RoboLDA: Probabilistic Model for Uncovering the Embodied Hierarchical Structure of Robot Morphology
Supplementary Materials

Appendix A. The Derivation of Evidence Lower Bound

For purpose of illustration, here we present the detailed derivation of Evidence Lower Bound (ELBO) for a simplified version of RoboLDA. Specifically, we ignore the spatial positions of different voxels, but rather treat them as homogeneous. In this way, the "global" latents Θ_2 through Θ_4 in the paper would all reduce to two-dimensional matrices, of sizes $H \times A$, $A \times K$ and $K \times V$, respectively, with H, A, K and V denoting the total numbers of task types, robot types, organ types and voxel types. The original version of RoboLDA has a much more involved ELBO that is nearly impossible to derive by hand. Therefore, we resort to the automatic calculation of `Pyro` (a Python package devoted to probabilistic programming) for its variational inference.

Assume there are a total number of D robot samples. Denote their morphologies (eg. voxel matrices for voxel-based soft robots) as $\boldsymbol{X} = (X_1, \cdots, X_D)$, and their task types as $\boldsymbol{h} = (h_1, \cdots, h_D)$. Note that we altered some notations compared with those in the paper to ensure a concise derivation, but the notations in this appendix are still self-consistent.

The marginal likelihood function of observable variables, $\boldsymbol{X}$ and $\boldsymbol{h}$, is written as:

$$
\begin{aligned}
p(\boldsymbol{h}, \boldsymbol{X}) = \int \prod_{h=1}^{H} p(g_h) \prod_{a=1}^{A} p(\theta_a) \prod_{k=1}^{K} p(\phi_k) \cdot \\
\prod_{d=1}^{D} p(h_d) \left[\int p(Y_d|g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}}) \mathrm{d}Z_d \mathrm{d}Y_d \right] \mathrm{d}g \mathrm{d}\theta \mathrm{d}\phi,
\end{aligned}
$$

where g_h, θ_a and ϕ_k denote the parameters of multinomial distributions over robot types, organ types and material types, given a specific task type h, robot type a and organ type k, respectively. They all follow Dirichlet prior distributions. $p(h_d)$ is the prior of task type. Y_d stands for the robot type of the d-th robot. Z_{di} and X_{di} stand for the organ type and material type in the i-th voxel position, respectively. Y_d, Z_{di} and X_{di} follow multinomial distributions with g_{h_d}, θ_{Y_d} and $\phi_{Z_{di}}$ as parameters. Denote (g, θ, ϕ) collectively as β, and denote its approximate posterior as $q(\beta|\boldsymbol{h}, \boldsymbol{X})$.

Let us denote the following formula compactly as $p(\boldsymbol{h}, \boldsymbol{X}|\beta)$:

$$
\prod_{d=1}^{D} p(h_d) \left[\int p(Y_d|g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}}) \mathrm{d}Z_d \mathrm{d}Y_d \right]. \tag{0.1}
$$

According to Jensen's Inequality, we have

$$
\begin{aligned}
\log p(\boldsymbol{h}, \boldsymbol{X}) &= \log \int p(\beta) p(\boldsymbol{h}, \boldsymbol{X}|\beta) \mathrm{d}\beta \\
&= \log \int q(\beta|\boldsymbol{h}, \boldsymbol{X}) \cdot \frac{p(\beta) p(\boldsymbol{h}, \boldsymbol{X}|\beta)}{q(\beta|\boldsymbol{h}, \boldsymbol{X})} \mathrm{d}\beta \\
&\geq \int q(\beta|\boldsymbol{h}, \boldsymbol{X}) \log \frac{p(\beta) p(\boldsymbol{h}, \boldsymbol{X}|\beta)}{q(\beta|\boldsymbol{h}, \boldsymbol{X})} \mathrm{d}\beta \\
&= \mathbb{E}_{\beta \sim q(\beta|\boldsymbol{h}, \boldsymbol{X})} \log \frac{p(\beta) p(\boldsymbol{h}, \boldsymbol{X}|\beta)}{q(\beta|\boldsymbol{h}, \boldsymbol{X})}.
\end{aligned} \tag{0.2}
$$

Now we proceed to address $p(\boldsymbol{h}, \boldsymbol{X}|\beta)$. Denote the approximate posterior of Y_d as $q(Y_d|h_d, X_d)$. Hence, according to Jensen's Inequality, we have

$$\begin{aligned}
&\log \int p(Y_d|h_d, g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}}) \mathrm{d}Z_d \mathrm{d}Y_d \\
&= \log \int q(Y_d|h_d, X_d) \frac{\int p(Y_d|g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}}) \mathrm{d}Z_d}{q(Y_d|h_d, X_d)} \mathrm{d}Y_d \\
&\geq \int q(Y_d|h_d, X_d) \log \frac{\int p(Y_d|g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}}) \mathrm{d}Z_d}{q(Y_d|h_d, X_d)} \mathrm{d}Y_d \\
&= \mathbb{E}_{Y_d \sim q(Y_d|h_d, X_d)} \log \frac{\int p(Y_d|g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}}) \mathrm{d}Z_d}{q(Y_d|h_d, X_d)}.
\end{aligned} \tag{0.3}$$

Further denote the approximate posterior of Z_d as $q(Z_d|h_d, X_d, Y_d)$. By invoking Jensen's Inequality once more, if follows

$$\begin{aligned}
&\log \int p(Y_d|g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}}) \mathrm{d}Z_d \\
&= \log \int q(Z_d|h_d, X_d, Y_d) \cdot \frac{p(Y_d|g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}})}{q(Z_d|h_d, X_d, Y_d)} \mathrm{d}Z_d \\
&\geq \int q(Z_d|h_d, X_d, Y_d) \log \frac{p(Y_d|g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}})}{q(Z_d|h_d, X_d, Y_d)} \mathrm{d}Z_d \\
&= \mathbb{E}_{Z_d \sim q(Z_d|h_d, X_d, Y_d)} \log \frac{p(Y_d|g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}})}{q(Z_d|h_d, X_d, Y_d)}.
\end{aligned} \tag{0.4}$$

Substituting (0.4) into (0.3), we have

$$\begin{aligned}
&\log \int p(Y_d|h_d, g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}}) \mathrm{d}Z_d \mathrm{d}Y_d \\
&\geq \mathbb{E}_{\substack{Y_d \sim q(Y_d|h_d, X_d) \\ Z_d \sim q(Z_d|h_d, X_d, Y_d)}} \log \frac{p(Y_d|g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}})}{q(Y_d|h_d, X_d) q(Z_d|h_d, X_d, Y_d)}.
\end{aligned}$$

Further substituting the above formula into (0.1), we obtain

$$\begin{aligned}
\log p(\boldsymbol{h}, \boldsymbol{X}|\beta) &= \sum_{d=1}^{D} \left[\log p(h_d) + \log \int p(Y_d|h_d, g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}}) \mathrm{d}Z_d \mathrm{d}Y_d \right] \\
&\geq \sum_{d=1}^{D} \left[\log p(h_d) + \mathbb{E}_{\substack{Y_d \sim q(Y_d|h_d, X_d) \\ Z_d \sim q(Z_d|h_d, X_d, Y_d)}} \log \frac{p(Y_d|g_{h_d}) \prod_{i=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}})}{q(Y_d|h_d, X_d) q(Z_d|h_d, X_d, Y_d)} \right].
\end{aligned} \tag{0.5}$$

Eventually, substitute (0.5) into (0.2) and we end up with the ELBO of the simplified version of RoboLDA:

$$\begin{aligned}
\log p(\boldsymbol{h}, \boldsymbol{X}) \geq &\sum_{d=1}^{D} \log p(h_d) + \mathbb{E}_{\beta \sim q(\beta|\boldsymbol{h}, \boldsymbol{X})} \left[\log p(\beta) - \log q(\beta|\boldsymbol{h}, \boldsymbol{X}) \right] \\
&+ \sum_{d=1}^{D} \left[\mathbb{E}_{\substack{\beta \sim q(\beta|\boldsymbol{h}, \boldsymbol{X}) \\ Y_d \sim q(Y_d|h_d, X_d) \\ Z_d \sim q(Z_d|h_d, X_d, Y_d)}} \log \frac{p(Y_d|g_{h_d}) \prod_{d=1}^{N} p(Z_{di}|\theta_{Y_d}) p(X_{di}|\phi_{Z_{di}})}{q(Y_d|h_d, X_d) q(Z_d|h_d, X_d, Y_d)} \right].
\end{aligned}$$

With this, the derivation is finished.

Appendix B. Hyperparameter Settings

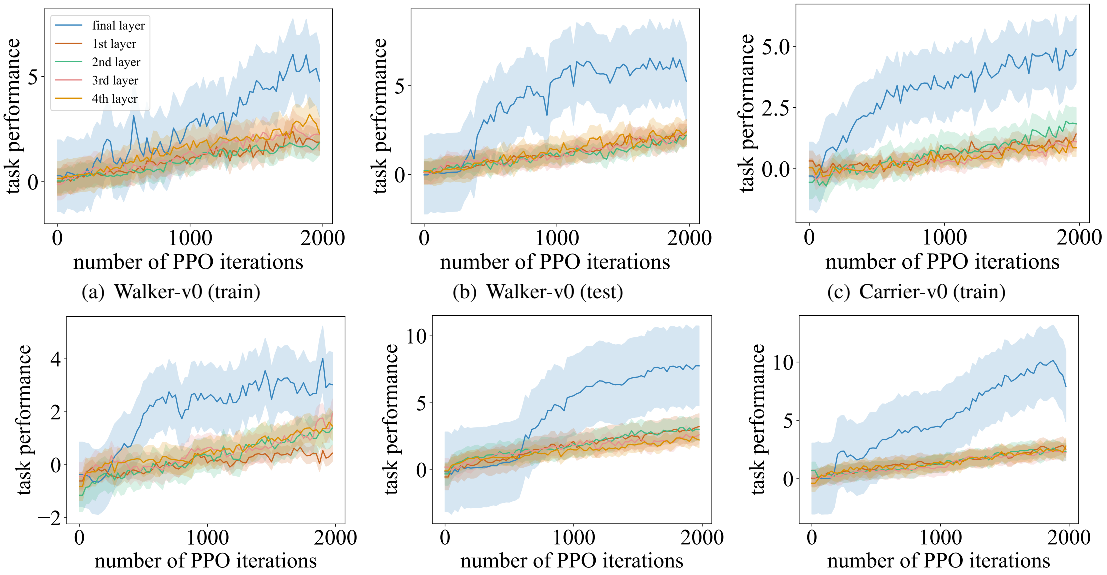
For reproducibility, in this section we list all our hyperparameters adopted for RoboLDA as well as the PPO algorithm. For more implementation details, including those of baselines, please refer to our GitHub codebase.

Table 1: Hyperparameter settings

hyperparameter	value
RoboLDA	
robot size	5×5
number of robot types	6
number of organ types	6
number of organ components	10
hidden dims of MLPs	[128]*3
learning rate	10^{-4}
number of training epochs	400
β_1 and β_2 in ADAM optimizer	(0.95,0.999)
PPO Policy Training	
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 (monolithic control/modular control)	1000/2000
number of epochs per iteration	4
number of mini-batches per epoch	4
λ in generalized advantage estimation (GAE)	0.95

Appendix C. Additional Results of Ablation Study

In this section, we report the results of ablation study on the remaining three tasks, which have not fit in the paper due to space limit. The results still show that applying the organ masks to the last self-attention layer yields the optimal performance, thus justifying our design choice. As explained in Section 4.3 of our paper, this is largely due to the fact that the last layer is the closest to the output of actuation signals and therefore brings the benefit of organs into the fullest play.



(a) Walker-v0 (train)
(b) Walker-v0 (test)
(c) Carrier-v0 (train)
(d) Carrier-v0 (test)
(e) Pusher-v0 (train)
(f) Pusher-v0 (test)

Figure 1: Results of ablation study on Walker-v0, Carrier-v0 and Pusher-v0.

Appendix D. Introduction to Task Environments

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

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

- **Position:** Denote with p^o the position of the center of mass of an object o, which consists of two components p_x^o and p_y^o, 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_x^o and v_y^o, 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.

D.1 Walker-v0

Figure 2: 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.

D.2 Pusher-v0

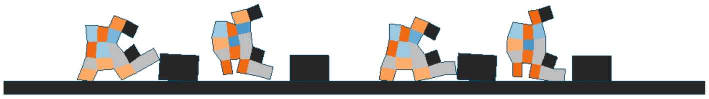


Figure 3: Pusher-v0

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.

D.3 Carrier-v0

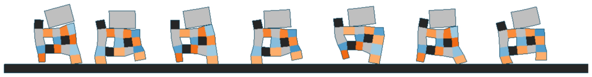


Figure 4: Carrier-v0

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.

D.4 BridgeWalker-v0

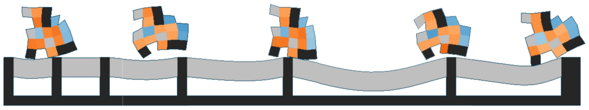


Figure 5: BridgeWalker-v0

In this task, the robot is required to walk as far as possible on a soft rope-bridge. $\mathcal{S} \in \mathbb{R}^{n+3}$ consists of v^r, θ^r and c^r with lengths 2, 1 and n respectively. $R = \Delta p_x^r$ 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.

D.5 UpStepper-v0

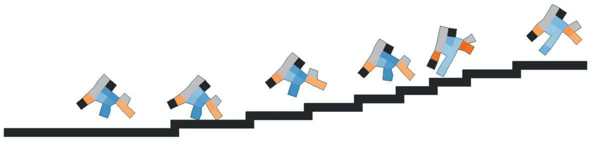


Figure 6: UpStepper-v0

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).

D.6 DownStepper-v0

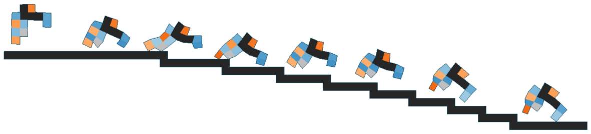


Figure 7: DownStepper-v0

In this task, the robot is required to climb down 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 90 degrees from its original orientation in either direction (after which the environment is reset).