# Hierarchical reinforcement learning with optimal level synchronization based on flow-based deep generative model

JaeYoon Kim, Junyu Xuan, *Member, IEEE* Christy Liang, *Member, IEEE* and Farookh Hussain, *Member, IEEE*

TABLE I: Key Notations.

| Symbol | Meaning |
|---|---|
| $t$ | action step |
| $\mu_{hi}$ | higher-level policy |
| $\mu_{lo}$ | lower-level policy |
| $\mu_z$ | the policy of forward part |
| $\mu_{z-\text{state}}$ | the policy of conditional part |
| $\mu_{\text{rnvp}}$ | the policy of FDGM part |
| $\mu^{-1}_{\text{rnvp}}$ | the inverse of the policy of FDGM part |
| $\Pi$ | the lower-level policy set composing of $[\mu_z, \mu_{z-\text{state}}, \mu_{\text{rnvp}}]$ |
| $L_z$ | the loss of forward part |
| $L_{z-\text{state}}$ | the loss of conditional part |
| $L_{\text{rnvp}}$ | the loss of FDGM part |
| $L_{\text{common}}$ | the averaged loss over the sum of loss of all three policies |
| $g_t \, or \, g$ | the action of higher-level policy called a goal |
| $\tilde{g_t}$ | a goal obtained through off-policy correction for training the higher-level policy $\mu_{hi}$ |
| $a_t$ | the action of lower-level policy |
| $R_t$ | the extrinsic reward of an environment |
| $r_t$ | the intrinsic reward of lower-level policy |
| $s_t \, or \, s$ | the state of the environment |
| $a_{t,z}$ | the action of forward part |
| $a_{t,\text{rnvp}}$ | the action of FDGM part of inverse part |
| $a_{t,z-\text{state}}$ | the action of conditional part of inverse part |
| $[g, a]$ | the concatenation of $g_{1:d}$ and an action $a_{t,z-\text{state}}$ |
| $[a_{\text{rnvp}}, a_{z-\text{state}}]$ | the concatenation of an action $a_{t,\text{rnvp}}$ and an action $a_{t,z-\text{state}}$ |
| $x_{t:t+c-1}$ | the sequence $x_t, \ldots, x_{t+c-1}$ |
| $c$ | the horizon of a higher-level policy |
| $D_R$ | replay buffer |

**Algorithm 1:** Find the goal, $\tilde{g}$, using only $\mu^{-1}_{rnvp}$ without forward part (Normal version)

**Input:** $a_{z-\text{state}}$ and $a_{\text{rnvp}}$
**Output:** $\tilde{g}$
**Init:**
- Set the layer dimension of all policies including $\mu_{hi}$ and the critic of $\mu_{rnvp}$ to be the same except for the actor of $\mu_{rnvp}$
- Set the layer dimension of actor of $\mu_{rnvp}$
- Set the action dimension of $\mu_{z-\text{state}}$

**while** $c$ **do**
  $a_{z-\text{state}} \, and \, a_{\text{rnvp}} \sim D_R$ ;
  $\tilde{g} \sim \mu^{-1}_{\text{rnvp}}(a_{z-\text{state}}, a_{\text{rnvp}})$ ;

**Algorithm 2:** Find the goal, $\tilde{g}$, using only $\mu^{-1}_{rnvp}$ with forward part (Variant version)

**Input:** $s, g \, and \, a_{\text{rnvp}}$
**Output:** $\tilde{g}$
**Init:**
- Set the layer dimension of all policies (a forward part and an inverse part) inside $\Pi$ to be the same except for $\mu_{hi}$
- Set the layer dimension of $\mu_{hi}$
- Set the action dimension of $\mu_{z-\text{state}}$

**while** $c$ **do**
  $s, g \, and \, a_{\text{rnvp}} \sim D_R$ ;
  $a_z^{'} \sim \mu_z(s, g)$ ;
  $a_{z-\text{state}}^{'} \sim \mu_{z-\text{state}}(s, a_z^{'})$ ;
  $\tilde{g} \sim \mu^{-1}_{\text{rnvp}}(a_{z-\text{state}}^{'}, a_{\text{rnvp}})$

## I. Preliminary Knowledge

### A. Hierarchical reinforcement learning

The structure of HRL is composed of several unit agents stacked hierarchically. A policy in goal-conditioned HRL typically has two inputs: the state of the environment $s_t$ and a goal $g_t$, which is a temporal abstraction provided by its higher-level policy $\mu_{hi}$. A unit agent is usually constructed based on an atomic RL framework, as it functions as a policy within a level of HRL. The lowest-level policy interacts with the environment by generating an action $a_t \sim \mu_{lo}(s_t, g_t)$ based on the goal $g_t$ from its higher-level policy, where $g_t \sim \mu_{hi}$ is updated every $c$ time steps ($c$ is the horizon of $\mu_{hi}$) or determined by a fixed goal transition function $h(s_{t-1}, g_{t-1}, s_t)$. Due to the hierarchical structure, the training period $c$ of a higher-level policy $\mu_{hi}$ is longer than that of its lower-level policy $\mu_{lo}$. The higher a policy is positioned in the HRL hierarchy, the more abstract its role becomes.

The environment responds to the action $a_t$ of the lowest-level policy with a reward $R_t$ and a next state $s_{t+1}$, sampled from the reward function $R(s_t, a_t)$ and the transition function $f(s_t, a_t)$, respectively. The intrinsic reward $r_t = r(s_t, g_t, a_t, s_{t+1})$, where $r$ is a fixed parameterized reward function for the lower-level policy in HRL, is typically provided by the higher-level policy. Notably, the highest-level policy uses the environmental reward $R_t$ instead of an intrinsic reward.

Specifically, in Fig. 1, the higher-level policy devises an abstract strategy (represented by the orange arrow), while

J. Kim, J. Xuan, and F. Hussain are with the Australian Artificial Intelligence Institute (AAII) at the University of Technology Sydney, Australia, Ulitmo, 2007. E-mail: JaeYoon.Kim@student.uts.edu.au, Junyu.Xuan@uts.edu.au, Farookh.Hussain@uts.edu.au

C. Liang is with the Visualisation Institute at the University of Technology Sydney. E-mail: jie.liang@uts.edu.au

Manuscript submitted May, 2025.

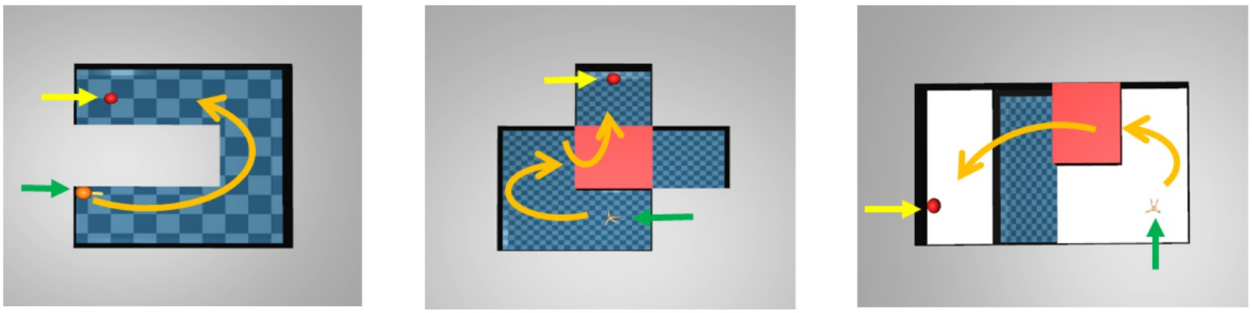


Fig. 1: An example of several tasks in the Ant environment used in this research is shown above. The target location indicated by the yellow arrow serves as the reward point for the agent's approach. The agent, represented by the green arrow, must navigate to this target. The trained policy requires the agent to: (1) combine multiple action sequences while (2) interacting with a pink obstacle block in two out of the three above tasks. These tasks include: a series of directional movements (third from the right), clearing away an obstacle (second from the right), and constructing a bridge using an object (rightmost task). The higher-level policy guides its lower-level policy by providing abstract actions represented by the orange arrow, which specify the desired positions and orientation of the ant and its limbs to achieve the reward.

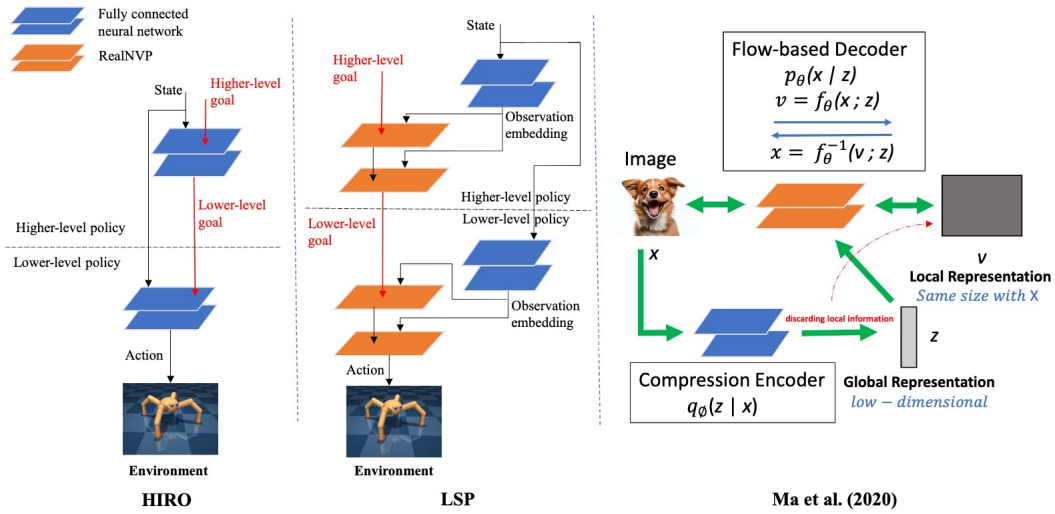



Fig. 2: The model architecture of HIRO (left), LSP (center) and Ma et al. (right)

the lower-level policy executes physical actions based on the goal from its higher-level policy and the current state. The model architectures, such as the one depicted in Fig. 2, illustrate the outlined structure of HRL.

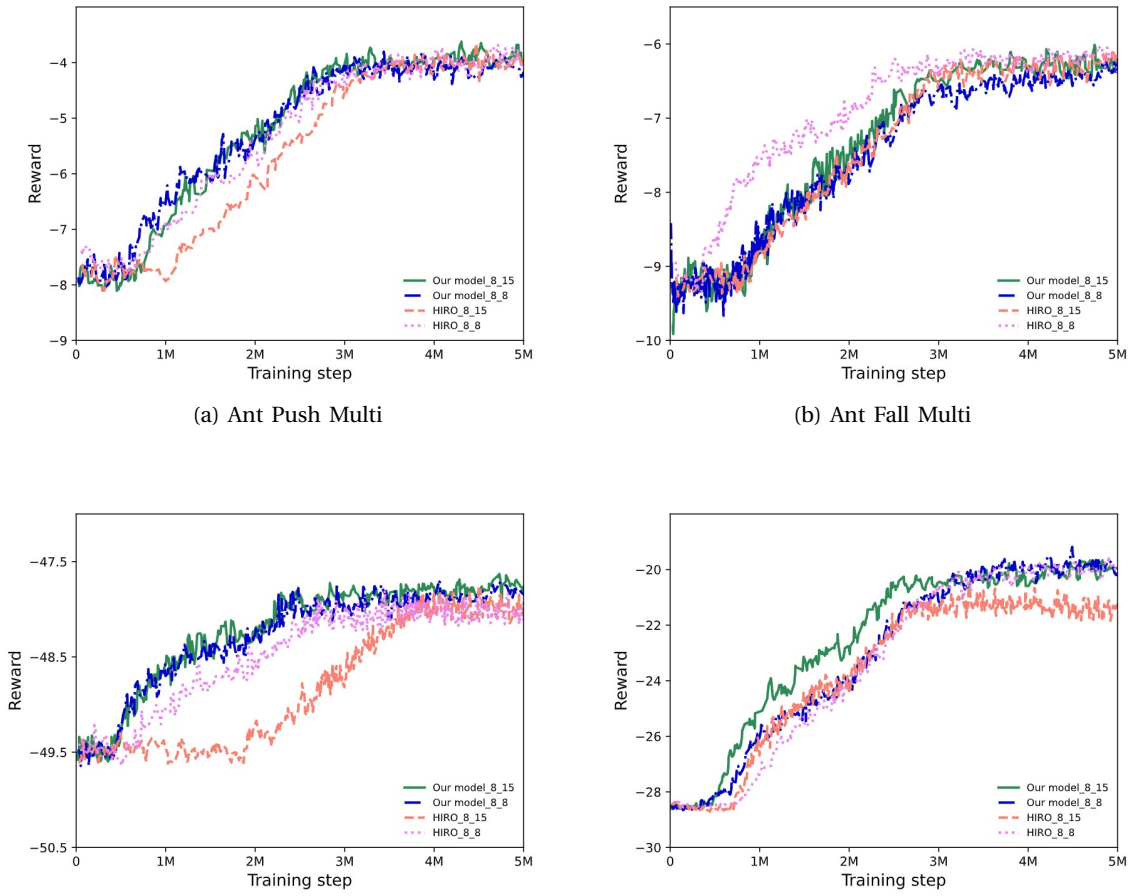
### B. Off-policy correction

It is crucial to devise an optimal training solution to enhance the performance of HRL. Off-policy methods in HRL have been proposed to address the local minimum issue associated with on-policy methods. In off-policy policy gradient methods, experience replay involves storing sampled trajectory data from past episodes in a replay buffer, which significantly improves sample efficiency. This approach enables better exploration by collecting samples using a behavior policy that differs from the target policy. The objective function of the off-policy policy gradient is defined as follows:

$$J(\theta) = \mathbb{E}_{S\sim d^\beta}\left[\sum_{a\in A} Q^\pi(s,a)\pi^\theta(a|s)\right] \tag{1}$$

where $d^\beta(s)$ is the stationary distribution of the behaviour policy $\beta(a|s)$, $\pi^\theta(a|s)$ is the target policy and $Q^\pi(s,a)$ is the action-value function estimated with respect to the target policy $\pi^\theta(a|s)$ [1].

However, the off-policy method in goal-conditioned HRL faces a non-stationary issue due to the bottom-up training approach in off-policy RL. Previously used outputs, such as goals from the higher-level policy, may no longer be suitable for training, as they do not reflect the updated parameter values $\theta$ of the newly trained lower-level policy. Consider




(a) Ant Push Multi

(b) Ant Fall Multi

(c) Ant Push Single

(d) Ant Fall Single

Fig. 3: The average rewards of our model (green, blue) for the task of the higher-level policy are compared with that of HIRO (salmon, violet) on condition of a non-optimal goal space. The numbers of (our model or HIRO)_8_(15 or 8) are the goal dimension of higher-level policy and that of lower-level policy respectively.

the operation of a lower-level policy $\mu_{lo}$ in HRL as follows:

$$a_t \sim \mu^{\theta}_{lo}(s_t, g_t), \text{ at time } t. \tag{2}$$

After $\mu^{\theta}_{lo}$ is trained for $c$ steps,

$$\begin{aligned} a_{t+c} \sim \mu^{\theta'}_{lo}(s_{t+c}, g_t), \text{ assume } s_{t+c} = s_t \\ a_t \neq a_{t+c} \text{ or } a_t = a_{t+c} \end{aligned} \tag{3}$$

where $a_{t+c}$ and $s_{t+c}$ are the action of $\mu^{\theta'}_{lo}$ and the state of the environment after $c$ steps, respectively. The action $a_{t+c}$ may differ from $a_t$ due to changes in the parameter values of $\mu^{\theta}_{lo}$ and $\mu^{\theta'}_{lo}$.

To address this issue, an off-policy correction is proposed in HIRO [2] through an indirect estimation strategy.

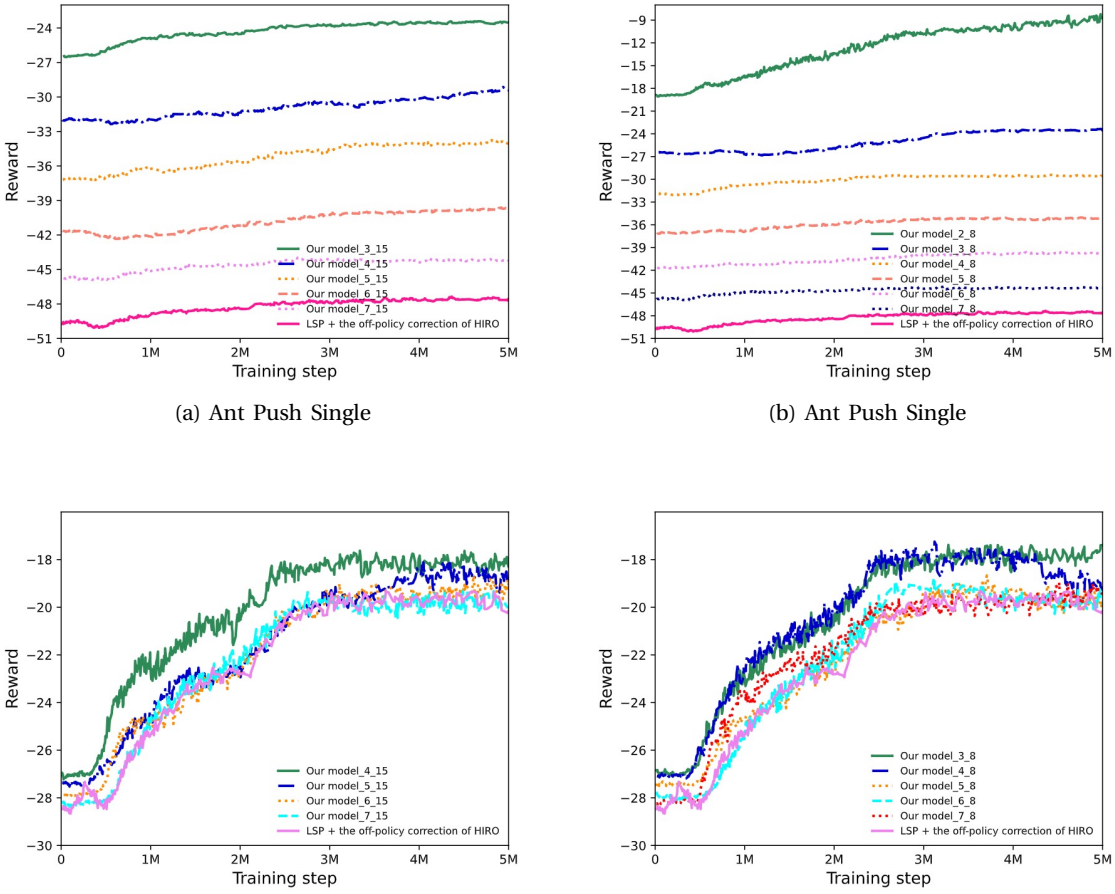
## II. Ablation Study

The importance of $a_{t,z}$ is of particular interest due to its significant role in addressing the chronic issue of FDGM, specifically the biased log-density problem. To explore its impact on our model's performance, we investigate how replacing $a_{t,z}$ with the goal $g_t$ as the input to the conditional part affects the model's performance.

- Ant Fall Multi of variant model – Actor of FDGM : 150, All other NNs : 180, $a_{t,z-\text{state}} = 19$

In this experiment, we do not set the parameters of each policy to match those of our model in the comparative analysis. Instead, we first identify the parameters of the variant model with the optimal goal space to achieve its best performance in the Ant Fall Multi task from sub-section A. Comparative analysis with the optimal goal space Section IV of manuscript. We then compare the performance of our original model from the previous Ant Fall Multi experiment with the optimal goal space to the variant model with the newly determined parameters.

The results show that while both models perform similarly until 3.5M, our original model marginally outperforms the variant model up to 7.5M. After this point, both models exhibit similar performance up to 10M. The performance comparison between our model and the variant model is illustrated in Fig. 5.




(a) Ant Push Single

(b) Ant Push Single

(c) Ant Fall Single

(d) Ant Fall Single

Fig. 4: For the task of the higher-level policy, the average rewards of our model (green, blue, orange, salmon, violet and navy for Ant Push Single and green, blue, orange, cyan and red for Ant Fall Single) are compared with that of LSP (pink for Ant Push Single and violet for Ant Fall Single). They are based on the change of higher-level goal dimension of a) our model_8_15, (b) our model_8_8 (c) our model_8_15 (d) our model_8_8 used in the corresponding task of Fig. 3.

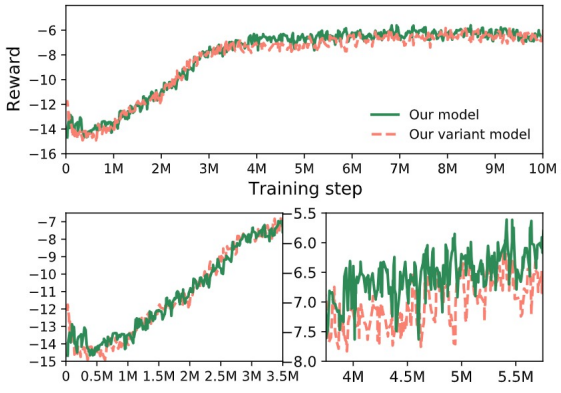



Fig. 5: Average result of our method (green) compared with that of the variant model (salmon) in Ant Fall Multi

## APPENDIX A
## THE TRAINING PARAMETERS FOR OUR MODEL

Fig. 6 illustrates an example of creating a non-optimal goal space for the higher-level and lower-level policies of our model, starting from the default optimal goal space tuples of the higher-level and lower-level policies in HIRO.

Additionally, the training parameters for our model and the key points for the experiments are summarized as follows.

The training parameters defined in our model for all tasks are the same as those of HIRO, as follows:

- Discount $\gamma$ = 0.99 for both controllers.
- Adam optimizer; actor learning rate 0.0001; critic learning rate 0.001.
- Soft update targets $\tau$ = 0.005 for both controllers.
- Replay buffer of size 200,000 for both controllers.
- Lower-level train step and target update performed every 1 environment step.
- Higher-level train step and target update performed every 10 environment steps.

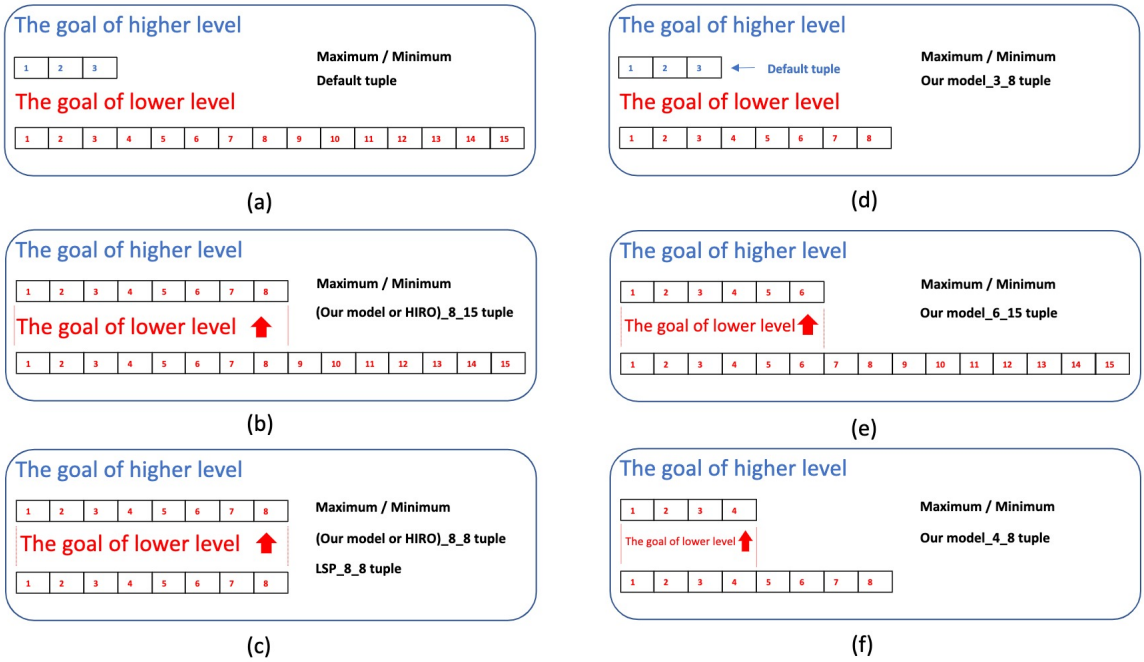



Fig. 6: An example of how to create a non-optimal goal space for the higher-level and lower-level policies of our model, starting from the default optimal goal space tuples of HIRO's higher-level and lower-level policies. Each level's default tuple in (a) is represented with a color—blue for the higher-level policy or red for the lower-level policy—and the position number of the tuple element. Only the tuple elements of the destination which is higher-level's goal are replaced by the corresponding tuple elements of the source which is lower-level's goal. In the case of (Our model, HIRO, or LSP)_8_8 for the lower-level's goal, the last 7 tuple elements of the lower-level goal are removed. (a) The default tuple, which is the optimal tuple, for the higher-level and lower-level policies of HIRO. Appendix A provides the definition of each level's tuple. (b) Our model_8_15 (c) Our model_8_8 (d) Our model_3_8: The goal space of the higher-level policy for 'Our model_2_8' and 'Our model_3_8' is constrained to match the default goal space of Ant Push Single and Ant Fall Single in HIRO, respectively. (d) Our model_6_15 (e) Our model_4_8

- No gradient clipping.
- Reward scaling of 1.0 for lower-level; 0.1 for higher-level.
- Lower-level exploration is Gaussian noise with $\sigma$ = 1.0.
- Higher-level exploration is Gaussian noise with $\sigma$ = 1.0.
- state $s_t$ : 30 dim.
- action $a_{t,z}$ : 8 dim.
- Training step : 10M steps
- Higher-level goal space for ant_fall_(multi or single)
  - meta_context_range = ((-4, -4, 0), (12, 28, 5))
  - goal dim. in higher-level policy : 3 dim.
- Higher-level goal space for ant_push_(multi or single)
  - meta_context_range = ((-16, -4), (16, 20))
  - goal dim. in higher-level policy : 2 dim.
- lower-level goal space :
  - CONTEXT_RANGE_MIN = (-10, -10, -0.5, -1, -1, -1, -1, -0.5, -0.3, -0.5, -0.3, -0.5, -0.3, -0.5, -0.3)
  - CONTEXT_RANGE_MAX = ( 10, 10, 0.5, 1, 1, 1, 1, 0.5, 0.3, 0.5, 0.3, 0.5, 0.3, 0.5, 0.3)
  - goal dim. in lower-level policy : 15 dim.

## APPENDIX B
## THE KEY POINTS FOR EXPERIMENTS

The key points to note during the experiment are reiterated below:

- Although the dimensions of $a_z$ and $a_{rnvp}$ are fixed based on the environment and the input of FDGM, the dimension of $a_{t,z-\text{state}}$ is determined through a trade-off process involving experimentation to identify the optimal dimension.
- The layer dimensions of all policies of 'Our model_8_8' in Fig. 3 are the same as those in 'Our model_8_15'.
- In Fig. 4, the goal space of the higher-level policy for 'Our model_2_8' and 'Our model_3_8' is constrained to match the original goal space of Ant Push Single and Ant Fall Single in HIRO, respectively.
- The layer dimensions of all policies in our model in Fig. 4 are the same as those in our model in Fig. 3.
- The optimal size of $a_{t,z-\text{state}}$ for each experiment in Fig. 4 is selected through experimentation.
- Similar to Ant Fall Single in Fig. 3, our model in Ant Fall Single in Fig. 4 leverages the combination of the inverse operation from Algorithm 2 and the neural network size from Algorithm 1.

- Although the LSP framework does not inherently include off-policy correction, LSP in Fig. 4 incorporates off-policy correction to ensure a fair comparison in the experiment.

## APPENDIX C
## THE COMPARISON TABLE

Our proposed model and all reference models are evaluated based on two key criteria: (1) off-policy correction capability and (2) FDGM integration. In Table II, 'O' indicates that a model satisfies a given criterion, while 'X' denotes its absence.

TABLE II: Model comparison

| | Off-policy correction | FDGM |
|---|---|---|
| **Our model** | O | O |
| **HIRO** | O | X |
| **LSP** | X | O |
| **Ma et al.** | X | O |

## REFERENCES

[1] T. Degris, M. White, and R. S. Sutton, "Off-policy actor-critic," *arXiv preprint arXiv:1205.4839*, 2012.
[2] O. Nachum, S. Gu, H. Lee, and S. Levine, "Data-efficient hierarchical reinforcement learning," *arXiv preprint arXiv:1805.08296*, 2018.