# Supplementary Material of

"Homeostatic Reinforcement Learning through Soft Behavior Switching with Internal Body State"

Naoto Yoshida[1], Hoshinori Kanazawa[1,2], and Yasuo Kuniyoshi[1,2]

[1]The University of Tokyo
[2]Next Generation Artificial Intelligence Research Center

## A Hyperparameters of PPO

Our implementation of Proximal Policy Optimization (PPO) is based on CleanRL [1]. In the original PPO implementation, the return after terminal states is forced to zero. In homeostatic RL, however, we have found that this treatment of terminal states has a negative effect. This is because the idea of resetting in homeostatic RL is to prevent the agent's behavior from deviating from the set-point and to end an unsuccessful episode. Moreover, forcing the future returns to zero means that the agent is fixed at the setpoint! To avoid this problem, we removed the zero-forcing at the end state and instead used the value estimate as the approximate future return.

Table 1: PPO hyperparameters used in experiments

| Hyperparameter | Value |
|---|---|
| SGD update epochs | 30 |
| Learning rate (initial) | $3 \times 10^{-4}$ |
| Adam epsilon | $10^{-5}$ |
| Maximum gradient norm | 0.5 |
| Discount factor ($\gamma$) | 0.99 |
| GAE lambda ($\lambda$) | 0.95 |
| Clipping parameter ($\epsilon$) | 0.3 |
| Value loss clipping parameter | 10 |
| Value loss coefficient | 0.5 |
| Entropy coefficient | $10^{-3}$ |
| # of minibatches | 6 |
| # of sampler threads | 10 |

Table 2: Sampling Parameters in *Line* Environment

| Hyperparameter | Value |
|---|---|
| Total time steps | $5 \times 10^6$ |
| # of rollout samples per thread in a single iteration | $5 \times 10^3$ |

Table 3: Sampling Parameters in *Field* Environment

| Hyperparameter | Value |
|---|---|
| Total time steps | $1.5 \times 10^6$ |
| # of rollout samples per thread in a single iteration | $5 \times 10^3$ |

# B Details of Environments for Benchmarking Homeostatic RL

We used the same homeostatic reward setting in all environments. Following previous studies [2], we used the reward scaling

$$\tilde{r}_h(s_i, s_i') = 100\,(D(s_i) - D(s_i')) \tag{1}$$

to facilitate learning speed.

## B.1 Line Environment

In this environment, the agent can move in a one-dimensional space by applying a force to the point. The dynamics of the agent's position is defined as follows:

$$\delta y_t = \mathrm{clip}(k_1 \delta y_t + k_2 a_t, -k_3, k_3) \tag{2}$$

$$y_t = \mathrm{clip}(y_t + v_t, 0, n_{\mathrm{resource}} + 1) \tag{3}$$

where $k_1 = 0.3$, $k_2 = 0.5$, $k_3 = 2$, $v_t$ is the velocity of the agent at time step $t$, $y_t$ is the position of the agent at the corresponding time step, $\mathrm{clip}(x, a, b)$ is the clipping function that restricts $x$ to the range $[a, b]$ by $\mathrm{clip}(x, a, b) = a$ if $x < a$ and $\mathrm{clip}(x, a, b) = b$ if $b < x$. In addition, the agent has a $n_{\mathrm{resource}}$ dimensional continuous nutrient state and the environment has the corresponding number of resource supply locations (orange, blue, and green areas in the panel). The space for the agent's position is $y \in [0, n_{\mathrm{resource}} + 1]$ and each resource supply site is placed at $y^j_{\mathrm{res}} = j$, where $j$ is the index of the corresponding resource dimension ($j \in \{0, 1, \ldots, n_{\mathrm{resource}} - 1\}$). The agent can recharge the predefined

Table 4: Sampling Parameters in *TRP* and *Thermal* Environment

| Hyperparameter | Value |
|---|---|
| Total time steps | $8 \times 10^7$ |
| # of rollout samples per thread in a single iteration | $3 \times 10^4$ |

amount of the associated resource at these resource locations. Specifically, the agent's resource dynamics are described as follows.

$$\begin{aligned} \delta \boldsymbol{z}_t^{\text{inlet}} &= k_4 \delta \boldsymbol{z}_t^{\text{inlet}} + k_5 \boldsymbol{i}_t, && (4) \\ \delta \boldsymbol{z}_t &= \text{clip}(\delta \boldsymbol{z}_t^{\text{inlet}}, -k_6, k_6) - k_7, && (5) \\ \boldsymbol{z}_t &= \boldsymbol{z}_t + \delta \boldsymbol{z}_t, && (6) \end{aligned}$$

where $k_4 = 0.5$, $k_5 = 0.02$, $k_6 = 0.1$, $k_7 = 0.005$, and $\boldsymbol{i}_t$ is the indicator function, where in the $j$-th element of the vector $\boldsymbol{i}_t^j = 1$ if $y_t \in [y_{\text{res}}^j - 0.2, y_{\text{res}}^j + 0.2]$, otherwise $\boldsymbol{i}_t^j = 0$. To summarize, the action space is a continuous space $a \in [-1, +1]$ and the state space is an eight-dimensional continuous space when $n_{\text{res}} = 3$, which consists of

$$\begin{aligned} s &\triangleq (y/(n_{\text{resource}} + 1), \delta y/2, \delta \boldsymbol{z}, s_i), && (7) \\ s_i &\triangleq \boldsymbol{z}. && (8) \end{aligned}$$

When the environment is reset, the agent's position $y$ and nutrient state $z$ are randomly initialized, while the velocity of each state $\delta y$ and $\delta z$ is initialized with zero values. The initial $y_0$ and $z_0$ are sampled from a uniform distribution $y_0 \sim \mathcal{U}(0, n_{\text{resource}} + 1)$ and $z_0^j \sim \mathcal{U}(-1/6, 1/6)$ for all $j \in \{1, 2, \ldots, n_{\text{resource}}\}$. A single episode is terminated if any dimension of the internal state of the nutrient exceeds the range $|z_t^j|_2 < 1$. The environment is then reset accordingly. Finally, the reward in this environment is defined by $r = \tilde{r}_h$.

## B.2 Field Environment

This is the homeostatic RL environment with two-dimensional space. The agent can move freely in the square area by applying the force to the point. The dynamics of the environment are the same as in the *Line* environment, with the differences that the position $\boldsymbol{y}$ is a two-dimensional space, and the resource supply locations are also two-dimensional. In this environment, the agent's position is constrained to the range $\boldsymbol{y}_t \in [-1, 1]^2$. The behavior of the indicator function $\boldsymbol{i}$ is different from the *Line* environment. The $j$-th element of the vector becomes $\boldsymbol{i}_t^j = 1$ if the distance between the agent's position and the $j$-th resource site is sufficiently close ($\|\boldsymbol{y}_t - \boldsymbol{y}_{\text{res}}^j\|_2 \leq 0.2$), otherwise $\boldsymbol{i}_t^j = 0$. Each resource site is placed at the position $\boldsymbol{y}_{\text{res}}^j = (0.4\cos(\kappa j), 0.4\sin(\kappa j))^\top$ for the $j$-th resource type, where $\kappa = 2\pi/n_{\text{resource}}$. Finally, the internal nutrient dynamics parameter uses $k_7 = 0.001$. This environment has a two-dimensional continuous action space $a \in [-1, +1]^2$ and the state space is the ten-dimensional continuous space when $n_{\text{resource}} = 3$, which consists of

$$\begin{aligned} s &\triangleq (\boldsymbol{y}, \delta \boldsymbol{y}/2, \delta \boldsymbol{z}, s_i), && (9) \\ s_i &\triangleq \boldsymbol{z}. && (10) \end{aligned}$$

The reward definition is same with *Line* environment.

## B.3 Two-Resource Problem Environment

This environment contains an agent and randomly distributed food resources in the field. The agent was developed using the *Mujoco* dynamics simulator [3]. The environment contains a quadruped robot *Ant* [4] in a food gathering environment [5,6]. We used a *low-gear* Ant model asset as suggested by previous studies [2, 6]. The agent had a two-dimensional continuous nutritional state $\boldsymbol{z}$ corresponding to its nutritional level. Two types of food resources (represented by four red balls and six blue balls) corresponded to the dimensions of the agent's nutritional state. When the position of the agent's torso (body sphere) comes within the range of a food resource ($<$ 1 m in the simulator), the food is consumed and the nutritional state is recharged with a predefined amount. New food resources are then generated at random positions. We used a simple metabolic model of the nutritional state described in a previous study [2, 7]. The updating of the nutritional state $\boldsymbol{z}$ is described as

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + k_1^{\mathrm{TRP}} \boldsymbol{i}_t - k_2^{\mathrm{TRP}}, \tag{11}$$

where $k_1^{\mathrm{TRP}}$ is the scaling factor for food input, $k_2^{\mathrm{TRP}}$ is the default resource consumption. $\boldsymbol{i}_t^j$ is the indicator function of food capture for $j$-th resource at time step $t$. We used the $k_1^{\mathrm{TRP}} = 0.1$ and $k_2^{\mathrm{TRP}} = 0.00015$ as suggested in [2,7]. During the training phase, a single episode starts with nutritional states uniformly sampled from $\mathcal{U}[-\frac{1}{6}, \frac{1}{6}]$ for each nutrient. As in the previous environments, an episode is terminated if any dimension of the internal state of the nutrient exceeds the range $|z_t^j|_2 < 1$. The action of the agent was the motor torque of each joint. The dimension of the control was eight, and the control space was normalized to the eight-dimensional cube $a \in [-1, 1]^8$. The agent's observations consist of a 40-dimensional exteroception vector $\boldsymbol{s}_e$, a 27-dimensional proprioception vector $\boldsymbol{s}_p$, and a two-dimensional interoception vector $\boldsymbol{s}_i$. Therefore,

$$s \triangleq (\boldsymbol{s}_e, \boldsymbol{s}_p, \boldsymbol{s}_i), \tag{12}$$

$$s_i \triangleq \boldsymbol{z}. \tag{13}$$

Here, exteroception refers to the perceptual signals from stimuli outside the body, consisting of range sensor stimuli (20 different directions around the agent) for two types of food resources. Proprioception is the 27-dimensional observation that reports the self-motion and positions of the agent's body; in this study, it was assumed to include the agent's joint angles, rotational and positional speed, height, and trunk posture information [4]. Interoception is the two-dimensional internal environment of the agent $s_i \triangleq \boldsymbol{z}$. The reward in this environment is defined by a homeostatic term in addition to the cost terms

$$r = \tilde{r}_h - 0.005\|\tilde{\boldsymbol{p}} - \tilde{\boldsymbol{p}}^*\|_2^2 - 0.001\|\boldsymbol{a}\|_2^2, \tag{14}$$

where $\tilde{p}$ and $\tilde{p}^*$ are the rotation angles of the agent's torso and its target value (the agent's torso standing vertically).

## B.4 Thermal Regulation Environment

In this environment, the agent must regulate its core body temperature through environmental interactions while maintaining energy homeostasis. The environment is similar to that of *TRP*. Six food resources were randomly distributed over a square area with randomly generated terrain. We used temperature dynamics in addition to one-dimensional nutrition dynamics. The core temperature dynamics were obtained from a previous study [2] based on the thermal dynamics of a reptile [8, 9] and the thermal dynamics of an electric motor [10]. The dynamics of the body temperature $\tau$ is described by the differential equation $C\frac{d\tau}{dt} = \delta Q(\tau, u, u_{\mathrm{ev}})$, where $\tau$ is the agent's core body temperature, $C$ is the agent's representative heat capacity, and $\delta Q$ is the amount of heat added to the agent's body. A detailed calculation of $\delta Q$ is provided in [2].The agent performed a nine-dimensional action $a \in [-1, 1]^9$ consisting of an eight-dimensional motor output and a one-dimensional evaporative action. We used the normalized body temperature by mapping $\tau \in [307, 315]$ in Kelvin degrees to $\tilde{\tau} = [-1, 1]$, and the normalized body core temperature setpoint was set to zero. The agent receives the same exteroception $s_e$ and proprioception $s_p$ as in the *TRP* environment, where the rangefinder stimulus represents the distance to food resources (red balls, 20 dimensions). The interoceptive signal $s_i$ is defined as a two-dimensional signal composed of a one-dimensional nutrient state and the normalized core temperature of the agent $s_i \triangleq (\boldsymbol{z}, \tilde{\tau})$. The full observation in this environment is

$$\boldsymbol{s} \triangleq (\boldsymbol{s}_e, \boldsymbol{s}_p, \boldsymbol{s}_i), \tag{15}$$

$$\boldsymbol{s}_i \triangleq (z, \tilde{\tau}). \tag{16}$$

In the thermal environment, the reward is as defined in the *TRP*.

# C Visualization of the dynamics of internal switching of IBS and FBS policies.

In Figure 1 we have visualized the behavior of the mixing ratio $\boldsymbol{u}$ in policy architectures using the *TRP* environment. The horizontal axes show 500 consecutive time steps in the environment, and the vertical axes represent a mixture ratio of policies. The colors of the bar represent the index of multiple behavior modules (value embeddings). The top panel shows an example of behavior switching dynamics in the IBS policy. The bottom panel shows the switching dynamics of the FBS policy.

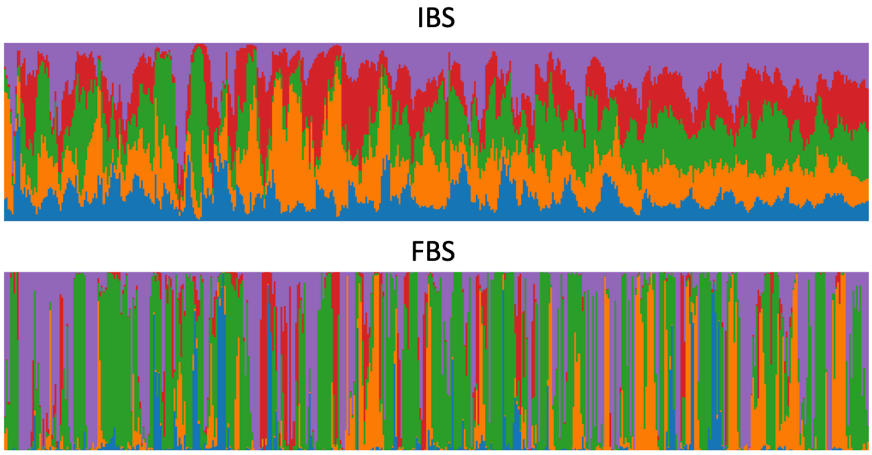


Figure 1: The vertical axis represents the ratio of policies $\boldsymbol{u}$. The color of the bar represents the index of the multiple behavior modules. Horizontal axes show 500 consecutive time steps. Top: Example of IBS switching dynamics. Bottom: Example of FBS switching dynamics.

# D Additional Studies

We performed the ablation study using the *Line* environment. The performance (IQM + CIs) was obtained using the same procedure as in the main text, except for the change in the number of experiments (10 experiments in ablation studies). Figure 2 shows the results of the ablation study on the IBS architecture. In the top panels, we compared the proposed IBS architecture ($n$ models with 256 hidden nodes, weighted sum, LayerNorm) with two models without attention mechanisms (*Averager* and *Flat*). *Averager*; the model outputs the average of $n$ models, and *Flat*; a single flat model with $256 \times n$ hidden nodes.

The bottom panels show the comparison between the IBS and the IBS architecture without layer normalization. We observe the performance degradation when the layer normalization is omitted from the IBS. This result may suggest the potential benefit of layer normalization with other architectures in the homeostatic RL domain. However, it needs to be discussed with further investigations in other homeostatic RL environments.


Figure 2: The results of ablation study.

# E Results of the environments with different number of resources

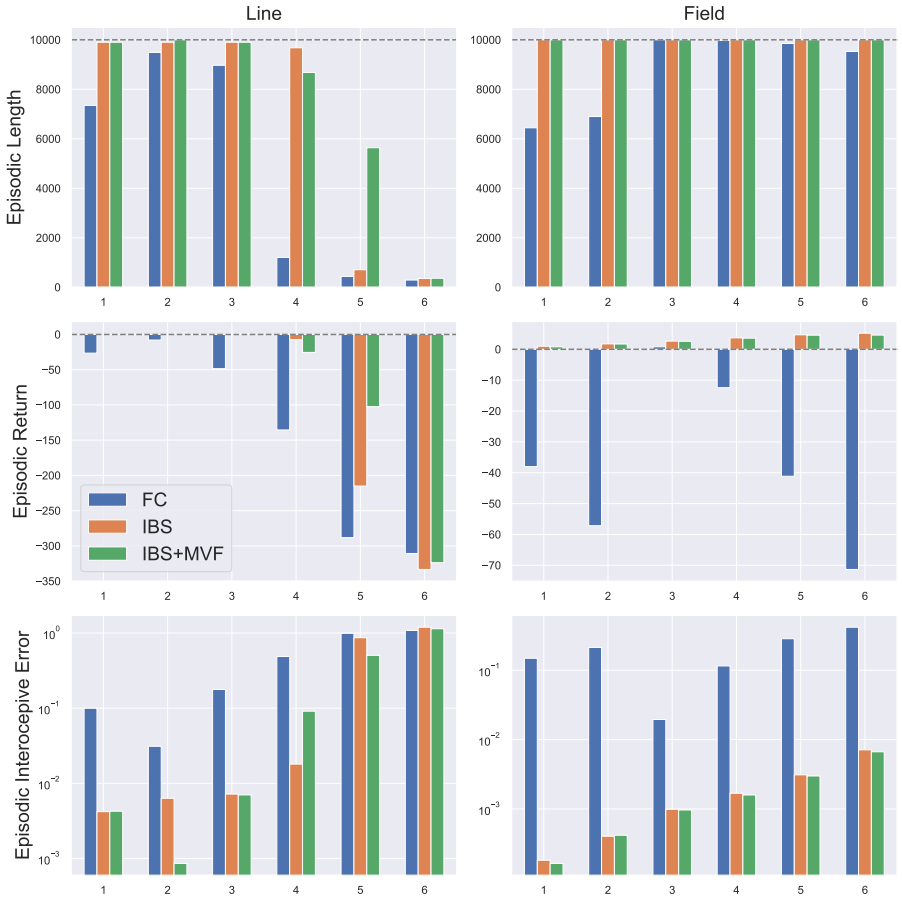


Figure 3: Evaluation of the Modular Value Function architecture (MVF) in homeostatic RL environments with different numbers of resource types.

# F Evaluation of IBS+MVF in benchmark environments

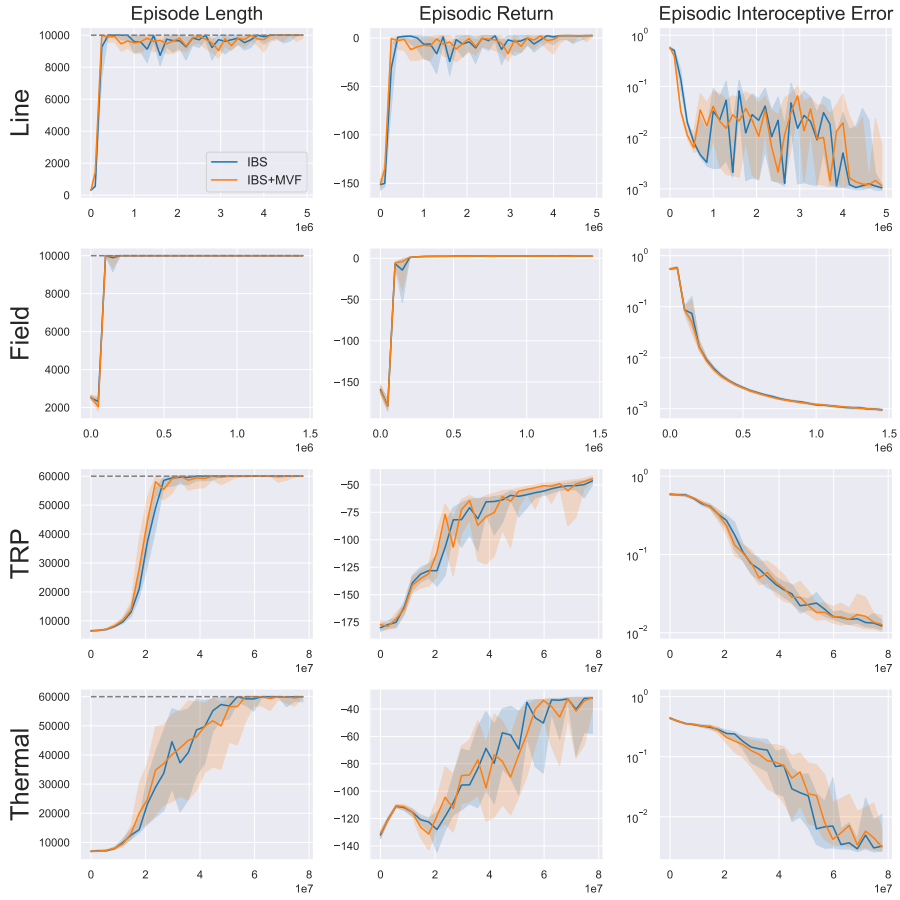


Figure 4: Evaluation of Interoceptive Behavior Switching (IBS) and Modular Value Function architecture (IBS+MVF) in four homeostatic RL environments.

# References


[1] S. Huang, R. F. J. Dossa, C. Ye, J. Braga, D. Chakraborty, K. Mehta, and J. G. Araújo, "Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms," *Journal of Machine Learning Research*, vol. 23, no. 274, pp. 1–18, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1342.html

[2] N. Yoshida, T. Daikoku, Y. Nagai, and Y. Kuniyoshi, "Embodiment perspective of reward definition for behavioural homeostasis," in *Deep RL Workshop NeurIPS 2021*, 2021.

[3] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[4] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *International Conference on Learning Representations (ICLR)*, 2016.

[5] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International conference on machine learning*. PMLR, 2016, pp. 1329–1338.

[6] A. Li, C. Florensa, I. Clavera, and P. Abbeel, "Sub-policy adaptation for hierarchical reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2020.

[7] G. Konidaris and A. Barto, "An adaptive robot motivational system," in *From Animals to Animats 9*. Springer, 2006, pp. 346–356.

[8] W. P. Porter, J. W. Mitchell, W. A. Beckman, and C. B. DeWitt, "Behavioral implications of mechanistic ecology," *Oecologia*, vol. 13, no. 1, pp. 1–54, 1973.

[9] T. Fei, A. K. Skidmore, V. Venus, T. Wang, M. Schlerf, B. Toxopeus, S. Van Overjijk, M. Bian, and Y. Liu, "A body temperature model for lizards as estimated from the thermal environment," *Journal of Thermal Biology*, vol. 37, no. 1, pp. 56–64, 2012.

[10] B. Venkataraman, B. F. Godsey, W. Premerlani, E. Shulman, M. Thakur, and R. Midence, "Fundamentals of a motor thermal model and its applications in motor protection," in *58th Annual Conference for Protective Relay Engineers, 2005.* IEEE, 2005, pp. 127–144.