Appendix
Options of Interest: Temporal Abstraction with Interest Functions

Khimya Khetarpal,[1,2] **Martin Klissarov,**[1,2] **Maxime Chevalier-Boisvert,**[2,3]
Pierre-Luc Bacon,[4] **Doina Precup**[1,2,5]
[1]McGill University, [2]Mila, [3]Universite de Montreal,
[4]Stanford University, [5]Google DeepMind
khimya.khetarpal@mail.mcgill.ca, martin.klissarov@mail.mcgill.ca
maxime.chevalier-boisvert@mila.quebec, plbacon@cs.stanford.edu, dprecup@cs.mcgill.ca

A Appendix

A.1 Reproducibility Checklist

We follow the reproducibility checklist by Pineau (2019) and point to relevant sections explaining them here.

For all algorithms presented, check if you include:

- **A clear description of the algorithm.** See main paper Algorithm 1. Also see included codebase provided as a zip file.
- **An analysis of the complexity (time, space, sample size) of the algorithm.** We do not perform a complexity analysis of the algorithm.
- **A link to a downloadable source code, including all dependencies.** See experimental section in Appendix and main paper, the code is included with supplemental as zip file also.

For any theoretical claim, check if you include:

- **A statement of the result.** See main paper and Proofs section in the appendix A.2.
- **A clear explanation of any assumptions.** See Proofs section in the appendix A.2.
- **A complete proof of the claim.** See Proofs section in the appendix A.2.

For all figures and tables that present empirical results, check if you include:

- **A complete description of the data collection process, including sample size.** We use custom variants of the tasks in standard benchmarks such as Mujoco (Todorov et al., 2012) and MiniWorld (Chevalier-Boisvert, 2018). We provide the code for the same in the supplementary material.
- **A link to a downloadable version of the dataset or simulation environment.** See (Chevalier-Boisvert, 2018) and (Todorov et al., 2012) for standard versions of these benchmarks. For fourrooms domain and custom changes to the tasks in Mujoco and MiniWorld see the the code is included with supplementary material.
- **An explanation of any data that were excluded, description of any pre-processing step.** We did not exclude any data in the environments we used. We do not require any data pre-processing step for our experiments.

- **An explanation of how samples were allocated for training / validation / testing.** We do not use a split as we are examining the optimization performance. Therefore, we report the performance during the learning process as shown in the figures. Once the model has been trained, we load the stored weights to demonstrate the performance via the videos and images as well.
- **The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results.** Besides maintaining consistency the default values of the baselines, we optimize both the baseline and our algorithm for other parameters. See Experimental Details for complete implementation and hyper-parameter details in Appendix Sec A.3.
- **The exact number of evaluation runs.** We used 10 independent seeds for TMaze experiments, 10 independent runs for MiniWorld first set of experiments, and 5 independent seeds for remaining experiments of HalfCheetah and MiniWorld. For tabular, we averaged performance across 70 independent runs. We did not run for more number of seeds due to time and computational constraints.
- **A description of how experiments were run.** See Experimental Results section in the main paper and for additional details see Appendix Sec A.3.
- **A clear definition of the specific measure or statistics used to report results.** We average discounted returns across all seeds as reported in the performance curves.
- **Clearly defined error bars.** We report the standard error in all cases.
- **A description of results with central tendency (e.g. mean) & variation (e.g. stddev).** We report the standard error in all cases. All figures with the returns show the standard error across the independent random seeds.
- **A description of the computing infrastructure used.** For tabular experiments, we use 1 CPU. For function approximation, we distribute all runs across 1 CPU and 1 GPU per run. Our CNN code takes longer to run (10 hours/run) as compared to our TMaze, HalfCheetah code (2 hours/run).

A.2 Proofs

A.2.1 Proof of Interest Function Gradient Theorem

Theorem. *Given a set of Markov options with differentiable interest functions $I_{\omega,z}$, where z is the parameter vector, the gradient of the expected discounted return with respect to z at (s,ω) is:*

$$\sum_{s',\omega'} \hat{\mu}_\Omega(s',\omega'|s,\omega)\beta_\omega(s')\frac{\partial \pi_{I_{\omega,z}}(\omega'|s')}{\partial z}Q_\Omega(s',\omega')$$

where $\hat{\mu}_\Omega(s',\omega'|s,\omega)$ is the discounted weighting of the state-option pairs along trajectories starting from (s,ω) sampled from the distribution determined by $\pi_{I_{\omega,z}}$, β_ω is the termination function and Q_Ω is the value function over options corresponding to $\pi_{I_{\omega,z}}$.

Proof. Let us start with the option-value function; $Q_{\Omega,\theta}(s,\omega)$

$$Q_\Omega(s,\omega) = \sum_a \pi_{\omega,\theta}(a|s)Q_U(s,\omega,a) \tag{1}$$

which depends on the interest function I which is parameterized by z. Therefore taking the derivation of Q_Ω w.r.t z.

$$\frac{\partial Q_\Omega(s,\omega)}{\partial z} = \frac{\partial}{\partial z}\Big\{\sum_a \pi_{\omega,\theta}(a|s)Q_U(s,\omega,a)\Big\} \tag{2}$$

Expanding Q_U; we get

$$\frac{\partial Q_\Omega(s,\omega)}{\partial z} = \frac{\partial}{\partial z}\Big\{\sum_a \pi_{\omega,\theta}(a|s)\Big(r(s,a)+\gamma\sum_{s'}P(s'|s,a)U(\omega,s')\Big)\Big\}$$

$$= \frac{\partial}{\partial z}\Big\{\sum_a \pi_{\omega,\theta}(a|s)\Big(r(s,a) + \gamma\sum_{s'} P(s'|s,a)\Big\{(1-\beta_{\omega,\nu}(s'))Q_\Omega(s',\omega) + \beta_{\omega,\nu}(s')V_\Omega(s')\Big\}\Big)\Big\}$$

$$= \sum_a \pi_{\omega,\theta}(a|s)\sum_{s'}\gamma P(s'|s,a)\Big\{(1-\beta_{\omega,\nu}(s'))\frac{\partial Q_\Omega(s',\omega)}{\partial z} + \beta_{\omega,\nu}(s')\frac{\partial V_\Omega(s')}{\partial z}\Big\} \tag{3}$$

Now the state-value function can be expressed in terms of option-value function as follows:

$$V_\Omega(s) = \sum_\omega \pi_{I_{\omega,z}}(\omega|s)Q_\Omega(s,\omega) \tag{4}$$

where

$$\pi_{I_{\omega,z}}(\omega|s) = I_{\omega,z}(s)\pi_\Omega(\omega|s)\Big/\sum_{\omega'} I_{\omega',z}(s)\pi_\Omega(\omega'|s)$$

Note that $\pi_\Omega(\omega|s)$ is fixed and not parameterized here.

Then, taking gradient w.r.t z yields:

$$\frac{\partial V_\Omega(s')}{\partial z} = \sum_\omega\Big(\frac{\partial \pi_{I_{\omega,z}}(\omega|s')}{\partial z}Q_\Omega(s',\omega) + \pi_{I_{\omega,z}}(\omega|s')\frac{\partial Q_\Omega(s',\omega)}{\partial z}\Big) \tag{5}$$

Substituting (5) in (3), we get:

$$\frac{\partial Q_\Omega(s,\omega)}{\partial z} = \sum_a \pi_{\omega,\theta}(a|s)\sum_{s'}\gamma P(s'|s,a)\Big\{(1-\beta_{\omega,\nu}(s'))\frac{\partial Q_\Omega(s',\omega)}{\partial z} + \beta_{\omega,\nu}(s')\Big\{\sum_{\omega'}\Big(\frac{\partial \pi_{I_{\omega,z}}(\omega'|s')}{\partial z}Q_\Omega(s',\omega') + \pi_{I_{\omega,z}}(\omega'|s')\frac{\partial Q_\Omega(s',\omega')}{\partial z}\Big)\Big\}\Big\}$$

Collecting coefficients together;

$$\frac{\partial Q_\Omega(s,\omega)}{\partial z} = \sum_a \pi_{\omega,\theta}(a|s)\sum_{s'}\gamma P(s'|s,a)\sum_{\omega'}\beta_{\omega,\nu}(s')\frac{\partial \pi_{I_{\omega,z}}(\omega'|s')}{\partial z}Q_\Omega(s',\omega') + \sum_a \pi_{\omega,\theta}(a|s)\sum_{s'}\gamma P(s'|s,a)\sum_{\omega'}\Big((1-\beta_{\omega,\nu}(s')) + \beta_{\omega,\nu}(s')\pi_{I_{\omega,z}}(\omega'|s')\Big)\frac{\partial Q_\Omega(s',\omega')}{\partial z}$$

Rearranging summations for term 2, we get:

$$\frac{\partial Q_\Omega(s,\omega)}{\partial z} = \sum_a \pi_{\omega,\theta}(a|s)\sum_{s'}\gamma P(s'|s,a)\sum_{\omega'}\beta_{\omega,\nu}(s')\frac{\partial \pi_{I_{\omega,z}}(\omega'|s')}{\partial z}Q_\Omega(s',\omega') + \sum_{s'}\sum_{\omega'}\Big(\sum_a \pi_{\omega,\theta}(a|s)\gamma P(s'|s,a)\Big((1-\beta_{\omega,\nu}(s')) + \beta_{\omega,\nu}(s')\pi_{I_{\omega,z}}(\omega'|s')\Big)\Big)\frac{\partial Q_\Omega(s',\omega')}{\partial z}$$

In the above equation, one-step discounted transition probability in the augmented space is given as

$$P_\gamma^{(1)}(s',\omega'|s,\omega) = \sum_a \pi_{\omega,\theta}(a|s)\gamma P(s'|s,a)\Big((1-\beta_{\omega,\nu}(s'))\mathbb{1}_{\omega=\omega'} + \beta_{\omega,\nu}(s')\pi_{I_{\omega,z}}(\omega'|s')\Big)$$

Thus we rewrite $\frac{\partial Q_\Omega(s,\omega)}{\partial z}$ as:

$$\frac{\partial Q_\Omega(s,\omega)}{\partial z} = \sum_a \pi_{\omega,\theta}(a|s)\sum_{s'}\gamma P(s'|s,a)\sum_{\omega'}\beta_{\omega,\nu}(s')\frac{\partial \pi_{I_{\omega,z}}(\omega'|s')}{\partial z}Q_\Omega(s',\omega') + \sum_{s'}\sum_{\omega'}P_\gamma^{(1)}(s',\omega'|s,\omega)\frac{\partial Q_\Omega(s',\omega')}{\partial z}$$

Since we have recursion between current s, ω with consecutive state-option pairs, using the k-steps augmented process as shown in Section A.2.4, we obtain the following:

$$\frac{\partial Q_\Omega(s,\omega)}{\partial z} = \sum_{k=0}^{\infty} \sum_{s',\omega'} P_\gamma^{(k)}(s',\omega'|s,\omega) \sum_a \pi_{\omega,\theta}(a|s') \beta_{\omega,\nu}(s') \frac{\partial \pi_{I_{\omega,z}}(\omega'|s')}{\partial z} Q_\Omega(s',\omega') \tag{6}$$

Therefore, the interest-function gradient is given as follows:

$$\boxed{= \sum_{s',\omega'} \hat{\mu_\Omega}(s',\omega'|s,\omega) \beta_{\omega,\nu}(s') \frac{\partial \pi_{I_{\omega,z}}(\omega'|s')}{\partial z} Q_{\Omega,\theta}(s',\omega')} \tag{7}$$

where $\hat{\mu}_\Omega(s',\omega'|s,\omega)$ is the discounted weighting of the state-option pairs along trajectories starting from (s,ω) sampled from the distribution determined by $\pi_{I_{\omega,z}}$, β_ω is the termination function and Q_Ω is the value function over options corresponding to $\pi_{I_{\omega,z}}$. Note that this differs from the discounted weighting of state-option pairs in the option-critic derivation. For the interest function gradient update, the derivation of $\pi_{I_{\omega,z}}(\omega|s)$ with respect to the parameter z is shown in Section A.2.5 in this appendix. □

A.2.2 Proof of Intra-Option Policy Gradient Theorem

Theorem. *Given a set of Markov options with stochastic, differentiable intra-option policies $\pi_{\omega,\theta}$, the gradient of the expected discounted return with respect to θ and initial condition (s_0, ω_0) is:*

$$\sum_{s,\omega} \hat{\mu}_\Omega(s,\omega|s_0,\omega_0) \sum_a \frac{\partial \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s,\omega,a)$$

where $\hat{\mu}_\Omega(s,\omega|s_0,\omega_0)$ is the discounted weighting of the state-option pairs along trajectories starting from (s_0,ω_0) sampled from the new option sampling distribution determined by $I_{\omega,z}(s)$.

Proof. Starting with the option-value function $Q_\Omega(s,\omega)$, we take the gradient of this with respect to θ as follows:

$$\frac{\partial Q_\Omega(s,\omega)}{\partial \theta} = \frac{\partial}{\partial \theta}\left\{ \sum_a \pi_{\omega,\theta}(a|s) Q_U(s,\omega,a) \right\}$$

$$= \sum_a \left(\frac{\partial \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s,\omega,a) + \pi_{\omega,\theta}(a|s) \frac{\partial Q_U(s,\omega,a)}{\partial \theta} \right)$$

$$= \sum_a \left(\frac{\partial \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s,\omega,a) + \pi_{\omega,\theta}(a|s) \sum_{s'} \gamma P(s'|s,a) \frac{\partial U(\omega,s')}{\partial \theta} \right) \tag{8}$$

Now $U(\omega,s') = (1-\beta_{\omega,\nu}(s'))Q_\Omega(s',\omega) + \beta_{\omega,\nu}(s')V_\Omega(s')$. Substituting $V_\Omega(s')$ and then taking the gradient of U with respect to θ, we get:

$$\frac{\partial U(\omega,s')}{\partial \theta} = (1-\beta_{\omega,\nu}(s')) \frac{\partial Q_\Omega(s',\omega)}{\partial \theta} + \beta_{\omega,\nu}(s') \left(\sum_{\omega'} I_{\omega,z}(s') \pi_\Omega(\omega'|s') \Big/ \sum_{\omega} I_{\omega,z}(s') \pi_\Omega(\omega|s') \right) \frac{\partial Q_\Omega(s',\omega')}{\partial \theta}$$

$$= \sum_{\omega'} \left[\left((1-\beta_{\omega,\nu}(s')) \mathbb{1}_{\omega'=\omega} + \beta_{\omega,\nu}(s') \left(I_{\omega,z}(s') \pi_\Omega(\omega'|s') \Big/ \sum_{\omega} I_{\omega,z}(s') \pi_\Omega(\omega|s') \right) \right) \frac{\partial Q_\Omega(s',\omega')}{\partial \theta} \right] \tag{9}$$

Substituting (9) in (8), we get:

$$\frac{\partial Q_\Omega(s,\omega)}{\partial \theta} = \sum_a \frac{\partial \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s,\omega,a) + \sum_a \pi_{\omega,\theta}(a|s) \sum_{s'} \gamma P(s'|s,a) \times$$

$$\sum_{\omega'} \left((1-\beta_{\omega,\nu}(s')) \mathbb{1}_{\omega'=\omega} + \beta_{\omega,\nu}(s') \left(I_{\omega,z}(s') \pi_\Omega(\omega'|s') \Big/ \sum_{\omega'} I_{\omega',z}(s') \pi_\Omega(\omega'|s') \right) \right) \frac{\partial Q_\Omega(s',\omega')}{\partial \theta}$$

Substituting one-step discounted probability from the augmented process as shown in Section A.2.4, we get:

$$= \sum_a \frac{\partial \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s,\omega,a) + \sum_{s'} \sum_{\omega'} P_\gamma^{(1)}(s',\omega'|s,\omega) \frac{\partial Q_\Omega(s',\omega')}{\partial \theta}$$

By extension to k time steps, we get:

$$= \sum_{k=0}^{\infty} \sum_{s',\omega'} P_\gamma^{(k)}(s',\omega'|s,\omega) \sum_a \frac{\partial \pi_{\omega,\theta}(a|s')}{\partial \theta} Q_U(s,\omega,a)$$

$$\boxed{= \sum_{s,\omega} \hat{\mu}_\Omega(s,\omega|s_0,\omega_0) \sum_a \frac{\partial \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s,\omega,a)}$$

where $\hat{\mu}_\Omega(s,\omega|s_0,\omega_0) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s, \omega_t = \omega|s_0,\omega_0)$. Note that this differs from the discounted weighting of state-option pairs in the option-critic derivation. □

A.2.3 Proof of Termination-Gradient Theorem

Theorem. *Given a set of Markov options with stochastic, differentiable termination functions $\beta_{\omega,\nu}$, the gradient of the expected discounted return with respect to ν and initial condition (s_1,ω_0) is:*

$$-\sum_{s',\omega} \hat{\mu}_\Omega(s',\omega|s_1,\omega_0) \sum_a \frac{\partial \beta_{\omega,\nu}(s')}{\partial \nu} A_\Omega(s',\omega)$$

where $\hat{\mu}_\Omega(s,\omega|s_0,\omega_0)$ is the discounted weighting of the state-option pairs along trajectories starting from (s_0,ω_0) sampled from the new option sampling distribution determined by $I_{\omega,z}(s)$.

Proof. The proof is fairly simple and follows through similar to Bacon et al. (2017). The only and the key difference in the result is the new discounted weighting of state-option pairs as shown in Section A.2.4. $\hat{\mu_\Omega}(s,\omega|s_0,\omega_0)$ now includes interest functions influencing how policy over options are chosen in any given state as opposed to all options being present everywhere.

This time, we work with the expected sum of discounted rewards starting from $(s_1,\omega_0$:

$$U(\omega_0,s_1) = E\Big[\sum_{t=1}^{\infty} \gamma^{t-1} r_t | s_1,\omega_0\Big]$$

U can be expanded as follows:

$$U(\omega,s') = (1-\beta_{\omega,\nu}(s'))Q_\Omega(s',\omega) + \beta_{\omega,\nu}(s')V_\Omega(s')$$

With interest functions, this is further expanded:

$$U(\omega,s') = (1-\beta_{\omega,\nu}(s'))Q_\Omega(s',\omega) + \beta_{\omega,\nu}(s') \sum_\omega \Big\{ I_{\omega,z}(s')\pi_{\Omega,\theta}(\omega|s') \Big/ \sum_\omega I_{\omega,z}(s')\pi_\Omega(\omega|s')Q_\Omega(s',\omega') \Big\}$$

Expanding U now;

$$\begin{aligned} &= (1-\beta_{\omega,\nu}(s')) \sum_a \pi_{\omega,\theta}(a|s) \Big(r(s',a) + \sum_{s''} \gamma P(s''|s',a) U(\omega,s'') \Big) + \\ &\beta_{\omega,\nu}(s') \sum_{\omega'} \Big\{ I_{\omega',z}(s')\pi_\Omega(\omega'|s') \Big/ \sum_{\omega'} I_{\omega',z}(s')\pi_\Omega(\omega'|s') \Big\} \times \\ &\sum_a \pi_{\omega',\theta}(a|s') \Big(r(s',a) + \sum_{s''} \gamma P(s''|s',a) U(\omega',s'') \Big) \end{aligned}$$

Gradient of U then becomes:

$$\frac{\partial U(\omega,s')}{\partial \nu} = \frac{\partial \beta_{\omega,\nu}(s')}{\partial \nu}(V_\Omega(s') - Q_\Omega(s',\omega)) + (1-\beta_{\omega,\nu}(s')) \sum_a \pi_{\omega,\theta}(a|s') \sum_{s''} \gamma P(s''|s',a) U(\omega,s'')$$

Substituting the advantage function $A_\Omega(s',\omega) = Q_\Omega(s',\omega) - V_\Omega(s')$ and using the augmented process, we get:

$$\frac{\partial U(\omega, s')}{\partial \nu} = -\frac{\partial \beta_{\omega,\nu}(s')}{\partial \nu} A_\Omega(s',\omega) + \sum_{\omega'}\sum_{s''} P_\gamma^{(1)}(s'',\omega'|s',\omega)\frac{\partial U(\omega', s'')}{\partial \nu}$$

Using the recursive form and extending to k-time steps, we get:

$$= -\sum_{\omega',s''}\sum_{k=0}^{\infty} P_\gamma^{(k)}(s'',\omega'|s',\omega)\frac{\partial \beta_{\omega',\nu}(s'')}{\partial \nu} A_\Omega(s'',\omega')$$

This gives us the following result:

$$\boxed{\frac{\partial U(\omega_0, s_1)}{\partial \nu} = -\sum_{s',\omega} \hat{\mu}_\Omega(s',\omega|s_1,\omega_0)\frac{\partial \beta_{\omega,\nu}(s')}{\partial \nu} A_\Omega(s',\omega)}$$

where $\hat{\mu}_\Omega(s,\omega|s_0,\omega_0) = \sum_{t=0}^{\infty}\gamma^t P(s_t = s, \omega_t = \omega|s_0,\omega_0)$. Note that this differs from the discounted weighting of state-option pairs in the option-critic derivation □

A.2.4 Augmented Process with Interest Functions

Following the augmented process shown in Bacon et al. (2017), we present how the new formulation of option-critic with interest functions impacts this process. Let us consider that an option ω_t has been initiated or is executing at time t, then the discounted probability of transitioning to (s_{t+1},ω_{t+1}) is now given as:

$$P_\gamma^{(1)}(s_{t+1},\omega_{t+1}|s_t,\omega_t) = \sum_a \pi_{\omega_t,\theta}(a|s_t)\gamma P(s_{t+1}|s_t,a)\Bigg((1-\beta_{\omega_t,\nu}(s_{t+1}))\mathbb{1}_{\omega_t=\omega_{t+1}} +$$
$$\beta_{\omega_t,\nu}(s_{t+1})\Bigg(I_{\omega_{t+1},z}(s_{t+1})\pi_\Omega(\omega_{t+1}|s_{t+1})\Big/\sum_\omega I_{\omega,z}(s_{t+1})\pi_\Omega(\omega|s_{t+1})\Bigg)\Bigg)$$

When transitioning from $(s_t,\omega_{t-1}) \longrightarrow (s_{t+1},\omega_t)$, the discounted probability is:

$$P_\gamma^{(1)}(s_{t+1},\omega_t|s_t,\omega_{t-1}) = (1-\beta_{\omega_{t-1},\nu}(s_t))\mathbb{1}_{\omega_t=\omega_{t-1}} + \beta_{\omega_{t-1},\nu}(s_t)\times$$
$$\Bigg(I_{\omega_t,z}(s_{t+1})\pi_\Omega(\omega_t|s_{t+1})\Big/\sum_\omega I_{\omega,z}(s_{t+1})\pi_\Omega(\omega_t|s_{t+1})\Bigg)\sum_a \pi_{\omega_t,\theta}(a|s_t)\gamma P(s_{t+1}|s_t,a)$$

More generally, the k-steps discounted probability can be expressed recursively as follows:

$$P_\gamma^{(k)}(s_{t+k},\omega_{t+k}|s_t,\omega_t) = \sum_{s_{t+1}}\sum_{\omega_{t+1}}\Bigg(P_\gamma^{(1)}(s_{t+1},\omega_{t+1}|s_t,\omega_t)P_\gamma^{(k-1)}(s_{t+k},\omega_{t+k}|s_{t+1},\omega_{t+1})\Bigg)$$

$$P_\gamma^{(k)}(s_{t+k},\omega_{t+k-1}|s_t,\omega_{t-1}) = \sum_{s_{t+1}}\sum_{\omega_t}\Bigg(P_\gamma^{(1)}(s_{t+1},\omega_t|s_t,\omega_{t-1})P_\gamma^{(k-1)}(s_{t+k},\omega_{t+k-1}|s_{t+1},\omega_t)\Bigg)$$

The augmented process with the introduction of Interest Functions turns out to be same as in (Bacon et al., 2017) with the only and main difference in how policy over options are selected in any given state is now determined as following: $\pi_{I_{\omega,z}}(\omega|s) = I_{\omega,z}(s)\pi_\Omega(\omega|s)\Big/\sum_\omega I_{\omega,z}(s)\pi_\Omega(\omega|s)$.

A.2.5 Derivation of $\pi_{I_{\omega,z}}(\omega|s)$

In this section, we show the derivation of the interest-functions induced probability distribution for sampling options in any given state:

$$\pi_{I_{\omega,z}}(\omega|s) = I_{\omega,z}(s)\pi_\Omega(\omega|s)\Big/\sum_{\omega'} I_{\omega,z}(s)\pi_\Omega(\omega|s)$$

Using the log-trick and taking gradient with respect to z, we could rewrite it as:

$$\nabla_z \pi_{I_{\omega,z}}(\omega|s) = \pi_{I_{\omega,z}}(\omega|s) \nabla_z \log \pi_{I_{\omega,z}}(\omega|s)$$

Substituting the value of $\pi_{I_{\omega,z}}(\omega|s)$

$$= \pi_{I_{\omega,z}}(\omega|s) \nabla_z \log \left(I_{\omega,z}(s) \pi_\Omega(\omega|s) \Big/ \sum_\omega I_{\omega,z}(s) \pi_\Omega(\omega|s) \right)$$

Using $\log(a/b) = \log a - \log b$

$$= \pi_{I_{\omega,z}}(\omega|s) \nabla_z \left(\log I_{\omega,z}(s) \pi_\Omega(\omega|s) - \log \sum_\omega I_{\omega,z}(s) \pi_{\Omega,\theta}(\omega|s) \right)$$

Using $\log(ab) = \log a + \log b$

$$= \pi_{I_{\omega,z}}(\omega|s) \left(\nabla_z \log I_{\omega,z}(s) + \nabla_z \log \pi_\Omega(\omega|s) - \nabla_z \log(\sum_\omega I_{\omega,z}(s) \pi_\Omega(\omega|s)) \right)$$

$$= \pi_{I_{\omega,z}}(\omega|s) \nabla_z \log I_{\omega,z}(s) + \pi_{I_{\omega,z}}(\omega|s) \nabla_z \log \pi_\Omega(\omega|s) - \pi_{I_{\omega,z}}(\omega|s) \nabla_z \log(\sum_\omega I_{\omega,z}(s) \pi_\Omega(\omega|s))$$

The second term is equal to 0, so we obtain the following result:

$$\boxed{= \pi_{I_{\omega,z}}(\omega|s) \frac{1}{I_{\omega,z}(s)} \nabla_z I_{\omega,z}(s) - \pi_{I_{\omega,z}}(\omega|s) \frac{1}{\sum_\omega I_{\omega,z}(s) \pi_\Omega(\omega|s)} \sum_\omega \nabla_z I_{\omega,z}(s) \pi_\Omega(\omega|s)}$$

A.3 Experimental Details

A.3.1 Four Rooms Domain

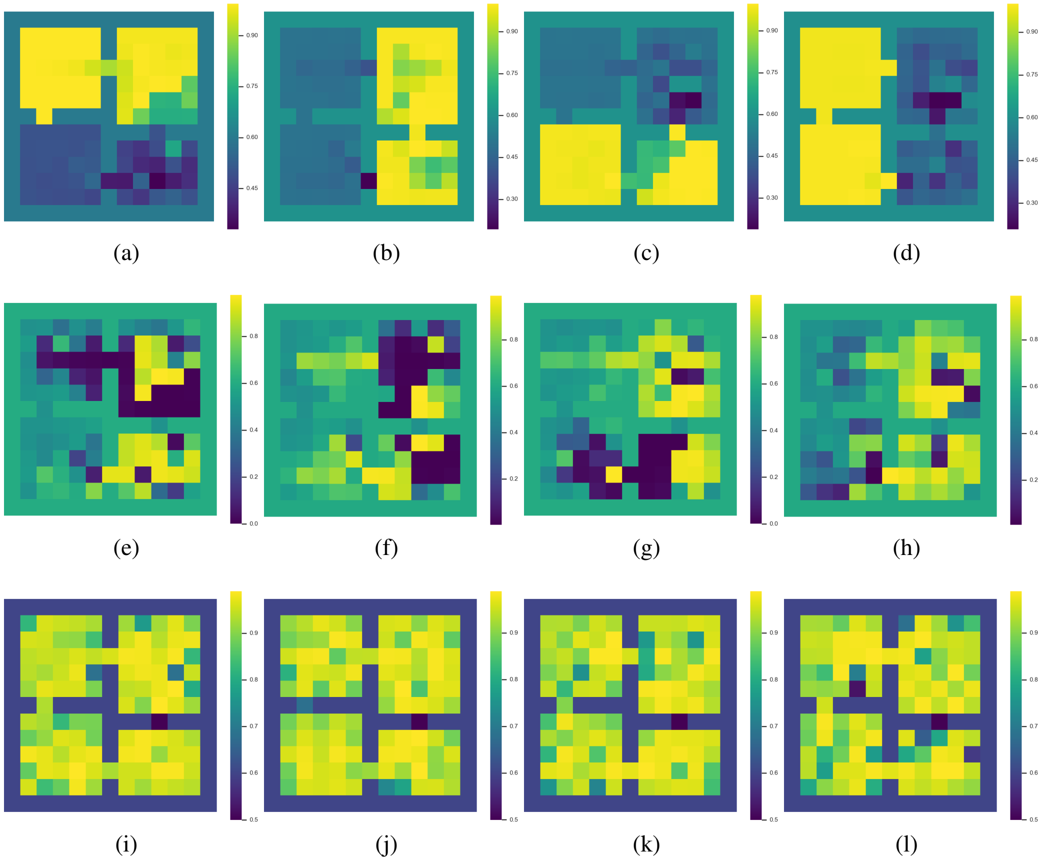
Implementation details: The discount factor is 0.99, and the reward is $+50$ at the goal and 0 otherwise. We used 4 options, whose intra-option policies were parameterized with Boltzmann distributions, and termination and interest functions represented as linear-sigmoid functions. Options were learned using either Interest-Option-Critic (IOC) or Option-Critic (OC) with tabular intra-option Q-learning, as described in Algorithm 1.

Hyper-parameter details: Based on the values reported in Bacon (2018), we used a baseline for the gradient estimator and a temperature of 0.01 for the intra-option policies, a learning rate of 0.5 for the critic and 0.25 for the termination and intra-option updates for both OC and IOC. For the IOC agent, a learning rate of 0.15 was used for the interest function updates. This was picked to be lower than the learning rate of β chosen based on a small search. Learning proceeds for a total of 500 episodes, with a maximum of 2000 time steps allowed per episode. Interest functions weights were initialized with a room specific structure prior. All other weights are initialized to zeros. We ran 70 independent runs and average returns and steps across these runs. The code is provided with the supplementary material in a zip file.

Here we show a qualitative analysis of the options learned by the IOC and the OC agent (Fig. A1).

A.3.2 TMaze

Implementation details: For function approximation, we built on top of the Proximal Policy Option-Critic (PPOC) algorithm (Klissarov et al., 2017) incorporating learning interest functions. The update rules are consistent with Algorithm 1. To represent the interest functions, we add another network to the PPOC algorithm: a 2-layer feed forward network with tanh activation. The input of the network is the state while the output is a sigmoid with size equal to the number of options, representing the interest for each option at the state. For the termination function, intra-option policies and state-option value functions, we keep the architecture consistent to Klissarov et al. (2017). We train for 150



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure A1: **Visualization of Interest Functions (a-d), IOC (e-h) and OC (i-l) Termination Conditions** at the end of 500 episodes in task 1 with 4 options. Brighter colors represent higher values. Option learned with interest functions emerge with specific interest in different regions of the state space as shown in Figures (a) to (d). Each row represents Option 1 to 4 going from left to right. IOC Termination conditions for each option emerges complimenting the interest of that option as opposed to the termination conditions (i-l) for each option in OC which assumes that all options are available everywhere and therefore result in options terminating everywhere.

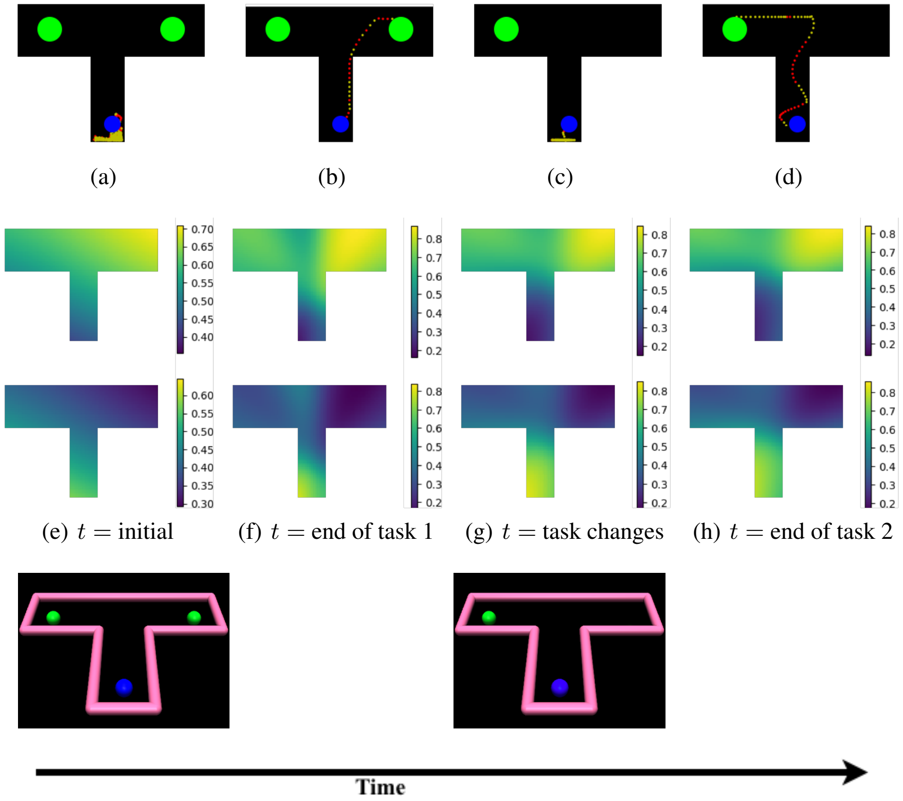
iterations and average performance over 10 independent runs of the algorithm. The code is also provided with the supplementary material in a zip file.

Hyper-parameter details - TMaze Domain: For each algorithm, we report results for the best hyperparameters configuration, after performing a sweep on the intra-option policies, value function, termination function and interest function learning rates (Tab. 1). For the transfer experiments: In addition to the learning rates of interest function, intra-option policies, a sweep across learning rate of policy over option was also performed and the optimum parameters reported for both OC and IOC agents. All other parameters are kept consistent with the baseline.

Table 1: Hyper-parameter details - TMaze domain

π_ω **lr**	$1e-4$	$3e-4$	$5e-4$	$7e-4$	$9e-4$
$I_{\omega,z}$ **lr**	$1e-4$	$3e-4$	$5e-4$	$7e-4$	$9e-4$
π_Ω **lr**	$1e-4$	$3e-4$	$5e-4$	$7e-4$	$9e-4$
Seed	0	1	2	..	9

Analysis of options learned by the OC agent: We also investigated the options learned by the OC agent. Fig. A2 illustrates the timeline from the beginning of task 1 to the end of task 2. Since options in OC do not have interest functions, we visualize the learned policy over options to analyze the nature of options learned. Over time, the options learned by the OC agent have not achieved any



(a) (b) (c) (d)

(e) $t =$ initial (f) $t =$ end of task 1 (g) $t =$ task changes (h) $t =$ end of task 2

Figure A2: **Analysis of Options learned by the OC agent Row 1**: depicts sampled trajectories with Option 0 indicated by red dots and Option 0 by yellow dots. **Row 2 & 3**: show the policy over options which is initialized at random initially. Over time, the options learned by the OC agent have not achieved any specialization.

specialization. This is a direct consequence of the assumption in OC that *all options are available everywhere.* Upon inspecting the sampled trajectories at the end of task 1 & 2, it is observed that the options cannot be interpreted as skills which have specialization is different regions of the state space. On the contrary, the interest of each options in the IOC agent lends meaning and focus to each option as depicted in the Fig. 4 in the main paper.

A.3.3 MiniWorld

Implementation details: We use the *Oneroom* task where the agent has to navigate to a randomly placed red block in a closed room (Fig.3(f)). This requires for the agent to turn around and scan the room to find the red box. The observation space is a 3-channel RGB image of 60 × 80 dimension. The action space consists of discrete 8 actions. At the start of each episode the red box is placed randomly in the closed room. The episode terminates if the agent reaches the red box or a max time steps of 180 is reached.

In our experiments, a deep convolutional neural network is employed as function approximator which takes in the state image as input and outputs the hidden layer. The interest and termination function of each option for states is parameterized by sigmoid functions, the output is linear representing the value functions and intra-option policies with softmax policy over option for both IOC and OC which is also being learnt alongside options. The CNN architecture is kept consistent with DQN (Mnih et al., 2015).

Hyper-parameter details - Miniworld Domain: The baseline algorithm is PPOC Klissarov et al. (2017) adapted for discrete actions. Performance is reported over 10 independent runs (for the first experiment with a uniform fixed policy over option) after a complete sweep over the intra-option policies, value function, termination function and interest function learning rates (Tab. 2). For the transfer experiments: in addition to the learning rates of interest function, intra-option policies, a sweep across learning rate of policy over option was also performed and the optimum parameters reported for both OC and IOC agents. Performance is reported over 5 independent runs in this case. Complete range of hyper-parameters swept are mentioned in Tab. 2. All other parameters are kept consistent with the baseline.

Table 2: Hyper-parameter details - MiniWorld domain

π_ω **lr**	$3e-4$	$7e-4$	$1e-4$	$5e-4$	$2e-4$		
$I_{\omega,z}$ **lr**	$1e-04$	$3e-03$	$8e-04$	$8e-05$	$5e-04$	$3e-04$	$9e-05$
π_Ω **lr**	$3e-4$	$7e-4$	$1e-4$	$5e-4$	$2e-4$		
Seed	0	1	2	3	4	5	

A.3.4 HalfCheetah

Implementation details: The experiments for HalfCheetah use the exact same implementation details as the TMaze domain A.3.2. The environment has been customized to reflect our task specifications. The code is provided in the supplementary zip file.

Hyper-parameter details: Tab 3 enlists the complete hyper-parameter details.

Analysis of options learned by the OC agent: Analogous to the timeline of options learned by the IOC agent, we visualize the timeline of OC options over an episode (Fig. A3). We observe that the options are noisy and often switch to the other choices available. Options learned by the OC agent cannot be interpreted as distinct skills.

Figure A3: **Timeline of options used by OC agent in HalfCheetah** where each option is represented by a distinct color (black & white).

Table 3: Hyper-parameter details - HalfCheetah domain

π_ω **lr**	$3e-4$	$7e-4$	$1e-4$	$5e-4$	$9e-4$
$I_{\omega,z}$ **lr**	$1e-04$	$3e-04$	$5e-04$	$7e-04$	$9e-04$
π_Ω **lr**	$3e-4$	$7e-4$	$1e-4$	$5e-4$	
Seed	0	1	2	3	4

A.4 Interest as an Attention Mechanism

To further analyze the meaning of interest functions learned by the IOC agent; we overlaid the interest of the active option on the timeline of an episode in HalfCheetah (Fig. A4). Our findings of how interest of an option activates across trajectories indicates that interest function is not just limited to being a measure of where an option initiates. It could be interpreted as an attention mechanism for where an option should attend to. In doing so, it lends the option an indication of where to stop attending as well (analogous to termination). To some extent, the interest functions learnt are able to override the termination degeneracies (only one option being active all the time, or options switching often) even though our approach does not tackle that problem directly.

References

Bacon, P.-L. (2018). *Temporal Representation Learning*. PhD thesis, McGill University, Montreal.

Bacon, P.-L., Harb, J., and Precup, D. (2017). The option-critic architecture. In *AAAI*, pages 1726–1734.

Chevalier-Boisvert, M. (2018). gym-miniworld environment for openai gym. `https://github.com/maximecb/gym-miniworld`.

Klissarov, M., Bacon, P., Harb, J., and Precup, D. (2017). Learnings options end-to-end for continuous action tasks. *CoRR*, abs/1712.00004.

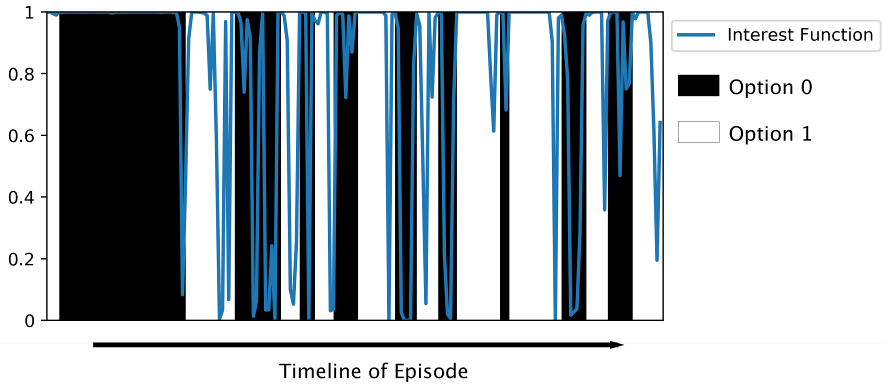


Figure A4: **Interest Function** overlaid on the timeline of an episode of HalfCheetah. Option 0 and 1 are depicted by black and white color respectively. The blue line shows the interest of the active option, which peaks everytime an option is active.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

Pineau, J. (2019). The machine learning reproducibility checklist. 1.2(1):1.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5026–5033. IEEE.