Hierarchical Reinforcement Learning with Universal Policies for Multi-Step Robotic Manipulation (Supplementary Material)

Xintong Yang[1], Ze Ji[1], Jing Wu[2], Yu-Kun Lai[2], Changyun Wei[3], Guoliang Liu[4], Rossitza Setchi[1]

I. Statistical analysis

In this section, we formally present the statistical analysis for the main experiments conducted in this work. For all of them, the *null hypotheses* assume that there is no statistically significant difference between the baseline methods and ours, i.e., the proposed methods show no improvements over the baseline. The *alternative hypotheses* assume that the proposed methods have statistically significant improvements over the baseline given the experiment data. We use the p-value method for statistical analysis, with a confidence level of $\alpha = 0.05$ for all tests. The experiment results are collected from three random seeds independently. Thus, we use the t-score to compute the p-values. Finally, we select different points at different epoch for statistical analysis according to the purposes of the corresponding experiments. For example, for the AAES experiment, we select the success rates of the 240-th epoch because AAES is design to speed up learning and the two curves in Fig. 7a seem to have the widest distance at this point. While for the comparison with HAC, we select the last point of the curves since their final performances are more meaningful.

We summarise the analyses results into Table I, which shows that, except for the use of binary high-level goals on task 1 (Fig. 6a), all the techniques proposed in this work can be considered statistically significant given a confidence level of 0.05.

Proposed Changes	Figure	Epoch	Baseline means	Baseline Std.	New means	New Std.	t-score	p-value
Block-gripper-informed goals	4a	90	0.127	0.100	0.933	0.032	43.644	0.0005
Binary high-level goals 1	5a	120	0.952	0.026	0.974	0.023	1.664	0.2380
Binary high-level goals 2	5b	280	0.519	0.096	0.852	0.037	15.608	0.0041
AAES	7q	240	0.446	0.088	0.796	0.063	9.595	0.0107
Parallel training	7	300	0.661	0.152	0.850	0.025	12.851	0.0060
Abstract demonstration (kinematic)	8a	90	0.303 (0.0-D)	0.032	0.830 (0.75-D)	0.010	91.221	0.0001
Abstract demonstration (planning)	8b	150	0.265 (0.0-D)	0.022	0.707 (0.75-D)	0.091	8.424	0.0138
Comparison with HAC 1	9c	300	0.193	0.046	0.989	0.011	124.130	0.0001
Comparison with HAC 2	9d	800	0.104	0.031	0.941	0.003	452.000	0.0001
Learning diverse combinatorial results 1	10a	150	0.937	0.055	0.981	0.013	6.000	0.0267
Learning diverse combinatorial results 2	10b	240	0.400	0.079	0.796	0.063	10.864	0.0084
Learning diverse combinatorial results 3	10c	700	0.442	0.016	0.841	0.033	20.965	0.0023
Learning diverse combinatorial results 4	10d	1000	0.519	0.040	0.571	0.002	47.000	0.0005

Table I: Statistical analysis results.

[1]Centre for Artificial Intelligence, Robotics and Human-Machine Systems (IROHMS), School of Engineering, Cardiff University, Cardiff, UK {yangx66, jiz1, setchi}@cardiff.ac.uk

[2]School of Computer Science and Informatics, Cardiff University, Cardiff, UK {wuj11, laiy4}@cardiff.ac.uk

[3]Robotics Engineering, Hohai University, Changzhou, China c.wei@hhu.edu.cn

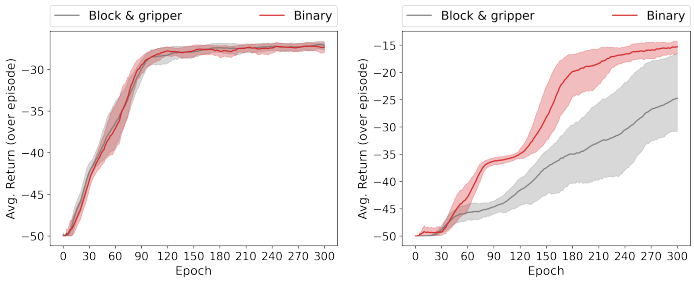
[4]School of Control Science and Engineering, Shandong University, Jinan, China liuguoliang@sdu.edu.cn

II. Mathematical Notation

Notation	Meaning	Section	Equation
$\mathcal{S}$	Set of states	III-A	-
$\mathcal{A}$	Set of actions	III-A	-
r	Reward (function)	III-A	-
$\gamma \in [0, 1]$	Discount factor	III-A	-
$p(\mathbf{s}_0)$	Initial state distribution	III-A	-
$p(\mathbf{s}_{t+1}\|\mathbf{s}_t, \mathbf{a}_t)$	The probability of transiting to $\mathbf{s}_{t+1}$ when taking action $\mathbf{a}_t$ at state $\mathbf{s}_t$	III-A	-
$\pi(\mathbf{a}\|\mathbf{s})$	Policy (the probability of taking action $\mathbf{a}$ at state $\mathbf{s}$)	III-A	-
$\mathcal{R}$	Return (accumulated rewards)	III-A	-
$\mathbb{E}[\mathcal{R}]$	Expectation of return	III-A	-
$Q^\pi(\mathbf{s}, \mathbf{a})$	Q value (the expected future return of taking action $\mathbf{a}$ at state $\mathbf{s}$)	III-A	-
$o\langle\mathcal{I}^o, \pi^o, \beta^o(\mathbf{s})\rangle$	Option	III-B	-
$\mathcal{O}$	Set of options	III-B	-
$\mathcal{I}^o \subseteq \mathcal{S}$	Set of initialisation states of an option	III-B	-
π^o	Intra-option policy of an option	III-B	-
$\beta^o(\mathbf{s}) \in [0, 1]$	Termination function of an option	III-B	-
$\Omega(o\|\mathbf{s})$	Inter-option policy (the probability of taking option o at state $\mathbf{s}$)	III-B	-
$Q^\Omega(\mathbf{s}, o)$	Option value (the expected future return of taking option o at state $\mathbf{s}$)	III-B	-
$\pi^\mathcal{L}$	Low-level policy (equivalent to the intra-option policy in this paper)	III-B	-
$\pi^\mathcal{H}$	High-level policy (equivalent to the inter-option policy in this paper)	III-B	-
$\mathcal{G}$	Set of goals	III-C	-
$f_g : \mathcal{S} \to \{0, 1\}$	A predicate that determine whether a goal is achieved at a state	III-C	-
$g^\mathcal{L}$	Low-level goal	IV-A	-
$\mathcal{G}^\mathcal{L}$	Set of low-level goals	IV-A	-
$o_g\langle\mathcal{I}_g, \pi_g^\mathcal{L}, \beta_g^\mathcal{L}\rangle$	Universal option (the three components are goal-conditioned)	IV-A	-
$\mathcal{I}_g$	Set of states from which a goal is achievable	IV-A	-
$\pi_g^\mathcal{L}(\mathbf{a}^\mathcal{L}\|\mathbf{s}, \mathbf{g}^\mathcal{L})$	Goal-conditioned low-level (intra-option) policy	IV-A	-
$\beta_g^\mathcal{L}(\mathbf{s})$	Goal-conditioned termination function	IV-A	-
$r_g^\mathcal{L}(\mathbf{s}, \mathbf{a}^\mathcal{L})$	Low-level goal-conditioned reward function	IV-A	-
$g^\mathcal{H}$	High-level goal	IV-A	-
$\mathcal{G}^\mathcal{H}$	Set of high-level goals	IV-A	-
$\pi_g^\mathcal{H}(\mathbf{a}^\mathcal{H}\|\mathbf{s}, \mathbf{g}^\mathcal{H})$	Universal high-level (inter-option) policy	IV-A	-
$r_g^\mathcal{H}(\mathbf{s}, \mathbf{a}^\mathcal{H})$	High-level goal-conditioned reward function	IV-A	-
N	The number of steps decomposed from a task	IV-B	-
$\psi^N : \mathcal{S}$	A mapping from states to N subsets of low-level goals	IV-B	-
$\mathcal{G}_n^\mathcal{L}$	Subset of low-level goals that correspond to the n-th step of a task	IV-B	-
$p^\mathcal{H}(\mathbf{s})$	The probability of the agent encountering state $\mathbf{s}'$ following the high-level policy	IV-C	-
$p^\mathcal{H}(\mathbf{s}'\|\mathbf{a}^\mathcal{H}, \mathbf{s})$	The probability of the agent entering state $\mathbf{s}'$ when taking a high-level action $\mathbf{a}^\mathcal{H}$ at state $\mathbf{s}$	IV-C	-
$\pi_g^\mathcal{L}(\mathbf{a}^\mathcal{L}\|\mathbf{a}^\mathcal{H}, \mathbf{s})$	Low-level policy	IV-C	-
$p^E(\mathbf{s}'\|\mathbf{a}^\mathcal{L}, \mathbf{s})$	The probability of the agent entering state $\mathbf{s}'$ when taking a low-level action $\mathbf{a}^\mathcal{L}$ at state $\mathbf{s}$	IV-C	-
$U(s, o)$	Option value upon arrival	V-A	Eq. 2, Eq. 3
ξ	A transition (experience)	V-A	Eq. 2, Eq. 3
$b_{\mathbf{sg}^\mathcal{L}}$	A binary variable indicating whether a low-level goal is achieved at state $\mathbf{s}$	V-A	Eq. 2, Eq. 3
D	Replay buffer	V-A	Eq. 2, Eq. 3
$\mathcal{U}(\cdot)$	Uniform distribution	V-A	Eq. 2, Eq. 3
θ, θ^-	Neural network parameters and a copy of them	V-A	Eq. 2, Eq. 3
τ	Ratio of the soft update of the target network parameters θ^-	V-A	-
$\mathcal{N}(\cdot)$	Normal distribution	V-B	-
$\boldsymbol{\alpha}_e$	An N-dimensional distribution mean vector of taking random actions, (e is the index of the last epoch)	V-B	Eq. 4
c_α	Upper bound constant of the probability mean of taking random actions	V-B	Eq. 4
$\mathbf{S}_e$	An N-dimensional probability vector of testing success rates	V-B	Eq. 4
$\boldsymbol{\sigma}_e$	An N-dimensional distribution standard deviation vector of taking random actions	V-B	Eq. 4
c_σ	Upper bound constant of the distribution standard deviation of taking random actions	V-B	Eq. 4
$\mathbf{S}_e^-$	A delayed copy of the success rate vector	V-B	Eq. 5
τ_s	Ratio of the soft update of the delayed copy of the success rate vector	V-B	Eq. 5

III. Supplementary figures

Figs. 1, 2, 3 and 4 are the average test return curves for comparing high-level goal representation, parallel training and separate high-level policies. They all look very similar to the success rate curves presented in Figs. 5, 7, 9 and 10 in the main content because we count a goal as successfully achieved when the return of the test episode for that goal is larger than $-1 \cdot episode\_length$. This is valid since we use a reward of 0 for a timestep where a goal is achieved and -1 otherwise. By comparing the return and success rate curves, one can see that the maximal return the agent can obtain is always limited by the actual timesteps needed to achieve a goal. In addition to indicating success, the larger return an agent can gather, the faster it can achieve a given goal. This means the agent is not only learning to achieve given goals, but also to achieve them as fast as possible.



(a) Task 1 (b) Task 2

Figure 1: Average test returns with block-gripper-informed and binary high-level goals.

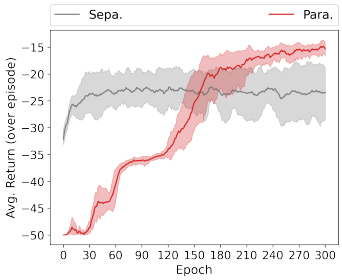
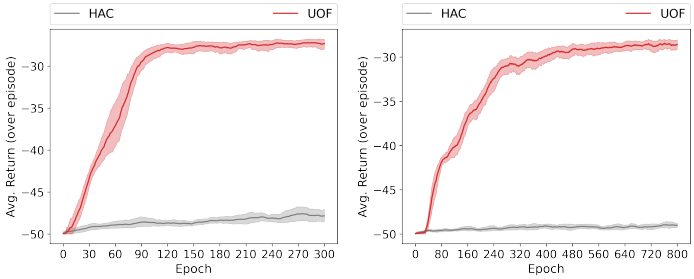
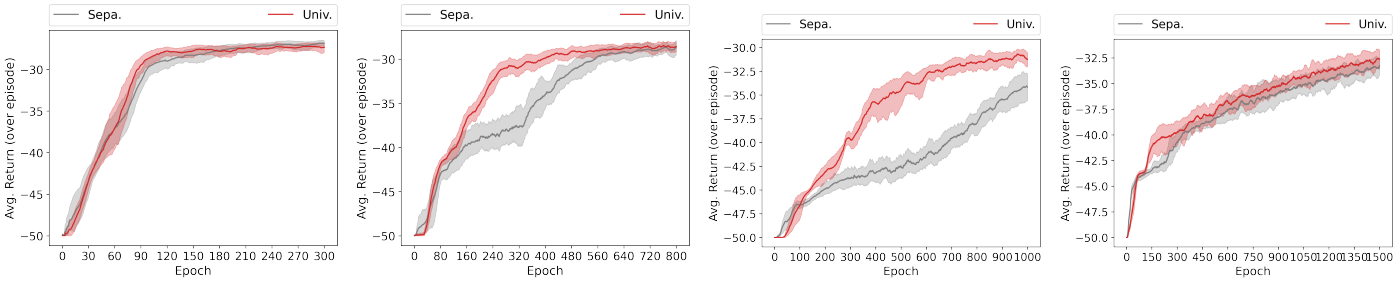


Figure 2: Average returns of high-level planning. **Sepa.**: Trained with a pre-trained low-level policy; **Para.**: Parallel training with low-level policy from scratch.



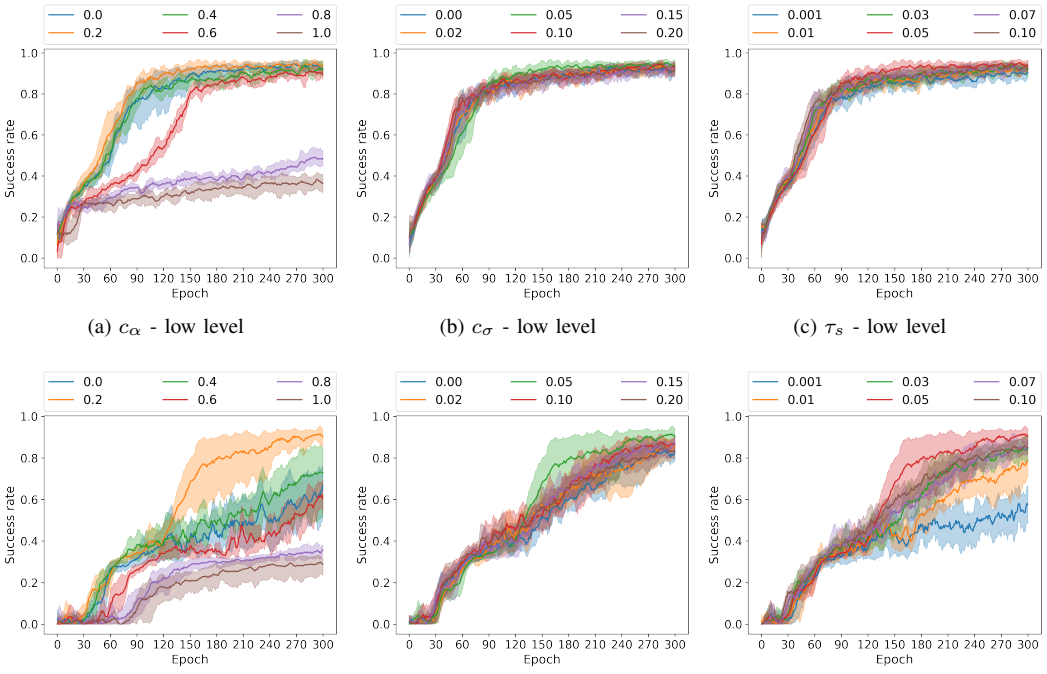
(a) Task 1 (b) Task 2

Figure 3: Average test returns of the high-level policy of HAC and UOF.



(a) Task 1 (b) Task 2 (c) Task 3 (d) Task 4

Figure 4: Average test returns of planning over multiple steps with universal and separated policies for the four tasks. **Univ**: single universal policy; **Sepa**: separated policies.



(a) c_α - low level (b) c_σ - low level (c) τ_s - low level

(d) c_α - high level (e) c_σ - high level (f) τ_s - high level

Figure 5: Average test success rates of both levels with different values of the AAES parameters: c_α, c_σ, and τ_s. The AAES for the low-level policy uses constant upper-bounds $c_\alpha = 0.2$ and $c_\sigma = 0.05$. We selected these values based on experiment results with $\tau_s = 0.05$. Figs. 5a and 5d are the results with different values of c_α, with $c_\sigma = 0.05$; while Figs.5b and 5e are the results with different values of c_σ, with $c_\alpha = 0.2$. They show that $c_\alpha = 0.2$ and $c_\sigma = 0.05$ are the best choices. For baselines without using AAES, α and σ are fixed at the upper bounds throughout training. The copy of the performance $S^-_{n,e}$ is updated with $\tau_s = 0.05$, the best choice given results shown in Figs. 5c and 5f (red lines).