A. Experimental Details

In this section we describe the environments used in our experiments (see Section 4) and the experiment design.

A.1. Highway

We build on the highway-v0 task from the highway-env traffic simulator (Leurent, 2018). The task is specified by:

1. **State space, $S$:** The kinematic information of the ego vehicle and the five closest vehicles (ordered from closest to the furthest) is used as the Markov state, i.e., $s_t = \{[x_t, y_t, \dot{x}_t, \dot{y}_t]\}_\text{ego}, \text{other}_1, \ldots, \text{other}_5 \in \mathbb{R}^{6 \times 4}$. The ego-car is illustrated in green and the other cars in blue.

2. **Action space, $A$:** We use a discrete action space, constructed by $K$-means clustering of the continuous actions of the intelligent driving model (Kesting et al., 2010). We found out that keeping 9 actions was sufficient, i.e., $a_t \in \{0, \ldots, 8\}$.

3. **Demonstrations, $D$:** At each time-step, the ego-car observes online the state-action pairs for the 5 closest cars.

A.2. Roundabout

We build on the roundabot-v0 task from the highway-env traffic simulator (Leurent, 2018). The task is specified by:

1. **State space, $S$:** The kinematic information of the ego vehicle and the five closest vehicles (ordered from closest to the furthest) is used as the Markov state, i.e., $s_t = \{[x_t, y_t, \dot{x}_t, \dot{y}_t]\}_\text{ego}, \text{other}_1, \ldots, \text{other}_3 \in \mathbb{R}^{4 \times 4}$. The ego-car is illustrated in green and the other cars in blue.

2. **Action space, $A$:** We use a discrete action space, constructed by $K$-means clustering of the continuous actions of the intelligent driving model (Kesting et al., 2010). We found out that keeping 6 actions was sufficient, i.e., $a_t \in \{0, \ldots, 5\}$.

3. **Demonstrations, $D$:** At each time-step, the ego-car observes online the state-action pairs for the 3 closest cars.

A.3. CoinGrid

We build a simple multi-task grid-world. The task is specified by:

1. **State space, $S$:** We use a symbolic, multi-channel representation of the $7 \times 7$ grid-world (Chevalier-Boisvert et al., 2019): the first three channels specify the presence or absence of the three different coloured boxes, the forth channel was the walls mask and the fifth and last channel was the position and orientation of the agent. We represent the orientation of the agent by ‘painting’ the cell in front of the agent. Therefore $s_t \in \{0, 1\}^{7 \times 7 \times 5}$.

2. **Action space, $A$:** We use the $\{\text{LEFT, RIGHT, FORWARD}\}$ actions from Minigrid (Chevalier-Boisvert et al., 2018) to navigate the maze, i.e., $a_t \in \{0, 1, 2\}$.

3. **Demonstrations, $D$:** At the beginning of training, the agent is given state-action pairs of other agents collecting either red or green coins.

A.4. Fruitbot

We build on the Fruitbot environment from OpenAI’s ProcGen benchmark (Cobbe et al., 2020). The task is specified by:

1. **State space, $S$:** We use the original high-dimensional $64 \times 64$ RGB observations, i.e., $s_t \in [0, 1]^{64 \times 64 \times 3}$.

2. **Action space, $A$:** We use the original 15 discrete actions, i.e., $a_t \in \{0, \ldots, 14\}$.

3. **Demonstrations, $D$:** At each time-step, the agent observes online the states and actions of 3 trained agents playing the game in parallel: One agent collects both fruits and other objects, one collects other objects and avoids fruits and the last one randomly selects actions.
B. Implementation Details

For our experiments we used Python (Van Rossum & Drake Jr, 1995). We used JAX (Bradbury et al., 2018; Babuschkin et al., 2020) as the core computational library, Haiku (Hennigan et al., 2020) and Acme (Hoffman et al., 2020) for implementing ΨΦ-learning and the baselines, see Section 4. We also used Matplotlib (Hunter, 2007) for the visualisations and Weight & Biases (Biewald, 2020) for managing the experiments.

B.1. Computation Graph

Figure 10. Computational graph of the ΨΦ-learning algorithm. Demonstrations D contain data from other agents for unknown tasks. We employ inverse temporal difference learning (ITD, see Section 3.1) to recover other agents’ successor features (SFs) and preferences. The ego-agent combines the estimated SFs of others along with its own preferences and successor features with generalised policy improvement (GPI, see Section 2.2), generating experience. Both the demonstrations and the ego-experience are used to learn the shared cumulants. Losses $\mathcal{L}_*$ are represented with double arrows and gradients flow according to the pointed direction(s).

B.2. Neural Network Architecture

Figure 11. Neural network architecture of the ΨΦ-learner. The rectangular nodes are tensors parametrised by MLPs and the circles are learnable vectors. We share an observation network/torso, $\mathcal{E}$, across all the network heads. The network heads that related to the other agents are in blue and trained from demonstrations D. The ego-agent’s experience $\mathcal{B}$ is used for training the green heads. The shared cumulants and torso are trained with both D and $\mathcal{B}$. An ensemble of two successor features approximators is used for the ego- and other-agents for combatting model overestimation, see Section 3.
B.3. Hyperparameters

Table 3. \(\Psi\Phi\)-learner’s hyperparameters per environment. The tuning was performed on a DQN (Mnih et al., 2013) baseline with population based training (Jaderberg et al., 2017) using Weights & Biases (Biewald, 2020) integration with Ray Tune (Liaw et al., 2018). We selected the best hyperparameters configuration out of 32 trials per environment and used this for our \(\Psi\Phi\)-learner.

<table>
<thead>
<tr>
<th></th>
<th>Highway</th>
<th>CoinGrid</th>
<th>Fruitbot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Torso network, (\mathcal{L})</td>
<td>MLP([512, 256])</td>
<td>IMPALA (Espeholt et al., 2018), shallow (no LSTM)</td>
<td>IMPALA (Espeholt et al., 2018), deep (no LSTM)</td>
</tr>
<tr>
<td>Cumulants approximator, (\Phi)</td>
<td>MLP([128, 128])</td>
<td>MLP([256, 128])</td>
<td>MLP([512, 256])</td>
</tr>
<tr>
<td>Successor features approximator, (\Psi)</td>
<td>MLP([256, 128])</td>
<td>MLP([512, 256])</td>
<td>MLP([512, 256])</td>
</tr>
<tr>
<td>Ensemble size, (\mathcal{E})</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(\ell_1) coefficient</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>Number of dimensions in (\Phi)</td>
<td>8</td>
<td>4</td>
<td>64</td>
</tr>
<tr>
<td>Minibatch size</td>
<td>512</td>
<td>64</td>
<td>32</td>
</tr>
<tr>
<td>(\pi)-step</td>
<td>4</td>
<td>8</td>
<td>128</td>
</tr>
<tr>
<td>Discount factor, (\gamma)</td>
<td>0.9</td>
<td>0.9</td>
<td>0.999</td>
</tr>
<tr>
<td>Target network update period</td>
<td>100</td>
<td>1000</td>
<td>2500</td>
</tr>
<tr>
<td>Optimiser</td>
<td>ADAM (Kingma &amp; Ba, 2014), (lr=1e-3)</td>
<td>ADAM (Kingma &amp; Ba, 2014), (lr=1e-4)</td>
<td>ADAM (Kingma &amp; Ba, 2014), (lr=5e-5)</td>
</tr>
</tbody>
</table>

B.4. Compute Resources

All the experiments were run on Microsoft Azure Standard_NC6s_v3 machines, i.e., with a 6-core vCPU, 112GB RAM and a single NVIDIA Tesla V100 GPU. The iteration cycle for (i) Highway experiments was 3 hours; (ii) CoinGrid experiments was 5.5 hours and (iii) Fruitbot experiments was 19 hours.
C. Proofs

First, we formalise the statement of Theorem 1.

**Theorem 1** (Validity of the ITD Minimiser). The minimisers of $\mathcal{L}_{BC-Q}$ and $\mathcal{L}_{ITD}$ are potentially-shaped cumulants that explain the observed reward-free demonstrations.

**Proof.** First, we prove the validity of the minimiser of the inverse temporal difference learning for the single-task setting. Next, we show that the result holds true in the vector (i.e., cumulants) case.

**Single task.** We assume that our demonstrations are generated by an expert, who samples actions from a Boltzmann policy, according to the optimal (for its task) action-value function $Q^\star(s, a)$ and temperature $\nu > 0$, i.e., $\mathcal{D} = \{(s, a)\} \sim \pi_{\text{expert}}$ s.t.

$$\pi_{\text{expert}}(a|s) \triangleq p(A = a|s) = \frac{\exp\left(\frac{1}{\nu}Q^\star_{\text{expert}}(s, a)\right)}{\sum_a \exp\left(\frac{1}{\nu}Q^\star_{\text{expert}}(s, a)\right)}, \forall s \in \mathcal{S}, a \in \mathcal{A}. \quad (15)$$

The minimiser of the behavioural cloning loss $\mathcal{L}_{BC-Q}(\theta_Q)$, i.e., Eqn. (8), for a single expert is s.t.

$$\theta_Q^* \in \arg\min_{\theta_Q} - \mathbb{E}_{(s, a) \sim \mathcal{D}} \log \frac{\exp(Q(s, a; \theta_Q))}{\sum_a \exp(Q(s, a; \theta_Q))} \Rightarrow Q(s, a; \theta_Q^*) = \frac{1}{\nu}Q^\star_{\text{expert}}(s, a) + F(s), \forall s \in \mathcal{S}, a \in \mathcal{A}, \quad (16)$$

$$\Rightarrow Q(s, a; \theta_Q^*) = \frac{1}{\nu}Q^\star_{\text{expert}}(s, a) + F(s), \forall s \in \mathcal{S}, a \in \mathcal{A}. \quad (17)$$

where $F : \mathcal{S} \mapsto \mathbb{R}$ is a state-dependent (bounded potential) function. We arrive at Eqn. (17) by (1) testing $\frac{1}{\nu}Q^\star_{\text{expert}}(s, a)$ as a solution and noting that the “softmax” function is convex in the exponent (Boyd et al., 2004) and (2) using the translation invariance property of the assumed Boltzmann policy parameterisation, i.e., for any $f : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ and $g : \mathcal{S} \mapsto \mathbb{R}$

$$\frac{\exp(f(s, a) + g(s))}{\sum_a \exp(f(s, a) + g(s))} = \frac{\exp(g(s)) \exp(f(s, a))}{\exp(g(s)) \sum_a \exp(f(s, a))} = \frac{\exp(f(s, a))}{\sum_a \exp(f(s, a))}. \quad (18)$$

The minimiser of the inverse temporal difference learning loss $\mathcal{L}_{ITD}(\theta_Q, \theta_r)$, Eqn. (9), for a single expert is s.t.

$$\theta_Q^*, \theta_r^* \in \arg\min_{\theta_Q, \theta_r} \mathbb{E}_{(s, a, s', a') \sim \mathcal{D}} \|Q(s, a; \theta_Q) - r(s, a; \theta_r) - \gamma Q(s', a'; \theta_Q)\| \quad (19)$$

where $\theta_Q^*$ is minimising $\mathcal{L}_{BC-Q}(\theta_Q)$ simultaneously, as in Eqn. (17). Therefore, it holds that $\mathcal{L}_{ITD}(\theta_Q^*, \theta_r^*) = 0$

$$r(s, a; \theta_r^*) = Q(s, a; \theta_Q^*) - \gamma Q(s', a'; \theta_Q^*) \quad (20)$$

$$\Rightarrow \frac{1}{\nu}Q^\star_{\text{expert}}(s, a) + F(s) - \gamma \frac{1}{\nu}Q^\star_{\text{expert}}(s', a') - \gamma F(s') \quad (21)$$

$$= \frac{1}{\nu} \left[ Q^\star_{\text{expert}}(s, a) - \gamma Q^\star_{\text{expert}}(s', a') \right] + F(s) - \gamma F(s') \quad (22)$$

$$= \frac{1}{\nu} r^\star_{\text{expert}}(s, a) + F(s) - \gamma F(s'), \quad (23)$$

where $r^\star_{\text{expert}}$ is the (unobserved) expert’s reward function. We have shown that the minimiser of $\mathcal{L}_{BC-Q}$ and $\mathcal{L}_{ITD}$ leads to a reward function $r(s, a; \theta_r^*)$ which is a potential-based shaped and scaled reward function of the expert reward function and hence the optimal policy for $r(s, a; \theta_r^*)$ is also optimal for $r^\star_{\text{expert}}(s, a)$ for all $s, a$ (Ng et al., 1999).

**Multiple tasks.** The minimiser of the behavioural cloning loss $\mathcal{L}_{BC-Q}(\theta_{q^k}, w^k)$, i.e., Eqn. (8), for the $k$-th expert is s.t.

$$\theta_{q^k}, w^k \in \arg\min_{\theta_{q^k}, w^k} - \mathbb{E}_{(s, a) \sim \mathcal{D}^k} \log \frac{\exp(\Psi(s, a; \theta_{q^k})^\top w^k)}{\sum_a \exp(\Psi(s, a; \theta_{q^k})^\top w^k)} \quad (24)$$

$$\Rightarrow \Psi(s, a; \theta_{q^k})^\top w^k = \frac{1}{\nu}Q^\star_{\text{expert}}(s, a) + F^k(s), \forall s \in \mathcal{S}, a \in \mathcal{A}, \quad (25)$$
where $Q_{\pi_{k-exp}}$ is the $k$-agent’s action-value function and $H^k : S \to \mathbb{R}$ a state-dependent (bounded potential) function. Next, the minimiser of the inverse temporal difference learning loss $\mathcal{L}_{\text{ITD}}(\Phi_{\phi_k}, \Phi_{\Psi})$, Eqn. (9), for the $k$-th expert is s.t.

$$\theta_{\phi_k}, \theta_{\Psi}^{*} \in \arg \min_{\theta_{\phi_k}, \theta_{\Psi}^{*}} \mathbb{E}_{(s, a, s', a') \sim D^k} \| \Psi(s, a; \theta_{\phi_k}) - \Phi(s, a; \theta_{\phi_k}) - \gamma \Psi(s', a'; \theta_{\phi_k}) \|$$

where $\theta_{\phi_k}^{*}$ is minimising $\mathcal{L}_{\text{BC-Q}}(\theta_{\phi_k}, w^k)$ simultaneously, as in Eqn. (25). Therefore, it holds that for $\mathcal{L}_{\text{ITD}}(\theta_{\phi_k}^{*}, \theta_{\phi_k}) = 0$

$$\begin{align*}
\Phi(s, a; \theta_{\phi_k}^{*}) &= \Psi(s, a; \theta_{\phi_k}^{*}) - \gamma \Psi(s', a'; \theta_{\phi_k}^{*}) \\
\Phi(s, a; \theta_{\phi_k}^{*})^\top w^{k*} &= \Psi(s, a; \theta_{\phi_k}^{*})^\top w^{k*} - \gamma \Psi(s', a'; \theta_{\phi_k}^{*})^\top w^{k*} \\
&\overset{(25)}{=} \frac{1}{\nu} Q_{\pi_{k-exp}}(s, a) + F^k(s) - \frac{1}{\nu} Q_{\pi_{k-exp}}(s', a') - \gamma F^k(s') \\
&= \frac{1}{\nu} \left[ Q_{\pi_{k-exp}}(s, a) - \gamma Q_{\pi_{k-exp}}(s', a') \right] + F^k(s) - \gamma F^k(s') \\
&= \frac{1}{\nu} Q_{\pi_{k-exp}}(s, a) + F^k(s) - \gamma F^k(s').
\end{align*}$$

We have shown that the minimiser of $\mathcal{L}_{\text{BC-Q}}$ and $\mathcal{L}_{\text{ITD}}$ leads to agent-agnostic cumulants $\Phi(s, a; \theta_{\phi_k}^{*})$ and agent-specific preference vector $w^k$, which when dot-produced, form a potential-based shaped and scaled reward function of the $k$-th expert reward function and hence the optimal policy for $\Phi(s, a; \theta_{\phi_k}^{*})^\top w^{k*}$ is also optimal for $r_{\pi_{k-exp}}(s, a)$ for all $s, a$ (Ng et al., 1999). The result holds for all $k \in \{1 \ldots K\}$ since no assumptions were made for the proof about $k$. \hfill \Box

Next, we formalise the statement of Theorem 2. When not specified the norm $\| \cdot \|$ refers to the 2-norm. Given a function $F : \mathcal{X} \to \mathbb{R}^d$ for some finite set $\mathcal{X}$, we will write $F(x)$ to denote the value of the function on input $x$ and $F$ to denote the matrix representation of this function in $||\mathcal{X}| \times d$.

**Theorem 2** (Generalisation Bound of $\Psi\Phi$-Learning). Let $C = (S, A, P, \gamma)$ be a CMP with a finite state space. Let $\phi : S \to \mathbb{R}^d$, and let $\Phi = \phi(S) \in \mathbb{R}^{|S| \times d}$. Let $(r_i)_{i=1}^k$ denote a set of reward functions on $C$, $\tilde{\Psi}$ be a collection of successor features approximations for policies $(\pi^i)_{i=1}^k (\pi_i$ optimal for $r_i$) with true successor feature values $\Psi^i$, and $\tilde{w}_i$ the best least-squares linear approximator of $r_i$ given $\Phi$, with errors

$$\|\Phi \tilde{w}_i - r_i\|_\infty < \delta_r \text{ and } \|\tilde{\Psi}^i - \Psi^i\| < \delta_\Psi \quad \forall i.$$

Let $w'$ be a new preference vector for a reward function $r'$, with maximal error $\delta_r$, as well. Let $\tilde{Q}^i = \tilde{\Psi}^i w'$. Let $\pi^*$ be the optimal policy for the ego task $w'$ and let $\pi$ be the GPI policy obtained from $\{Q^{\pi^i}\}$, with $\delta_r, \delta_\Psi$ the reward and successor feature approximation errors. Then for all $s, a$

$$Q^*(s, a) - Q^*(s, a) \leq \frac{2}{1 - \gamma} \left[ (\phi_{\max} \|w_j - w'\| + 2\delta_r) + \|w'\|\delta_\Psi + \frac{1}{(1 - \gamma)} \delta_r \right].$$

Barreto et al. (2017) construct their bound on the sub-optimality of the GPI policy as a function of the error of the value approximations $\tilde{Q}^i$. Because we bound the reward approximation error, rather than the value approximation error, we require an additional step to obtain a bound on the errors of the value function approximations. To prove Theorem 1, we must therefore first use the following lemma to bound the effect of the reward approximation error on the value approximation error. While this result is straightforward, we include a short proof for completeness.

**Lemma 1.** Fix some policy $\pi$. Let $r$ be reward vector and let $w$ be the least-squares solution to $\min \|\Phi w - r\|$. Let $\tilde{\Psi}$ be the true successor features for $\Phi$ under policy $\pi$, and let $Q^\pi$ be the value. Let $\delta_r = R(S) - \Phi w, \delta_{\max} = \|\delta_r\|_\infty$. Then letting $Q = \tilde{\Psi} w$, we have

$$\|Q^\pi - \tilde{Q}\|_\infty \leq \frac{1}{1 - \gamma} \delta_r$$

**Proof.**

$$\|Q^\pi - \tilde{Q}\|_\infty \leq \sum \gamma t \|P^{\pi t}(\Phi u - r)\|_\infty \leq \sum \gamma t \|P^{\pi t}\|_\infty = \sum \gamma t \max_{s' \in S} \sum_{s' \in S} |(P^\pi)^t(s', s)| \delta_r(s)$$

\[\|Q^\pi - \tilde{Q}\|_\infty \leq \sum \gamma t \|P^{\pi t}(\Phi u - r)\|_\infty \leq \sum \gamma t \max_{s' \in S} \sum_{s' \in S} |(P^\pi)^t(s', s)| \delta_r(s)\]
Since $P^n$ is a stochastic matrix, so are all of its powers, and so the rows of $(P^n)^t$ sum to 1.

$$\sum_t \gamma^t \max_{s'} \sum_{s} P^n(t, s') \delta_{\max} = \sum \gamma^t \delta_{\max} \leq \frac{1}{1 - \gamma} \delta_{\max}$$ (36)

$$\leq \frac{2}{1 - \gamma} \|r_j - r'\|_{\infty} + \frac{2}{1 - \gamma} \epsilon$$ (Barreto et al., 2017, Theorem 1)

$$\leq \frac{2}{1 - \gamma} \phi_{\max} \|w_j - w\|_{L} + \delta_r + \delta_r + \frac{2}{1 - \gamma} \epsilon$$ (Barreto et al., 2017, Lemma 1)

$$\leq \frac{2}{1 - \gamma} \phi_{\max} \|w_j - w'\|_{L} + \delta_r + \delta_r + \frac{2}{1 - \gamma} \epsilon$$ (Lemma 1)

We now prove the main result.

**Proof.** We follow the proof of Barreto et al. (2017, Theorem 2), with additional error terms to account for the reward and successor feature approximation errors.

$$Q^*(s, a) - Q^\pi(s, a) \leq Q^*(s, a) - Q^\pi_j(s, a) + \frac{2}{1 - \gamma} \epsilon$$

$$\leq \frac{2}{1 - \gamma} \|r_j - r'\|_{\infty} + \frac{2}{1 - \gamma} \epsilon$$ (Barreto et al., 2017, Theorem 1)

$$\leq \frac{2}{1 - \gamma} \phi_{\max} \|w_j - w\|_{L} + \delta_r + \delta_r + \frac{2}{1 - \gamma} \epsilon$$ (Barreto et al., 2017, Lemma 1)

$$\leq \frac{2}{1 - \gamma} \phi_{\max} \|w_j - w'\|_{L} + \delta_r + \delta_r + \frac{2}{1 - \gamma} \epsilon$$ (Lemma 1)

$$\leq \frac{2}{1 - \gamma} \phi_{\max} \|w_j - w'\|_{L} + \delta_r + \delta_r + \frac{2}{1 - \gamma} \epsilon$$ (Lemma 1)
D. Algorithms

Algorithm 1: Inverse Temporal Difference Learning

Input:  
\[ D = \{(s_1, a_1, \ldots, a_T; k)_{k=1}^{K}\} \]  
No-reward demonstrations  
\[ \lambda_w \]  
\[ L_1 \] loss coefficient

Output:  
\[ \theta_\phi \]  
Parameters of cumulants network  
\[ \{\theta_{\psi^k}\}_{k=1}^{K} \]  
Parameters of successor features approximators  
\[ \{w^k\}_{k=1}^{K} \]  
Preferences vectors for the \( K \) agents

// initialisations
1 Initialise parameters \( \theta_\phi, \{\theta_{\psi^k}, w^k\}_{k=1}^{K} \)

2 while budget do

3 Sample trajectories \( \{\tau_i = (s_1^{(i)}, a_1^{(i)}, \ldots, s_T^{(i)}, a_T^{(i)}; k^{(i)})\}_{i=1}^{N} \sim D \)

4 Calculate behavioural cloning loss \( \mathcal{L}_{BC-Q}(\theta_{\psi^k}, w^k) \) on samples \( \{\tau_i\}_{i=1}^{N} \)  \( \triangleright \) see Eqn. (8)

5 \[ \theta_{\psi^k} \leftarrow \nabla_{\theta_{\psi^k}} \mathcal{L}_{BC-Q}(\theta_{\psi^k}, w^k) \]  \( \triangleright \) update \( \psi \)

6 \[ w^k \leftarrow \nabla_{w^k} \left( \mathcal{L}_{BC-Q}(\theta_{\psi^k}, w^k) + \lambda_w \| w^k \|_1 \right) \]  \( \triangleright \) update \( w \)

7 Calculate inverse temporal difference loss \( \mathcal{L}_{ITD}(\theta_\phi, \theta_{\psi^k}) \) on samples \( \{\tau_i\}_{i=1}^{N} \)  \( \triangleright \) see Eqn. (9)

8 \[ \theta_\phi \leftarrow \nabla_{\theta_\phi} \mathcal{L}_{ITD}(\theta_{\psi^k}, \theta_\phi) \]  \( \triangleright \) update \( \phi \)

9 \[ \theta_{\psi^k} \leftarrow \nabla_{\theta_{\psi^k}} \mathcal{L}_{ITD}(\theta_{\psi^k}, \theta_\phi) \]  \( \triangleright \) update \( \psi \)
Algorithm 2: $\Psi \Phi$-Learning

**Input:**

- $D = \{(s_1, a_1, \ldots, a_T; k)_{k=1}^{K}\}$ No-reward demonstrations
- $\lambda_w$ $L_1$ loss coefficient

**Output:**

- $\theta_{\psi_{ego}}$ Ego successor features approximator
- $\theta_{\phi}$ Parameters of cumulants network
- $\{\theta_{\psi_k}\}_{k=1}^{K}$ Parameters of successor features approximators
- $\{w^{k}\}_{k=1}^{K}$ Preferences vectors for the $K$ agents

// initialisations
1. Empty replay buffer for ego-experience $B = \{}$
2. Initialise parameters $\theta_{\psi_{ego}}, w^{ego}, \theta_{\phi}, \{\theta_{\psi_k}, w^{k}\}_{k=1}^{K}$
3. while budget do

- // agent-environment interaction
  4. Reset episode, $s \leftarrow \text{env.reset}(), t \leftarrow 0$
  5. while not done do
  6. $w^{ego} \leftarrow \text{arg min}_{w} L_{R}(\theta_{\phi}, w; B)$ ◦ ego-task inference, see Eqn. (10)
  7. $a \leftarrow \pi_{\text{GPI}}(s; \psi^{ego}, w^{ego}, \{\theta_{\psi_k}\}_{k=1}^{K})$ ◦ GPI, see Eqn. (13)
  8. Step in the environment, $s', r^{ego}, \text{done} \leftarrow \text{env.step}(a)$
  9. Append transition in the replay buffer, $B \leftarrow B \cup (s, a, r^{ego}, s')$
  10. $s' \leftarrow s, t \leftarrow t + 1$

- (online demonstrations) Append demonstrations in $D$ ◦ optional

- // parameter updates/learning
  11. $\theta_{\phi}, \{\theta_{\psi_k}, w^{k}\}_{k=1}^{K} \leftarrow \text{ITD}(D, \lambda_w, \theta_{\phi}, \{\theta_{\psi_k}, w^{k}\}_{k=1}^{K})$ ◦ see Algorithm. (1)
  12. Sample transitions $\{(s^{(i)}, a^{(i)}, r^{ego,(i)}, s^{(i)})\} \sim B$
  13. Calculate the reward loss $L_{R}(\theta_{\phi}, w^{ego})$ ◦ see Eqn. (10)
  14. $\theta_{\phi} \leftarrow \nabla_{\theta_{\phi}} L_{R}(\theta_{\phi}, w^{ego})$ ◦ update $\Phi$
  15. Calculate TD losses $L_{Q}(\theta_{\psi_{ego}})$ and $L_{TD-\psi}(\theta_{\psi_{ego}})$ ◦ see Eqn. (11,12)
  16. $\theta_{\psi_{ego}} \leftarrow \nabla_{\theta_{\psi_{ego}}} \left( L_{Q}(\theta_{\psi_{ego}}) + \frac{1}{|\psi|} L_{TD-\psi}(\theta_{\psi_{ego}}) \right)$ ◦ update $\Psi^{ego}$
E. Visualisations

Figure 12. Qualitative evaluation of the learned cumulants in the CoinGrid task. Cumulants $\phi_1$, $\phi_2$, and $\phi_3$ seem to capture the red, green, and yellow blocks, respectively. The yellow blocks are captured by both $\phi_2$ and $\phi_3$. Therefore, linear combinations of the learned cumulants can represent arbitrary rewards in the environment, which involve stepping on the coloured blocks.

Figure 13. Sensitivity of our ITD (see Section 3.1) and $\Psi\Phi$-learning (see Section 3.2) algorithms to the dimensionality of the learned cumulants. We consistently observe across all three experiments (a)-(c) that for a small number of $\Phi$ dimensions the cumulants are not expressive enough to capture the axis of variation of the different agents’ reward functions (including the ego-agent in (c)). We also note that the performance of both ITD and $\Psi\Phi$-learning is relatively robust for a medium and large number of $\Phi$ dimensions. We attribute this to the used sparsity prior, i.e., $L_1$ loss, to the preferences $w$. In our experiments we selected the smallest number of $\Phi$ dimensions that demonstrated good performance to keep the number of model parameters as small as possible (in bold in the figures and reported in Table 3).