and n, respectively. The reward $R = R_1 + R_2$, where $R_1 = -\Delta|p_x^b - p_x^r|$ rewards the robot for approaching the box in the x-direction, and $R_2 = 0$ if $p_y^b \geq t_y$ and $10 \cdot \Delta p_y^b$ otherwise penalizes the robot for dropping the box below a threshold height t_y.

C Hyperparameter Settings

In this section, we list all the hyperparameters that we chose for VAE model and PPO algorithm for reproducibility, as shown in Table 1. For more details of our implementation, please refer to our codes in supplementary materials. Moreover, in Algorithm 2 of main text, m_t linearly increases from 50 to 250 to prevent premature convergence; $n_t = \min(t \times P, 50)$, where t and P denotes the number of generations and population size respectively; β_t linearly decreases from 50% to zero; All robot designs must be unique and pass the valid test (i.e. be connected and have actuators) before being included into the population.

hyperparameter	value
VAE	
hidden dims of MLPs	[128]*3
dimension of task embeddings	128
temperature parameter τ in (2) of main text	1.5/0.7
population size P	25
robot size	5×5
learning rate	0.0001
β_1 and β_2 in ADAM optimizer	(0.95,0.999)
maximal number of evaluations	1000
PPO	
number of parallel sampling processes	4
number of time steps in each process	128
learning rate	2.5×10^{-4}
ϵ in the clip function of PPO	0.1
number of iterations	1000
number of epochs per iteration	4
number of mini-batches per epoch	4
λ in generalized advantage estimation (GAE)	0.95

Table 1: Hyperparameter settings

D Evaluation of Diversity

As mentioned in Section 4.4, we first pick out high-performing robot designs, and then one-hot encode them so that the dimension of each one becomes 125 (i.e. 25 voxels $\times$ 5 voxel types). We finally compute the variances of all the 125 dimensions across the robot samples and sum them up, which leads to our metric of morphological diversity. More formally, denote the one-hot encoded high-performing robot samples of task h as $\tilde{x}_i^h$, with $i = 1, \cdots, n$, and denote the j-th dimension of $\tilde{x}_i^h$ as $\tilde{x}_{ij}^h$, with $j = 1, \cdots, 125$. Then, the diversity of these robot designs is evaluated as:

$$\text{diversity} = \frac{1}{n} \sum_{j=1}^{125} \sum_{i=1}^{n} \left(\tilde{x}_{ij}^h - \bar{\tilde{x}}_j^h \right)^2,$$

where $\bar{\tilde{x}}_j^h = \frac{1}{n} \sum_{i=1}^{n} \tilde{x}_{ij}^n$.

E Demonstration of Evolved Morphologies

In this section, we showcase the best-performing robot designs obtained by each method in one of three runs, as well as their fitness (denoted as f). It can be seen from Figure

7 that successful climbers and upsteppers share common compound modes of voxels across different algorithms, which justifies the argument in Section 4.5 of main text and Appendix F that these tasks require specific morphological structures, whereas in the remaining four tasks, optimal robot designs turn out more diversified and reveal less explicit directions of high-performing morphology. Also note that for the robot designs obtained by MorphVAE and roboGAN, much resemblance could be found across different tasks, and this is largely due to the multi-task optimization scheme they adopted which allows for inter-task knowledge sharing.

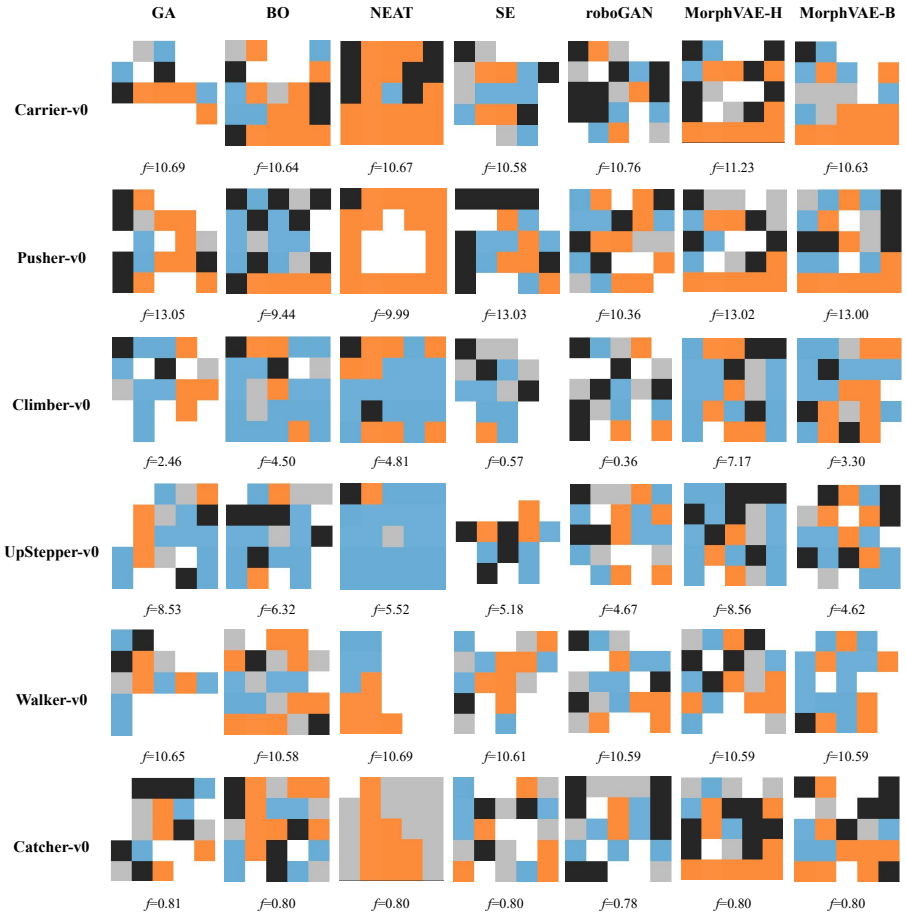

Figure 7: Visualization of best-performing robot designs

F Further Details of Comparisons in Section 4.5

As mentioned in Section 4.5 of the main text, similar to Climber-v0, UpStepper-v0 is also a challenging task that necessitates specific advantageous sub-structures for a robot to succeed, which makes MorphVAE stand out due to its ability to discern favorable patterns and significantly improved sample efficiency. To be more specific, a successful upstepper, as demonstrated in Figure 8, requires vertical actuators in both left and right parts of its body, serving as legs, and horizontal actuators in between (but absent in its upper part to avoid overstriding and thus backward toppling) to mount the steps. Any departure from such structures, as exemplified in Figure 9, would very likely lead to failure, such as a climber that hovers at the bottom of the pipe all the way, or an upstepper that gets stuck halfway.

Figure 8: Favorable structures in UpStepper-v0

Figure 9: Examples of departure from favorable structures in Climber-v0 (left) and UpStepper-v0 (right)

By contrast, Walker-v0 and Catcher-v0 have far less explicit directions of high-performing robot designs. For the former, this is because the task is so easy that even an arbitrary robot design could suffice for decent rewards, while for the latter, the main cause is its considerable difficulty that leads to noisy and hardly discriminative fitness evaluation. As a result, a variety of robot designs could achieve similar fitness, as demonstrated in Figure 10.

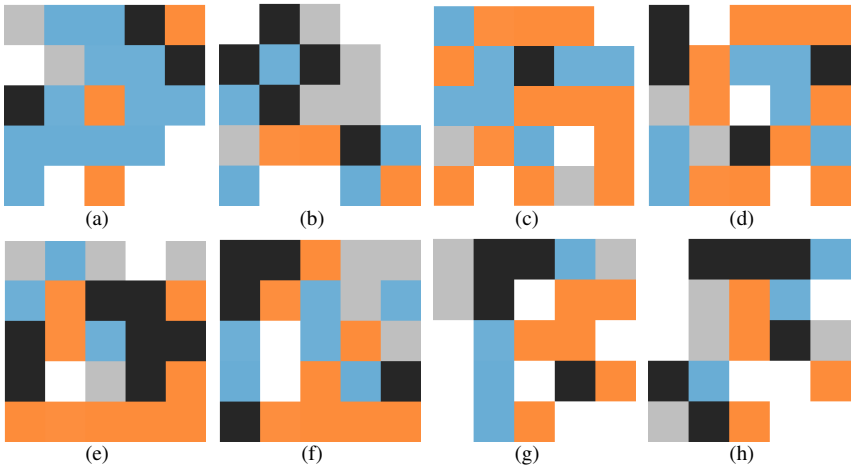

Figure 10: Examples of high-performing robot designs of Walker-v0 (first row) and Catcher-v0 (second row)

G Wilcoxon Rank Sum Test

In this section, we additionally provide the results of the Wilcoxon Rank Sum test (Table 2) to demonstrate the significance of MorphVAE's superiority to its counterparts. To be more specific, we test for the difference of robot fitnesses achieved throughout evolution between MorphVAE-H and its competitors, and found that MorphVAE-H almost consistently outperforms others with extremely small p values, except in Climber-v0 and Walker-v0 where the difference between MorphVAE-H and GA is insignificant.

		GA	BO	CPPN-NEAT	SE	roboGAN
Carrier-v0	test statistic	15.46	45.80	37.27	33.82	35.78
	p value	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
Pusher-v0	test statistic	11.23	42.15	42.33	33.76	41.02
	p value	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
Climber-v0	test statistic	0.88	23.40	15.90	9.96	19.19
	p value	0.38	< 0.001	< 0.001	< 0.001	< 0.001
UpStepper-v0	test statistic	14.70	40.49	38.95	35.15	41.31
	p value	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
Walker-v0	test statistic	-1.93	14.49	24.28	15.75	14.58
	p value	0.05	< 0.001	< 0.001	< 0.001	< 0.001
Catcher-v0	test statistic	10.45	21.93	22.04	18.80	12.10
	p value	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001

Table 2: Results of the Wilcoxon Rank Sum test with MorphVAE-H versus baselines

H Comparison of Median Fitness

Median fitness evaluates the capability of an algorithm to steadily produce high-performing robots, which could be relevant in practical deployment. Though not adopted in previous literature, we find such a metric beneficial for exhibiting that MorphVAE is much more informed during design search thanks to its full use of evaluated robot designs, and thus better distinguishing MorphVAE from existing EAs. As depicted in Figure 11, MorphVAE-H notably outperforms all of its counterparts in 5 out of 6 tasks, showcasing its consistent capability to propose well-performing robot designs.

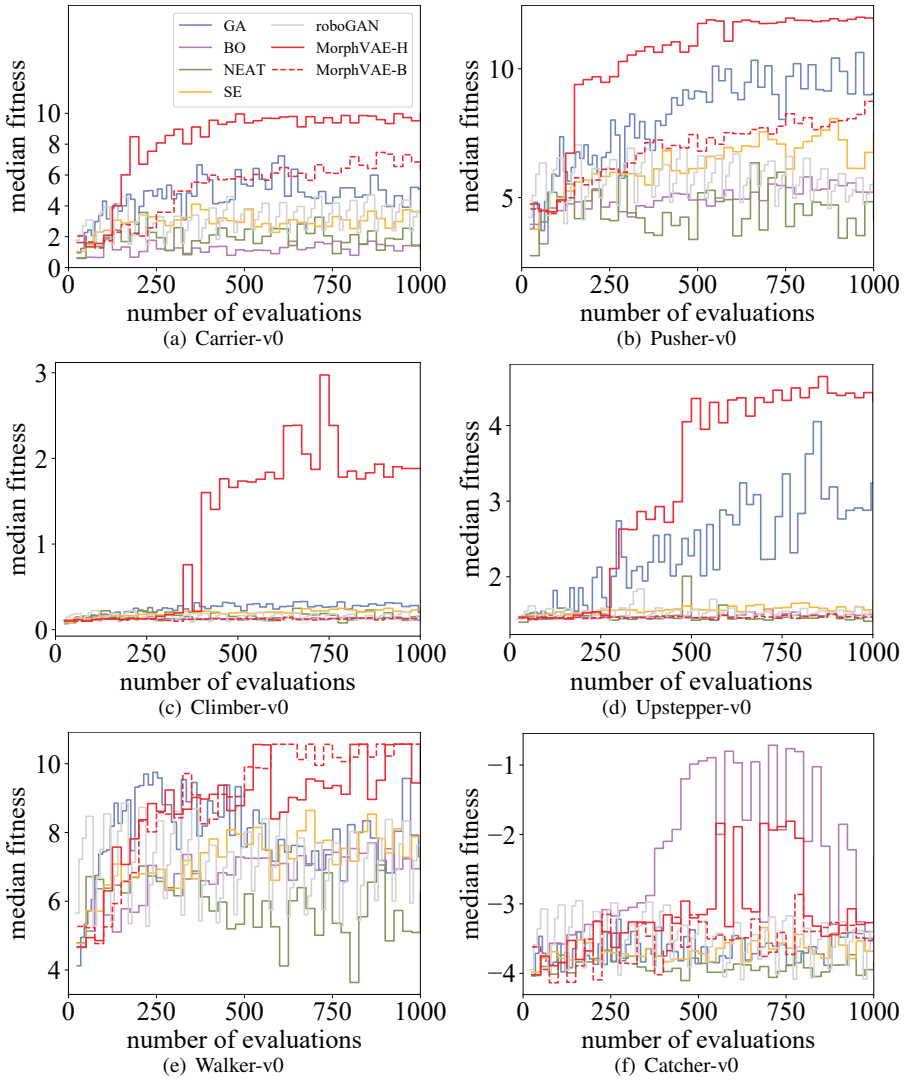

Figure 11: Performance comparison of median fitness