Transfer in Deep Reinforcement Learning Using Successor Features and Generalised Policy Improvement

Supplementary Material

André Barreto, **Diana Borsa**, **John Quan**, **Tom Schaul**, **David Silver**,
Matteo Hessel, **Daniel Mankowitz Augustin Žídek**, **Rémi Munos**,
{andrebarreto,borsa,johnquan,schaul,davidsilver,
mtthss,dmankowitz,augustinzidek,munos}@google.com

DeepMind

Abstract

In this supplement we give details of the theory and experiments that had to be left out of the main paper due to the space limit. For the convenience of the reader the statements of the theoretical results are reproduced before the respective proofs. We also report additional empirical analysis that could not be included in the paper. The citations in this supplement refer to the references listed in the main paper.

A. Proof of theoretical results

We restate Barreto et al.'s (2017) GPI theorem to be used as a reference in the derivations that follow.

Theorem 1. (Generalized Policy Improvement) *Let π_1, π_2, ..., π_n be n decision policies and let $\tilde{Q}^{\pi_1}$, $\tilde{Q}^{\pi_2}$, ..., $\tilde{Q}^{\pi_n}$ be approximations of their respective action-value functions such that*

$$|Q^{\pi_i}(s,a) - \tilde{Q}^{\pi_i}(s,a)| \leq \epsilon \text{ for all } s \in S, a \in A, \text{ and } i \in \{1,2,...,n\}.$$

Define

$$\pi(s) \in \operatorname*{argmax}_a \max_i \tilde{Q}^{\pi_i}(s,a).$$

Then,

$$Q^{\pi}(s,a) \geq \max_i Q^{\pi_i}(s,a) - \frac{2}{1-\gamma}\epsilon$$

for any $s \in S$ and any $a \in A$, where Q^{π} is the action-value function of π.

Lemma 1. *Let $\delta_{ij} = \max_{s,a} |r_i(s,a) - r_j(s,a)|$ and let π be an arbitrary policy. Then,*

$$|Q_i^{\pi}(s,a) - Q_j^{\pi}(s,a)| \leq \frac{\delta_{ij}}{1-\gamma}.$$

Proof. Define $\Delta_{ij} = \max_{s,a} |Q_i^{\pi}(s,a) - Q_j^{\pi}(s,a)|$. Then,

$$\begin{aligned}
|Q_i^{\pi}(s,a) - Q_j^{\pi}(s,a)| &= \left| r_i(s,a) + \gamma \sum_{s'} p(s'|s,a) Q_i^{\pi}(s', \pi(s')) - r_j(s,a) - \gamma \sum_{s'} p(s'|s,a) Q_j^{\pi}(s', \pi(s')) \right| \\
&= \left| r_i(s,a) - r_j(s,a) + \gamma \sum_{s'} p(s'|s,a) \left(Q_i^{\pi}(s', \pi(s')) - Q_j^{\pi}(s', \pi(s')) \right) \right| \\
&\leq |r_i(s,a) - r_j(s,a)| + \gamma \sum_{s'} p(s'|s,a) \left| Q_i^{\pi}(s', \pi(s')) - Q_j^{\pi}(s', \pi(s')) \right| \\
&\leq \delta_{ij} + \gamma \Delta_{ij}. \qquad (11)
\end{aligned}$$

Since (11) is valid for any $s, a \in S \times A$, we have shown that $\Delta_{ij} \leq \delta_{ij} + \gamma\Delta_{ij}$. Solving for Δ_{ij} we get

$$\Delta_{ij} \leq \frac{1}{1-\gamma}\delta_{ij}.$$

□

Lemma 2. *Let $\delta_{ij} = \max_{s,a} |r_i(s,a) - r_j(s,a)|$. Then,*

$$|Q_i^{\pi_i^*}(s,a) - Q_j^{\pi_j^*}(s,a)| \leq \frac{\delta_{ij}}{1-\gamma}.$$

Proof. To simplify the notation, let $Q_i^i(s,a) \equiv Q_i^{\pi_i^*}(s,a)$. Note that $|Q_i^i(s,a) - Q_j^j(s,a)|$ is the difference between the value functions of two MDPs with the same transition function but potentially different rewards. Let $\Delta_{ij} = \max_{s,a} |Q_i^i(s,a) - Q_j^j(s,a)|$. Then,

$$\begin{aligned}
|Q_i^i(s,a) - Q_j^j(s,a)| &= \left| r_i(s,a) + \gamma \sum_{s'} p(s'|s,a) \max_b Q_i^i(s',b) - r_j(s,a) - \gamma \sum_{s'} p(s'|s,a) \max_b Q_j^j(s',b) \right| \\
&= \left| r_i(s,a) - r_j(s,a) + \gamma \sum_{s'} p(s'|s,a) \left(\max_b Q_i^i(s',b) - \max_b Q_j^j(s',b) \right) \right| \\
&\leq |r_i(s,a) - r_j(s,a)| + \gamma \sum_{s'} p(s'|s,a) \left| \max_b Q_i^i(s',b) - \max_b Q_j^j(s',b) \right| \\
&\leq |r_i(s,a) - r_j(s,a)| + \gamma \sum_{s'} p(s'|s,a) \max_b \left| Q_i^i(s',b) - Q_j^j(s',b) \right| \\
&\leq \delta_{ij} + \gamma\Delta_{ij}. \qquad (12)
\end{aligned}$$

Since (12) is valid for any $s, a \in S \times A$, we have shown that $\Delta_{ij} \leq \delta_{ij} + \gamma\Delta_{ij}$. Solving for Δ_{ij} we get

$$\Delta_{ij} \leq \frac{1}{1-\gamma}\delta_{ij}.$$

□

Proposition 1. *Let $M \in \mathcal{M}$ and let $Q_i^{\pi_j^*}$ be the action-value function of an optimal policy of $M_j \in \mathcal{M}$ when executed in $M_i \in \mathcal{M}$. Given approximations $\{\tilde{Q}_i^{\pi_1}, \tilde{Q}_i^{\pi_2}, ..., \tilde{Q}_i^{\pi_n}\}$ such that $\left| Q_i^{\pi_j^*}(s,a) - \tilde{Q}_i^{\pi_j}(s,a) \right| \leq \epsilon$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $j \in \{1, 2, ..., n\}$, let*

$$\pi(s) \in \text{argmax}_a \text{max}_j \tilde{Q}_i^{\pi_j}(s,a).$$

Then,

$$\|Q^* - Q^\pi\|_\infty \leq \frac{2}{1-\gamma}\left(\|r - r_i\|_\infty + \min_j \|r_i - r_j\|_\infty + \epsilon \right),$$

where Q^* is the optimal value function of M, Q^π is the value function of π in M, and $\|f-g\|_\infty = \max_{s,a} |f(s,a) - g(s,a)|$.

Proof. The result is a direct application of Theorem 1 and Lemmas 1 and 2. Let π^* be an optimal value function of M. Then,

$$\begin{aligned}
Q^*(s,a) - Q^\pi(s,a) &= Q^{\pi^*}(s,a) - Q^\pi(s,a) \\
&= Q^{\pi^*}(s,a) - Q_i^{\pi_i^*} + Q_i^{\pi_i^*} - Q^\pi(s,a) \\
&= Q^{\pi^*}(s,a) - Q_i^{\pi_i^*} + Q_i^{\pi_i^*} - Q_i^\pi + Q_i^\pi - Q^\pi(s,a) \\
&\leq |Q^{\pi^*}(s,a) - Q_i^{\pi_i^*}| + Q_i^{\pi_i^*} - Q_i^\pi + |Q_i^\pi - Q^\pi(s,a)|
\end{aligned}$$

From Lemma 2, we know that

$$|Q^{\pi^*}(s,a) - Q_i^{\pi_i^*}| \leq \frac{\max_{s,a} |r(s,a) - r_i(s,a)|}{1-\gamma}.$$

From Theorem 1 we know that, for any $j \in \{1, 2, ..., n\}$, we have

$$\begin{aligned} Q_i^{\pi_i^*}(s,a) - Q_i^{\pi}(s,a) &\leq Q_i^{\pi_i^*}(s,a) - Q_i^{\pi_j^*}(s,a) + \frac{2}{1-\gamma}\epsilon && \text{(Theorem 1)} \\ &= Q_i^{\pi_i^*}(s,a) - Q_j^{\pi_j^*}(s,a) + Q_j^{\pi_j^*}(s,a) - Q_i^{\pi_j^*}(s,a) + \frac{2}{1-\gamma}\epsilon \\ &\leq |Q_i^{\pi_i^*}(s,a) - Q_j^{\pi_j^*}(s,a)| + |Q_j^{\pi_j^*}(s,a) - Q_i^{\pi_j^*}(s,a)| + \frac{2}{1-\gamma}\epsilon \\ &\leq \frac{2}{1-\gamma}\max_{s,a}|r_i(s,a) - r_j(s,a)| + \frac{2}{1-\gamma}\epsilon && \text{(Lemmas 1 and 2).} \end{aligned} \tag{13}$$

Finally, from Lemma 1, we know that

$$|Q_i^{\pi} - Q^{\pi}(s,a)| \leq \frac{\max_{s,a}|r(s,a) - r_i(s,a)|}{1-\gamma}.$$

□

B. Details of the experiments

In this section we give details of the experiments that had to be left out of the main paper due to the space limit.

B.1. Agents's architecture

The CNN used in Figure 2 is identical to that used by Mnih et al.'s (2015) DQN. The CNN outputs a 256-dimensional vector that serves as the LSTM state. As shown in Figure 2, the LSTM also receives the previous action of the agent as an input. The output of the LSTM is a vector of dimension 256, which in the paper we call the state signal $\tilde{s}$. The vector $\tilde{s}$ is the input of the $D + 1$ MLPs used to compute $\tilde{\boldsymbol{\phi}}$ and $\tilde{\boldsymbol{\psi}}^{\pi_i}$. These MLPs have 100 $\tanh$ hidden units and an output of dimension $D \times |\mathcal{A}|$—that is, for each action $a \in \mathcal{A}$ the MLP outputs a D-dimensional vector representing either $\tilde{\boldsymbol{\phi}}$ or one of the $\tilde{\boldsymbol{\psi}}^{\pi_i}$. These D-dimensional vectors are then multiplied by $\tilde{\mathbf{w}}$, leading to a $(D + 1) \times |\mathcal{A}|$ output representing $\tilde{r}$ and $\tilde{Q}^{\pi_i}$.

B.2. Agents's training

The losses shown in lines 9 and 13 of Algorithm 1 and in lines 6 and 9 of Algorithm 2 were minimised using the RMSProp method, a variation of the well-known back-propagation algorithm. As parameters of RMSProp we adopted a fixed decay rate of 0.99 and $\epsilon = 0.01$. For all algorithms we tried at least two values for the learning rate: 0.01 and 0.001. For the baselines "$DQ(\lambda)$ fine tuning" and "$DQ(\lambda)$ from scratch" we also tried a learning rate of 0.005. The results shown in the paper are those associated with the best final performance of each algorithm.

As mentioned in the paper, the agents's training was carried out using the IMPALA architecture (Espeholt et al., 2018). In IMPALA the agent is conceptually divided in two groups: "actors", which interact with the environment in parallel collecting trajectories and adding them to a queue, and a "learner", which pulls trajectories from the queue and uses them to apply the updates. On the learner side, we adopted a simplified version of IMPALA that uses $Q(\lambda)$ as the RL algorithm (*i.e.*, no parametric representation of policies nor off-policy corrections). For the distributed collection of data we used 50 actors per task. Each actor gathered trajectories of length 20 that were then added to the common queue. The collection of data followed an ϵ-greedy policy with a decaying ϵ. Specifically, the value of ϵ started at 0.5 and decayed linearly to 0.05 in 10^6 steps. The results shown in the paper correspond to the performance of the ϵ-greedy policy (that is, they *include* exploratory actions of the agents).

For the results with SF&GPI-continual, in addition to the loss induced by equation (5), minimised in line 14 of Algorithm 1, we also used a standard $Q(\lambda)$ loss—that is, the gradients associated with both losses were combined through a weighted sum and then used to update $\boldsymbol{\theta}_\psi$. The weights for the standard $Q(\lambda)$ loss and the loss computed in line 13 of Algorithm 1 were 1 and 0.1, respectively. Using the standard $Q(\lambda)$ loss seems to stabilise the learning of $\tilde{\boldsymbol{\psi}}^{\pi_{n+1}}$; in this case (5) can be seen as a constraint for the standard RL optimisation. Obviously, if we want to add $\tilde{\boldsymbol{\psi}}^{\pi_{n+1}}$ to $\tilde{\Psi}$, we have to make sure that the SF semantics is preserved—that is, the result of the combined updates approximately satisfies (5). We confirmed this fact by monitoring the loss computed in line 13 of Algorithm 1. Figure 5 shows the average of this loss computed over 10 runs of SF&GPI-continual on all test tasks; as shown in the figure, the loss is indeed minimised, which implies that the resulting $\tilde{\boldsymbol{\psi}}^{\pi_{n+1}}$ are valid SFs that can be safely added to $\tilde{\Psi}$.

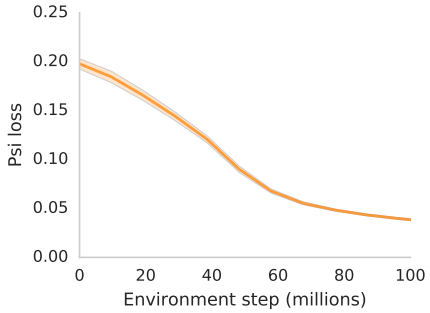


Figure 5. Loss in the approximation of $\tilde{\psi}^{\pi_{n+1}}$ (line 13 of Algorithm 1). Shaded region represents one standard deviation over 10 runs on all test tasks.

B.3. Environment

The room used in the environment and illustrated in Figure 1(a) was of size 13×13 (Beattie et al., 2016). The observations o_t were an 84×84 image with pixels re-scaled to the interval $[0, 1]$. The action space $\mathcal{A}$ contains 8 actions: move forward, move backwards, strafe left, strafe right, look left, look right, look left and move forward, and look right and move forward. Each action was repeated for 4 frames, that is, the agent was allowed to choose an action at every 4 observations (we note that the "environment steps" shown in the plots refer to actual observations, not the number of decisions made by the agent).

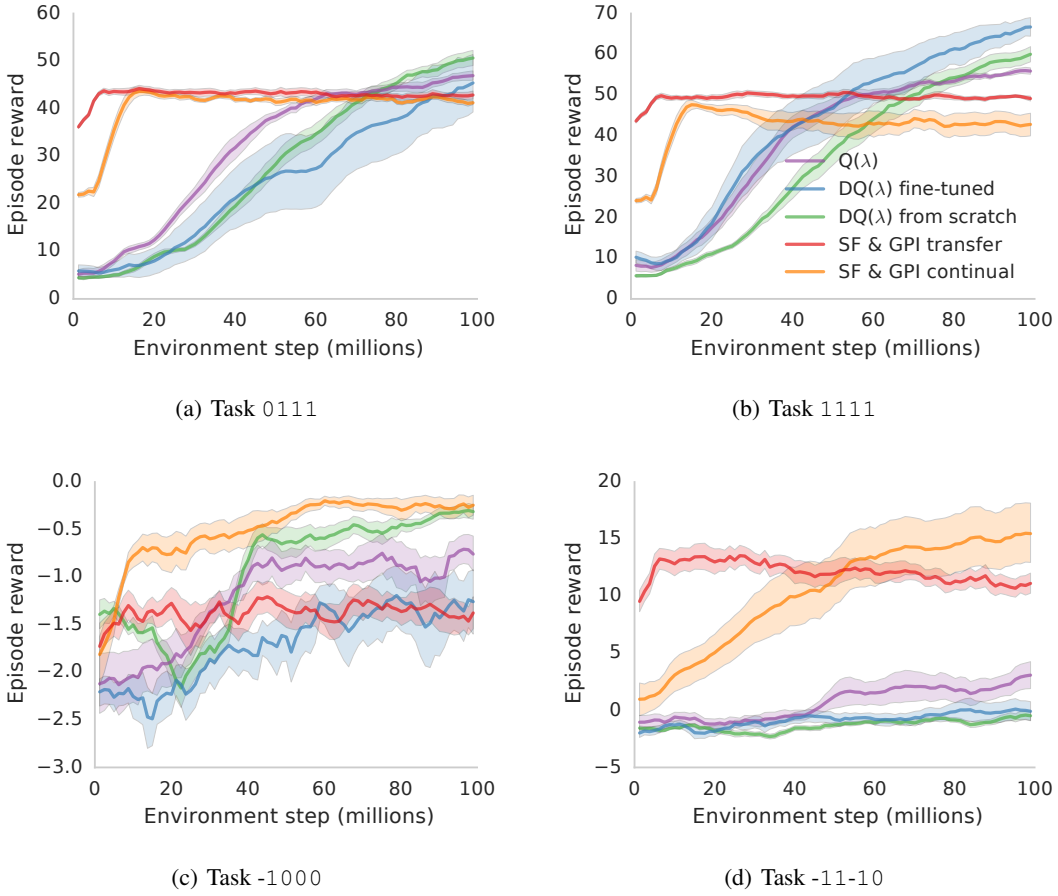
C. Additional results

In our experiments we defined a set of 9 test tasks in order to cover reasonably well three qualitatively distinct combinations of rewards: only positive rewards, only negative rewards, and mixed rewards. Figure 6 shows the results of SF&GPI-transfer and the baselines on the test tasks that could not be included in the paper due to the space limit.

As discussed in the paper, ideally our agent should rely on the GPI policy when useful but also be able to learn and use a specialised policy otherwise. Figures 7, 8 and 9 show that this is possible with SF&GPI-continual. Looking at Figure 7 we see that when the test task only has positive rewards the performances of SF&GPI-transfer and SF&GPI-continual are virtually the same. This makes sense, since in this case alternating between the policies π_i learned on $\hat{\mathcal{M}}$ should lead to good performance. Although initially the specialised policy π_{test} does get selected by GPI a few times, eventually the policies π_i largely dominate. The figure also corroborates the hypothesis that GPI is in general not computing a trivial policy, since even after settling on the policies π_i it keeps alternating between them.

Interestingly, when we look at the test tasks with negative rewards this pattern is no longer observed. As shown in Figures 8 and 9, in this case SF&GPI-continual eventually outperforms SF&GPI-transfer—which is not surprising. Looking at the frequency at which policies are selected by GPI, we observe the opposite trend as before: now the policy π_{test} steadily becomes the preferred one. The fact that a specialised policy is learned and eventually dominates is reassuring, as it indicates that π_{test} will contribute to the repository of skills available to the agent when added to $\tilde{\Psi}$.

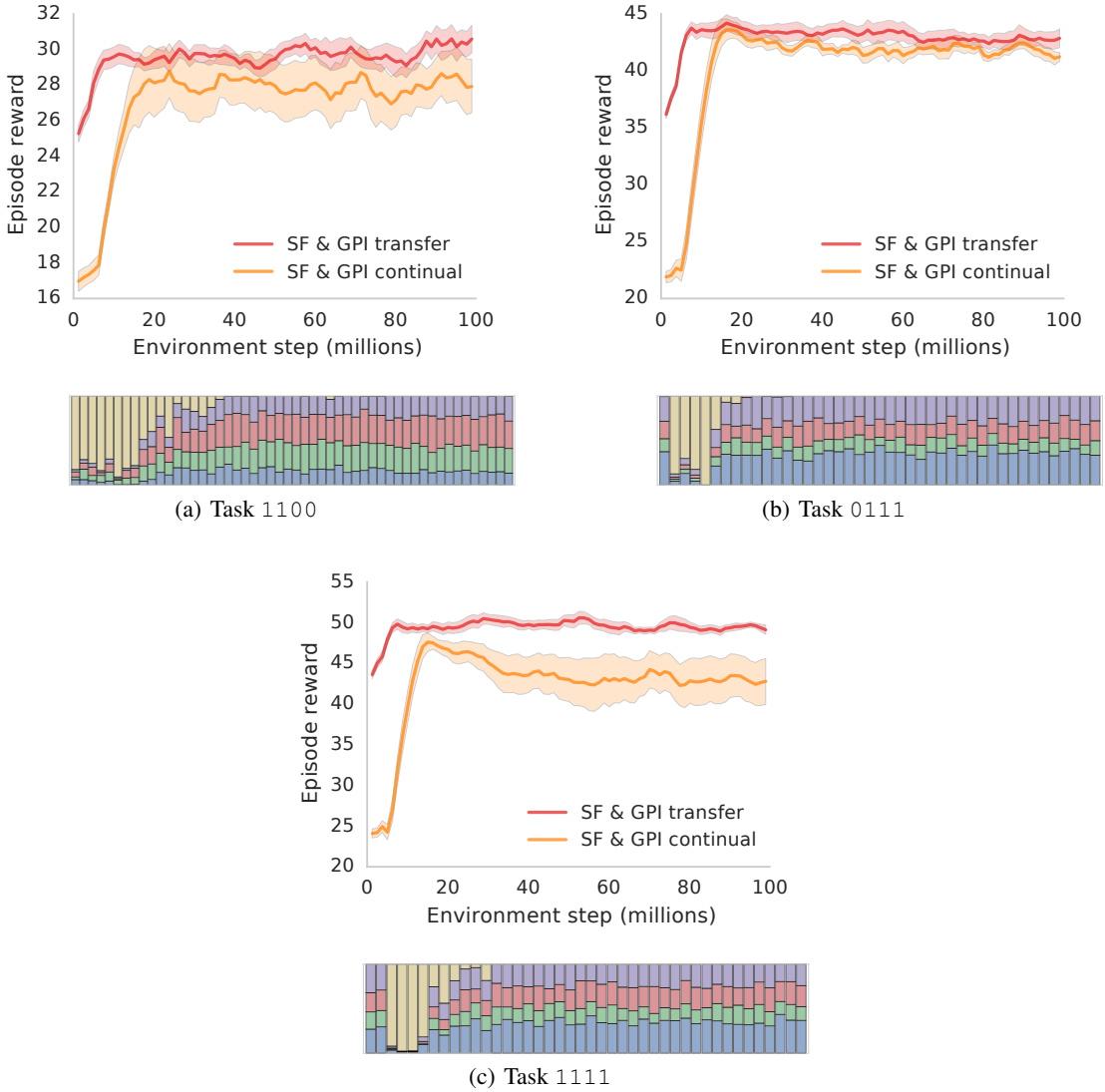
As discussed in the main paper, one of the highlights of the proposed algorithm is the fact that it can use any set of base tasks and enable transfer to related, unseen, tasks. To empirically verify this claim, we tested several sets of base tasks. We used as a reference our "canonical" set of base tasks $\hat{\mathcal{M}}$ that spans the environment $\mathcal{M}$. We further validated our approach on a linearly-depend set of base tasks $\hat{\mathcal{M}}'$ that does not span the set of (test) tasks we are interested in—these results are shown in Figure 4. In addition to these, we now present experiments with a third set of base tasks that does span the set of tasks but does not include any of the canonical tasks: $\hat{\mathcal{M}}'' = \{(1, \text{-}0.1, \text{-}0.1, \text{-}0.1), (\text{-}0.1, 1, \text{-}0.1, \text{-}0.1), (\text{-}0.1, \text{-}0.1, 1, \text{-}0.1), (\text{-}0.1, \text{-}0.1, \text{-}0.1, 1)\}$. In Figure 10 we report results obtained by SF&GPI-transfer, using the three sets of base tasks described, on the 9 previously-introduced test tasks. We can see that all sets of base tasks lead to satisfactory transfer, but the performance of the transferred policy can vary significantly depending on the relation between the base tasks used and the test task. This is again an illustration of the distinction between the "reward basis" and the "behaviour basis" discussed in Section 5.3.



(a) Task `0111` (b) Task `1111`

(c) Task `-1000` (d) Task `-11-10`

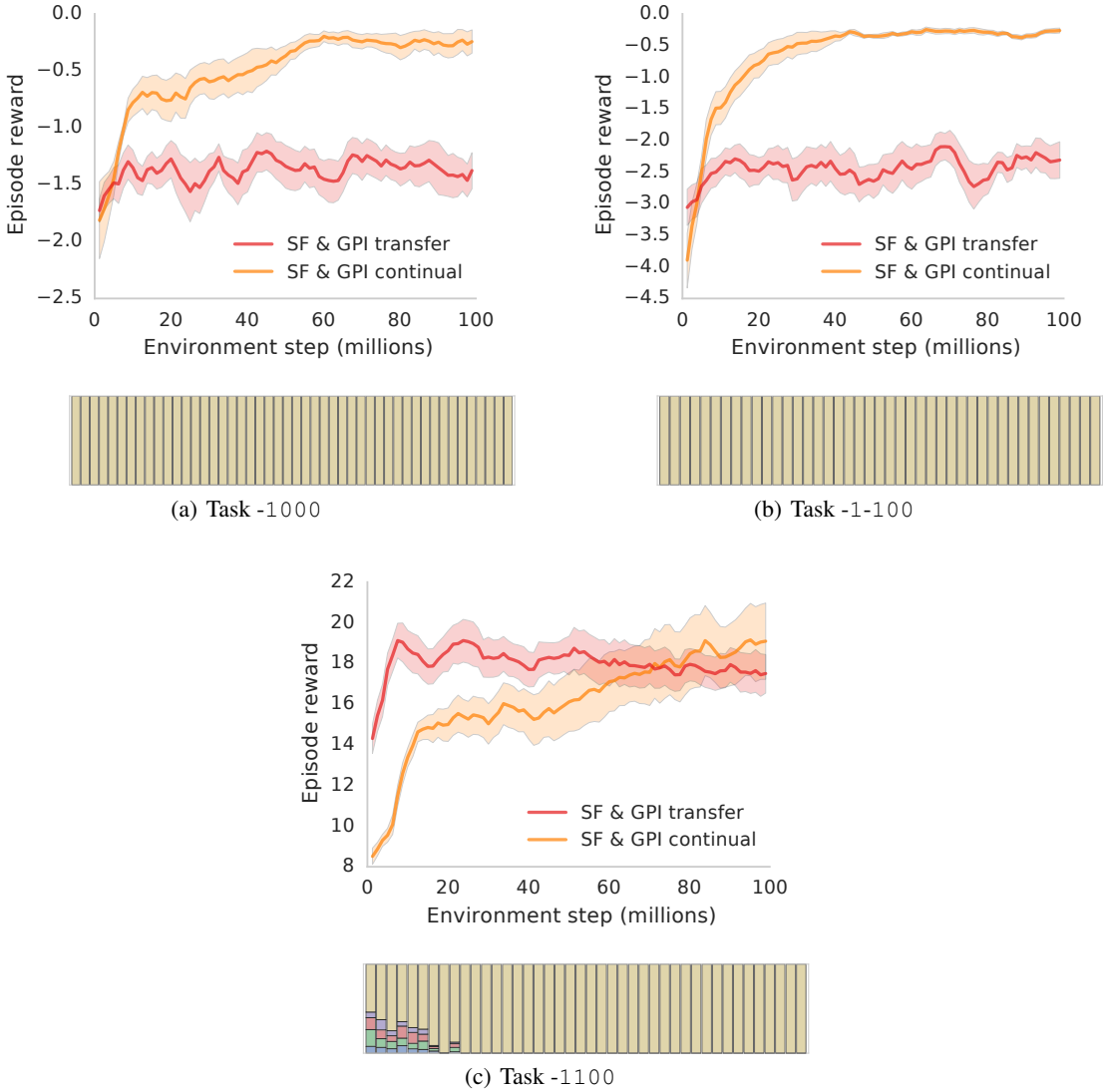
Figure 6. Average reward per episode on test tasks not shown in the main paper. The x axes have different scales because the amount of reward available changes across tasks. Shaded regions are one standard deviation over 10 runs.



(a) Task `1100` (b) Task `0111`

(c) Task `1111`

Figure 7. **Top figures**: Comparison between SF&GPI-transfer and SF&GPI-continual. Shaded regions are one standard deviation over 10 runs. All the runs of SF&GPI-transfer and SF&GPI-continual used the same basis $\tilde{\Psi}$. **Bottom figures**: Coloured bar segments represent the frequency at which the policies π_i were selected by GPI in one run of SF&GPI-continual, with each colour associated with a specific policy. The policy π_{test} specialised to the task is represented in light yellow.

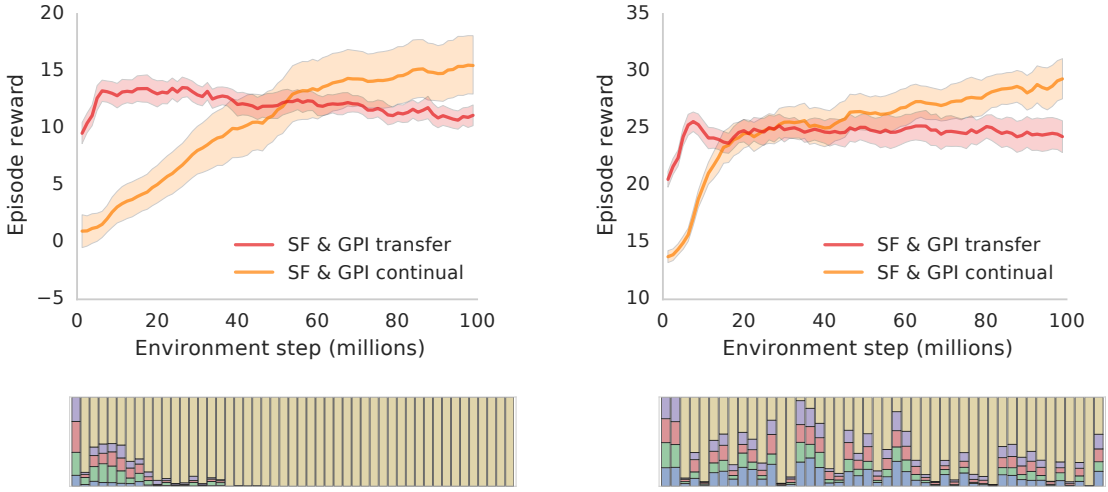


(a) Task -1000

(b) Task -1-100

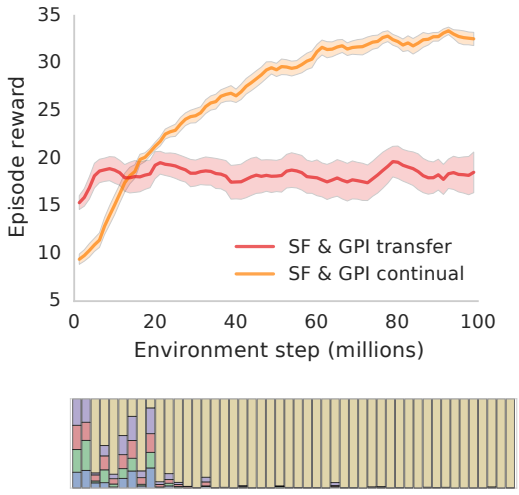
(c) Task -1100

Figure 8. **Top figures**: Comparison between SF&GPI-transfer and SF&GPI-continual. Shaded regions are one standard deviation over 10 runs. All the runs of SF&GPI-transfer and SF&GPI-continual used the same basis $\tilde{\Psi}$. **Bottom figures**: Coloured bar segments represent the frequency at which the policies π_i were selected by GPI in one run of SF&GPI-continual, with each colour associated with a specific policy. The policy π_{test} specialised to the task is represented in light yellow.



(a) Task -11-10

(b) Task -1101



(c) Task -11-11

Figure 9. **Top figures**: Comparison between SF&GPI-transfer and SF&GPI-continual. Shaded regions are one standard deviation over 10 runs. All the runs of SF&GPI-transfer and SF&GPI-continual used the same basis $\tilde{\Psi}$. **Bottom figures**: Coloured bar segments represent the frequency at which the policies π_i were selected by GPI in one run of SF&GPI-continual, with each colour associated with a specific policy. The policy π_{test} specialised to the task is represented in light yellow.

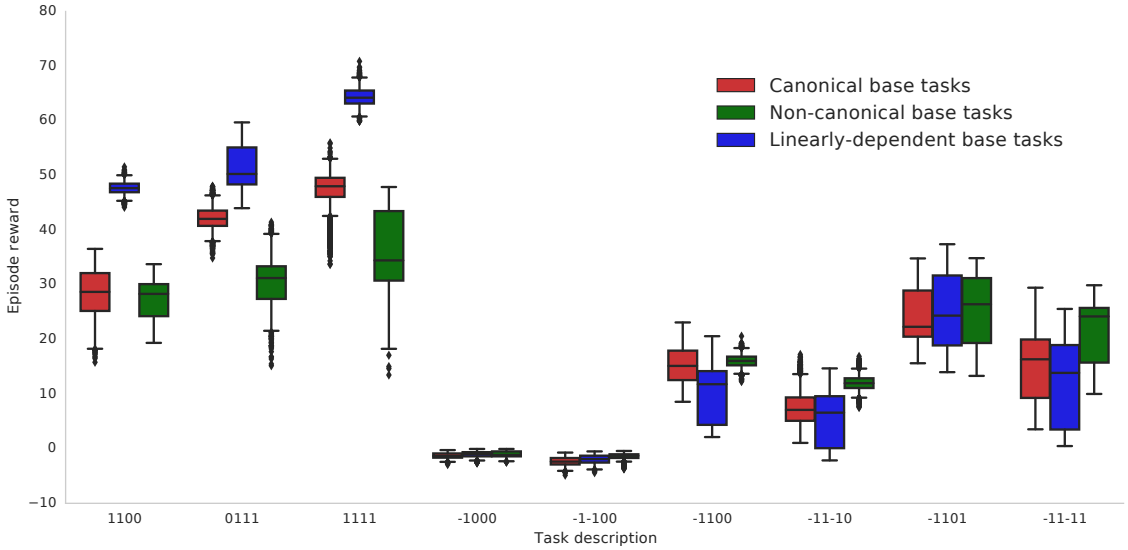


Figure 10. Performance of SF&GPI-transfer using base tasks $\hat{\mathcal{M}}$, $\hat{\mathcal{M}}'$ and $\hat{\mathcal{M}}''$, on the 9 test tasks. The box plots summarise the distribution of the rewards received per episode between 50 and 100 million steps of learning.